



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE ARTES Y HUMANIDADES  
INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN  
ARTES MULTIMEDIA**

**TÍTULO**

***DESARROLLO DE UNA APLICACIÓN INTERACTIVA PARA LA IMPLEMENTACIÓN DE REALIDAD VIRTUAL UTILIZANDO CASCOS DE INMERSIÓN QUE FACILITE EL APRENDIZAJE SOBRE EDUCACIÓN VIAL PARA PERSONAS QUE POSEEN LICENCIA DE CONDUCIR EN LA CIUDAD DE GUAYAQUIL.***

**AUTORES**

**Avilés Zavala Diana  
Reinoso Vásquez Alex**

**INGENIERO EN PRODUCCIÓN Y DIRECCIÓN EN ARTES  
MULTIMEDIA**

**TUTOR**

**Msc. Yamil Lambert Sarango**

**Guayaquil, Ecuador**

**2015**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE ARTES Y HUMANIDADES  
INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA**

## **CERTIFICACIÓN**

Certificamos que el presente trabajo fue realizado en su totalidad por Diana Gabriela Avilés Zavala, como requerimiento parcial para la obtención del Título de INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA.

### **TUTOR**

---

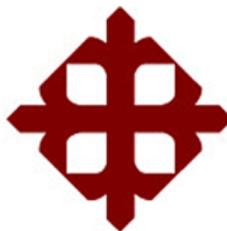
MSc. Yamil Lambert Sarango

### **DIRECTOR DE LA CARRERA:**

---

MSc. Víctor Hugo Moreno

**Guayaquil, a los 18 del mes de marzo del año 2015**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE ARTES Y HUMANIDADES  
INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA**

## **CERTIFICACIÓN**

Certificamos que el presente trabajo fue realizado en su totalidad Alex Santiago Reinoso Vásquez, como requerimiento parcial para la obtención del Título de INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA.

### **TUTOR**

---

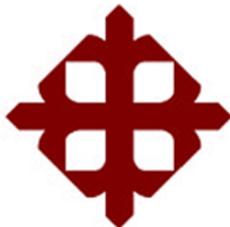
Msc. Yamil Lambert Sarango

### **DIRECTOR DE LA CARRERA**

---

MSc. Víctor Hugo Moreno

**Guayaquil, a los 18 del mes de marzo del año 2015**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE ARTES Y HUMANIDADES  
INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA**

## **DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Diana Gabriela Avilés Zavala**

### **DECLARO QUE:**

El Trabajo de Titulación, **DESARROLLO DE UNA APLICACIÓN INTERACTIVA PARA LA IMPLEMENTACIÓN DE REALIDAD VIRTUAL UTILIZANDO CASCOS DE INMERSIÓN QUE FACILITE EL APRENDIZAJE SOBRE EDUCACIÓN VIAL PARA PERSONAS QUE POSEEN LICENCIA DE CONDUCIR EN LA CIUDAD DE GUAYAQUIL**. Previa a la obtención del Título de **INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del Trabajo de Titulación referido.

**Guayaquil, a los 18 del mes de marzo del año 2015**

**EL AUTOR**

---

**Diana Gabriela Avilés Zavala**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE ARTES Y HUMANIDADES  
INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA**

## **AUTORIZACIÓN**

**Yo, Diana Gabriela Avilés Zavala**

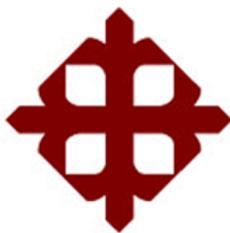
Autorizo a la Universidad Católica de Santiago de Guayaquil, la **publicación** en la biblioteca de la institución del Trabajo de Titulación: **DESARROLLO DE UNA APLICACIÓN INTERACTIVA PARA LA IMPLEMENTACIÓN DE REALIDAD VIRTUAL UTILIZANDO CASCOS DE INMERSIÓN QUE FACILITE EL APRENDIZAJE SOBRE EDUCACIÓN VIAL PARA PERSONAS QUE POSEEN LICENCIA DE CONDUCIR EN LA CIUDAD DE GUAYAQUIL.**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

**Guayaquil, a los 18 del mes de marzo del año 2015**

**EL AUTOR:**

---

**Diana Gabriela Avilés Zavala**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE ARTES Y HUMANIDADES  
INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA**

## **DECLARACIÓN DE RESPONSABILIDAD**

**Yo, Alex Santiago Reinoso Vásquez**

### **DECLARO QUE:**

El Trabajo de Titulación, **DESARROLLO DE UNA APLICACIÓN INTERACTIVA PARA LA IMPLEMENTACIÓN DE REALIDAD VIRTUAL UTILIZANDO CASCOS DE INMERSIÓN QUE FACILITE EL APRENDIZAJE SOBRE EDUCACIÓN VIAL PARA PERSONAS QUE POSEEN LICENCIA DE CONDUCIR EN LA CIUDAD DE GUAYAQUIL**. Previa a la obtención del Título de **INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del Trabajo de Titulación referido.

**Guayaquil, a los 18 del mes de marzo del año 2015**

**EL AUTOR**

---

**Alex Santiago Reinoso Vásquez**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE ARTES Y HUMANIDADES  
INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN ARTES MULTIMEDIA**

## **AUTORIZACIÓN**

Yo, **Alex Santiago Reinoso Vásquez**

Autorizo a la Universidad Católica de Santiago de Guayaquil, la **publicación** en la biblioteca de la institución del Trabajo de Titulación: **DESARROLLO DE UNA APLICACIÓN INTERACTIVA PARA LA IMPLEMENTACIÓN DE REALIDAD VIRTUAL UTILIZANDO CASCOS DE INMERSIÓN QUE FACILITE EL APRENDIZAJE SOBRE EDUCACIÓN VIAL PARA PERSONAS QUE POSEEN LICENCIA DE CONDUCIR EN LA CIUDAD DE GUAYAQUIL**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

**Guayaquil, a los 18 del mes de marzo del año 2015**

**EL AUTOR:**

---

**Alex Santiago Reinoso Vásquez**

## **AGRADECIMIENTO**

*Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.*

*Le doy gracias a mis padres Raúl y Katya por apoyarme en todo momento, por los valores que me han inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Sobre todo por su cariño que me inspiraron de forma incondicional, entendieron mis ausencias y mis malos momentos.*

*A Alex mi novio, por ser una parte importante de mi vida y compañero de Tesis. Por haberme apoyado en las buenas y en las malas, sobre todo por su paciencia y amor incondicional.*

*Le agradezco la confianza, apoyo y dedicación de tiempo a mi tutor: Msc Yamil Lambert por haber compartido con nosotros sus conocimientos y ayuda.*

**A todos ustedes mi mayor reconocimiento y gratitud....**

*Diana Gabriela Avilés Zavala*

## **AGRADECIMIENTO**

*Agradezco a Dios por haberme dado la fuerza, la valentía, la creatividad y el ingenio para seguir adelante en mis estudios, que gracias a él pude terminar una etapa más de mi vida llena de éxitos y triunfos, agradezco a mi madre quién con su gran ejemplo de lucha y superación a forjado en mí una gran admiración y motivación para seguir adelante, a mi hermana con quién he compartido toda mi vida momentos buenos y malos, a mi hermosa novia Diana Aviles quien ha estado conmigo todos estos años de estudio y quién ha sido de apoyo incondicional en mis decisiones y proyectos, a la miss Rocío Benavides quien a sido como una segunda madre con sus consejos y apoyo han sido parte importante en mi carrera profesional, al profesor Wellington Villota y Yamil Lambert quienes estuvieron siempre prestos apoyar en este y otros proyectos demostrando las ganas y empeño al momento de enseñar, agradezco a todos y quienes conforman la Facultad de Artes y Humanidades, Decana Dra. Lourdes Estrada, Director de Carrera Lcdo. Víctor Hugo Moreno, Profesores, compañeros y amigos.*

*Alex Santiago Reinoso Vásquez*

## **DEDICATORIA**

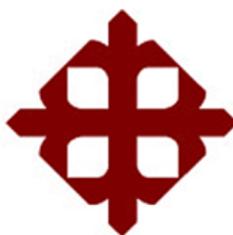
*Dedico mi tesis a Dios, a mis padres quienes me dieron vida, educación, apoyo y consejos. A mis compañeros de clase, a mis maestros y amigos., quienes sin su ayuda nunca hubiera podido hacer esta tesis. A todos ellos se los agradezco desde el fondo de mi alma.*

*Diana Gabriela Avilés Zavala*

## **DEDICATORIA**

*Dedico este proyecto a mi Dios quien me ha dado la inteligencia para poder realizar este proyecto, a mi familia quienes han estado presente en cada momento con su apoyo y motivación, a mi novia con quien he realizado este proyecto, a mi querida Facultad donde obtuve los conocimientos necesarios para poder realizar este proyecto, Compañeros y amigos*

*Alex Santiago Reinoso Vásquez*



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD DE ARTES Y HUMANIDADES  
INGENIERÍA EN PRODUCCIÓN Y DIRECCIÓN EN  
ARTES MULTIMEDIA**

**CALIFICACIÓN**

---

**MSc. Yamil Lambert**

# ÍNDICE GENERAL

INTRODUCCIÓN .....	1
<b>1. CAPITULO I .....</b>	<b>3</b>
1.1 JUSTIFICACIÓN DEL TEMA .....	3
1.1.1 Marco teórico .....	6
1.1.2 Bases teóricas .....	17
1.1.3 Componentes de un sistema de realidad virtual .....	21
1.2 DETERMINACIÓN DEL PROBLEMA .....	24
1.3 PREGUNTA/PROBLEMA DE INVESTIGACIÓN .....	24
1.4 OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS .....	25
<b>2. CAPITULO II .....</b>	<b>26</b>
2.1 INSTRUMENTOS DE INVESTIGACIÓN .....	26
2.2 RESULTADOS DE LA INVESTIGACIÓN .....	27
<b>3. CAPITULO III .....</b>	<b>31</b>
3.1 DESCRIPCIÓN DEL PROYECTO .....	31
3.2 ALCANCE .....	33
3.3 ESPECIFICACIONES FUNCIONALES .....	36
3.4 MÓDULOS DE LA APLICACIÓN .....	41
3.4.1 Módulo del hardware .....	41
3.4.2 Módulo del asistente virtual – Aprendizaje .....	43
3.5 ESPECIFICACIONES TÉCNICAS .....	43
3.6 FUNCIONES DEL APLICATIVO .....	46
3.7 DESCRIPCIÓN DEL USUARIO .....	47
3.7.1 Usuario .....	47
3.7.2 Asistente virtual .....	47
CONCLUSIONES Y RECOMENDACIONES .....	48
Conclusiones .....	48
Recomendaciones .....	52
BIBLIOGRAFÍA .....	53
ANEXOS .....	54
A) Encuesta realizadas a personas en la ciudad de Guayaquil .....	54
B) Porcentajes de las respuestas de las encuestas .....	57
C) Código que debimos resolver .....	59

# ÍNDICE DE TABLAS

<i>Tabla 1 Objetivos generales y específicos.....</i>	<i>25</i>
---	-----------

## ÍNDICE DE GRÁFICOS

<i>Ilustración 1 Reportes de incidentes del ECU911</i>	4
<i>Ilustración 2 Porcentaje de personas que poseen Smartphone en la ciudad de Guayaquil</i>	5
<i>Ilustración 3 Entorno de Unity</i>	10
<i>Ilustración 4 Unity domina el mercado</i>	13
<i>Ilustración 5 Participación por motor de los juegos móviles 3D</i>	14
<i>Ilustración 6 Giroscopio</i>	33
<i>Ilustración 7 Pantalla dividida</i>	34
<i>Ilustración 8 Insertar celular en case</i>	36
<i>Ilustración 9 Menú aplicación</i>	37
<i>Ilustración 10 Case</i>	37
<i>Ilustración 11 Seleccionar vistas de la aplicación</i>	38
<i>Ilustración 12 Vista Aérea</i>	38
<i>Ilustración 13 Vista dentro del vehículo</i>	39
<i>Ilustración 14 Vista de la señal de tránsito a explicar</i>	39
<i>Ilustración 15 Sistema de enseñanza</i>	40
<i>Ilustración 16 Hardware de case y teléfono celular</i>	42
<i>Ilustración 17 Asistente virtual</i>	43
<i>Ilustración 18 Botón PLAY</i>	46
<i>Ilustración 19 Botón TEST</i>	46
<i>Ilustración 20 Botón CRÉDITOS</i>	46
<i>Ilustración 21 Encuesta</i>	54
<i>Ilustración 22 Reporte encuestas</i>	57

## RESUMEN

Nuestro proyecto que tiene como problemática “LOS ACCIDENTES DE TRÁNSITO” en la ciudad de Guayaquil; los que son provocados obviamente por personas que ya poseen licencia de conducir pero debido al tiempo existen señales de tránsito que a los conductores se les olvida por no ser muy comunes ni están muy entendibles. Con ello, hemos realizado un análisis por medio de una encuesta y del cual el resultado fue a favor de nuestro problema planteado. La estrategia que queremos desarrollar consiste en realizar una APLICACIÓN para un celular Smartphone que en conjunto con un case te envuelven en un recorrido virtual donde un vigilante te mostrará y explicará las señales de tránsito que se van aproximando.

**Palabras Claves:** Multimedia, Realidad Virtual, Teléfonos inteligentes, educación vial, señales de tránsito, accidentes

## **ABSTRACT**

Our project's problematic is "traffic accidents" in Guayaquil city, which are mainly caused by people who already have driving license, that forgot unusual traffic signals which are not frequently used, or are even difficult to understand. So we made an analysis from a survey, and the results were in favor of our proposed Tesis.

The strategy we want to do consists in the development of an Smartphone application designed to work as a head-mounted display, that envelops you into a virtual tour in which a virtual guide will explain you every sign you see on the road.

**Palabras Claves:** Multimedia, Virtual Reality, Smartphones, driver education, traffic signals, accidents

## INTRODUCCIÓN

El siguiente trabajo tiene como objetivo desarrollar una aplicación interactiva que facilite el aprendizaje sobre educación vial a personas que poseen licencia de conducir y con este proyecto lograr reducir la tasa de accidentes que se frecuentan en las vías públicas; ya sea esta por desconocimiento de las señales de tránsito u olvido de las mismas, permitiéndonos ver una solución que ayude a reducir en cierta forma un problema común en las vías que circulamos diariamente.

El aprendizaje y captación de los significados de las señales de tránsito son muy importantes ya que nos informan, previenen y regulan las acciones que un conductor debe tomar al estar conduciendo, para lo cual es necesario realizar recordatorio de las señales de tránsito que existen en las zonas rurales y urbanas de las diferentes calles de la ciudad de Guayaquil.

Posteriormente, analizamos un sistema que en la actualidad se está formando como una tendencia tecnológica, ya que sus diferentes estudios aplicados como en la ciencia, medicina, entretenimiento y muchas mas han dado resultados muy interesantes, este sistema de realidad virtual del cual nos hemos acogido para realizar nuestro proyecto, nos propone buenos resultados puesto a que ahora en nuestro tiempo esta al alcance de los desarrolladores multimedia, proyecto del cual al combinar la educación con la tecnología hacen una fusión de impacto positivo en la que las personas por medio de esta aplicación puedan ser inmersas en un mundo virtual creadas por un ordenador, llegando a ellos por sus sentidos morfológicos como la vista, oído, tacto etc. En la actualidad esta tecnología se sigue implementando para varios usos y uno de ellos de los cuales aún no se lo encuentra en las tiendas de aplicaciones para Smartphone es sobre educación vial con realidad virtual.

Existen varias investigaciones sobre el uso de la realidad virtual como método de aprendizaje y por la cual se ha descifrado varios componentes que debemos

tener en cuenta al realizar nuestra aplicación, pasar de lo real a un mundo virtual se necesita de algunas herramientas, es por eso que en nuestro proyecto hemos utilizado los conocimientos que hemos adquirido en nuestra carrera más la investigación previa de autores que han ido desarrollando esta tecnología, por lo cual consideramos utilizar un motor gráfico para desarrollar el entorno virtual como software y en conjunto un celular y un casco de realidad virtual como hardware.

En nuestro proyecto se aprecia la construcción de calles, letreros, casas, árboles y más objetos que hay en nuestro entorno real, seguido de la interacción por medio de un punto rojo que creamos que al tocar cierto botón con el solo movimiento de nuestra cabeza mientras estamos con un casco de realidad virtual logramos una interacción buena ya que al utilizar este tipo de casco tenemos limitaciones de controles exteriores donde tengamos que usar las manos.

# 1. CAPITULO I

## 1.1 Justificación del tema

Este proyecto tiene una gran utilidad y es bastante práctico para las personas que necesiten reforzar y recordar sus conocimientos sobre las señaléticas y con ello puedan tener nuestra aplicación en sus hogares ya que es un proyecto con un esquema metodológico sencillo, interactivo y novedoso en la ciudad, en el que pueden adquirir y recordar sus conocimientos ya obtenidos mientras se divierten con este medio de una forma más dinámica como es la realidad virtual y con un plus que le da usando un case virtual en conjunto con de su Smartphone.

Con ello, el usuario podría representar en su hogar una calle virtual e ir recorriendo y así recordando y aprendiendo cada señal de tránsito que existen en nuestro país. Tratando así de poco a poco ir evitando y reduciendo futuras infracciones o accidentes de tránsito por motivo de desconocimiento de los peatones o conductores.

Teniendo más en cuenta aún, que según las estadísticas del último censo que se realizó en el 2014 según la INEC la SIS ECU 911 Samborondón reflejan que de enero a mayo de 2014 se han registrado 16.881 accidentes de tránsito en las provincias de Guayas. Guayaquil es el cantón donde más accidentes se han suscitado en este período con 11.558, cifra que representa el 68,47%.

Los datos estadísticos señalan además que en Guayas los principales tipos de accidentes de tránsito son: choques (53,87%), atropellos (12,68%), vehículos en mal estado (12,03%) y accidentes de motocicleta (6,97%).



## RESUMEN DE REPORTES DE INCIDENTES DEL CENTRO NACIONAL ECU 911 SAMBORONDÓN

### REPORTES DE INCIDENTES DEL CENTRO NACIONAL ECU 911 SAMBORONDÓN

RESUMEN	Fecha	Despachos Realizados 2012		
		Subtotal por Tipo	Promedio diario	Porcentaje
Tipo de incidentes	De 3 enero a 31 diciembre 2012			
Policia		273027	752,14	71,02%
CTE		39797	109,63	10,35%
Salud/IESS		32416	89,30	8,43%
SNGR		1364	3,76	0,35%
CNEL		21874	60,26	5,69%
FFAA		307	0,85	0,08%
Bomberos		15652	43,12	4,07%
Subtotal		384437	1059,06	100,00%

### REPORTES DE INCIDENTES DEL CENTRO NACIONAL ECU 911 SAMBORONDÓN

RESUMEN	Fecha	Despachos Realizados 2013		
		Subtotal por Tipo	Promedio diario	Porcentaje
Tipo de incidentes	De 1 de Enero del 2013 al 3 de Diciembre del 2013			
Policia		431745	1281,14	74,15%
CTE		44869	133,14	7,71%
Salud/IESS		59309	175,99	10,19%
SNGR		1813	5,38	0,31%
CNEL		20220	60,00	3,47%
FFAA		512	1,52	0,09%
Bomberos		22979	68,19	3,95%
Subtotal		582289	1727,86	99,86%

### REPORTES DE INCIDENTES DEL CENTRO NACIONAL ECU 911 SAMBORONDÓN

RESUMEN	Fecha	Despachos Realizados TOTAL		
		Subtotal por Tipo	Promedio diario	Porcentaje
Tipo de incidentes	De 3 de Enero 2012 al 3 de Diciembre del 2013			
Policia		704772	1006,82	72,97%
CTE		84666	120,95	8,77%
Salud/IESS		91725	131,04	9,50%
SNGR		3177	4,54	0,33%
CNEL		42094	60,13	4,36%
FFAA		819	1,17	0,08%
Bomberos		38631	55,19	4,00%
Subtotal		965884	1379,83	100,00%

Ilustración 1 Reportes de incidentes del ECU911

Otra estadística importante es el porcentaje de personas en la ciudad de Guayaquil que posea Smartphone; según las estadísticas de la INEC; quien dice que en el último censo del 2013 se registró del 51,3% de personas que poseen un teléfono celular, el 16,9% posee un Smartphone y va en aumento considerablemente y así poder utilizar nuestra aplicación.



El 16,9% de las personas que posee un celular tiene un teléfono inteligente (SMARTPHONE), frente al 8,4% del 2011, es decir 8,5 puntos más.

Porcentaje de personas que tienen teléfono inteligente (SMARTPHONE) a nivel nacional

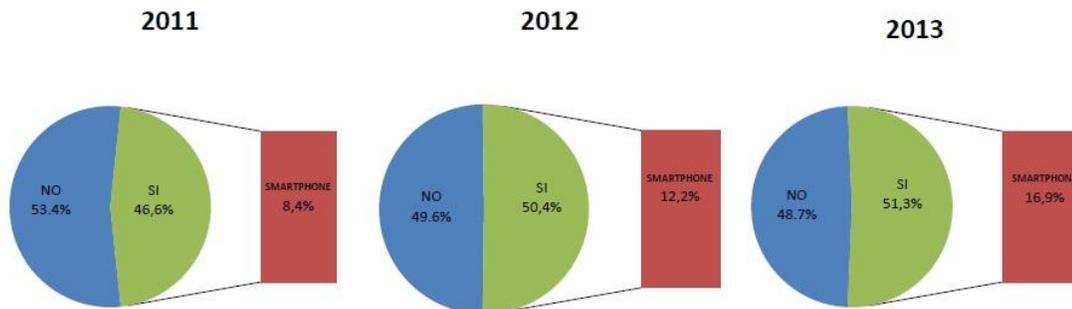


Ilustración 2 Porcentaje de personas que poseen Smartphone en la ciudad de Guayaquil

En nuestro proyecto utilizaremos el programa Unity 3D para poder elaborarlo como un videojuego interactivo en 3D. Usaremos ese programa ya que tenemos un conocimiento previo durante nuestra carrera en la Universidad y porque ese programa nos da una licencia para poder probarla primero. Unity 3D es un motor gráfico que sin duda alguna esta entre las mejores opciones en cuanto a desarrollo de entornos virtuales, integrando también en nuestro proyecto un casco de realidad virtual ya que actualmente existente una tendencia con esos artefactos, que así nos ayudará a tener una mayor interacción con las persona a que se integre a la aplicación, mostrándose así algunas animaciones, localidades, acciones y cuadros de dialogo de los diferentes ejemplos que se nos puede presentar mientras circulamos por las calles tanto como peatón y conductor.

### **1.1.1 Marco teórico**

Los accidentes de tránsito, catalogada como uno de los problemas de salud pública a nivel mundial con mayor impacto social y económico en la movilidad urbana. Dicho problema tiene como actores principales a los vehículos y peatones que interactúan frecuentemente por las vías de la ciudad, y básicamente la complejidad de la problemática radica en los factores que hacen posible su desarrollo, los cuales a su vez se encuentran estrechamente relacionados con el comportamiento propio de los actores viales ya mencionados.

En algunas partes del mundo se vienen realizando diferentes experimentos y propuestas interactivas que invitan a las personas a reflexionar acerca de ciertos comportamientos que intervienen en el desarrollo y el bienestar del espacio urbano. Un ejemplo claro de ello es "The Fun Theory" una iniciativa liderada por la compañía Volkswagen y la agencia de publicidad DDB Stockholm, que busca demostrar que mediante ideas novedosas y divertidas se puede incitar o cambiar el comportamiento de la personas en pro de un beneficio individual y/o colectivo (Volkswagen, 2009).. Este concepto, definido por B.J. Fogg, habla acerca de cómo un sistema computacional interactivo puede cambiar actitudes y comportamientos de la gente, haciendo que las personas se den cuenta de comportamientos inadecuados y obtengan indicios de cómo cambiarlos, para generar de esta manera cambios intencionales o voluntarios en su forma de actuar (Lab, 2007).

Técnicas de persuasión como la motivación, el cambio de comportamiento, la simulación y los cambios de puntos de vista, son algunas de las más utilizadas en tecnologías como el Internet, la realidad virtual, los dispositivos móviles y las redes sociales, que finalmente buscan influenciar a la gente para que adopte un comportamiento determinado Así pues, resulta interesante estudiar qué hay detrás de esa tendencia del diseño de tecnologías persuasivas, no con el objetivo de cambiar el comportamiento de las personas sino de más bien invitar a la gente a que reflexione sobre su forma de actuar dentro del espacio urbano, haciendo énfasis en un problema local como lo es el de los accidentes de tránsito de vehículos de cuatro ruedas dentro en la ciudad de Guayaquil.

Las definiciones de la realidad virtual son numerosas, quizás tantas como el número de autores que se han acercado al tema. Si nos detuviéramos en

algunas de ellas apreciaríamos que no siempre parecen estar hablándonos de lo mismo.

Desde la sencilla y parca definición de Aukstankanis y Blatner (1993:7) quienes afirman simplemente que “la realidad virtual es una forma humana de visualizar, manipular e interactuar con ordenadores y datos complejos” hasta las dudas terminológicas del francés Claude Cadoz (1994) que prefiere hablar de realidades virtuales o mejor aún de “representaciones integrales” el recorrido nos muestra las dificultades que presenta sintetizar en pocas palabras una técnica que aún no ha terminado de configurarse. Esto ha dado paso a que en demasiadas ocasiones se considere realidad virtual a aplicaciones que sólo colateralmente están relacionadas con ella.

Lo que define a un sistema de realidad virtual es, a nuestro juicio, su capacidad para estimular y engañar los sentidos a los que se dirige. Así, se puede considerar que un sistema de realidad virtual es:

Una base de datos interactivos capaz de crear una simulación que implique a todos los sentidos, generada por un ordenador, explorable, visualizable y manipulable en “tiempo real” bajo la forma de imágenes y sonidos digitales, dando la sensación de presencia en el entorno informático. Cuantos más sean los sentidos implicados en el engaño mayor será la intensidad de la experiencia simulada. ¿O deberíamos decir vivida? No faltan autores que así parecen sugerirlo, cuando advierten que la simulación digital multisensorial puede reforzar el riesgo de pérdida de la noción de realidad, “dando un carácter pseudoconcreto y pseudopalpable a entidades imaginarias” (Quéau 1995:41).

O cuando definen a un sistema de realidad virtual como un mundo que a pesar de no tener ninguna realidad física es capaz de darle al usuario, a través de una estimulación adecuada de su sistema sensorial, la impresión perfecta de estar en interacción con un mundo físico (Coiffet 1995:14). Así, para Biocca y Levy (1995:17) el objetivo de un interfaz de realidad virtual es conseguir “la inmersión completa de los canales sensomotores humanos en una experiencia vital generada por ordenador”

- Pimentel y Texeira (1995:240 y sigs.) por su parte señalan que la realidad virtual es un nuevo camino para explorar la realidad.

Una extensión de los sentidos mediante la cual podemos aprender o hacer algo con la realidad que no podíamos hacer antes. Una técnica que permite también percibir ideas abstractas y procesos para los cuales no existen modelos físicos o representaciones previas

Un sistema para poder ser considerado de realidad virtual debe ser capaz de generar digitalmente un entorno tridimensional en que el usuario se sienta presente y en el cual pueda interactuar intuitivamente y en “tiempo real” con los objetos que encuentre dentro de él.

Los objetos virtuales deben ser tridimensionales, poseer propiedades propias, tales como fricción y gravedad y mantener una posición y orientación en el ambiente virtual independiente del punto de vista del usuario. El usuario debe tener libertad para moverse y actuar dentro del entorno sintético de un modo natural. De tal forma que la sensación de presencia será mayor cuanto más sean los canales sensoriales estimulados.

De todos atributos mencionados, la sensación de presencia y la interactividad son los más importantes y los que distinguen a las realidades inmateriales de otros sistemas de simulación y de diseño asistido por ordenador (Wilson y al.1996:4).

El realismo de un entorno virtual está determinado por:

- Resolución y fidelidad de la imagen
- Reproducción de las propiedades de los objetos y de los escenarios virtuales.
- Reacciones de los objetos: Deben reaccionar del mismo modo que lo haría el objeto real en el momento de sufrir cualquier tipo de manipulación.
- Interactividad: El usuario debe poder moverse y actuar en el entorno virtual de un modo intuitivo y en “tiempo real”
- “Feed-Back” o respuesta sensorial: El usuario debe poder percibir tanto la firmeza o elasticidad del objeto virtual, como del resto de indicadores táctiles y propioceptivos.
- 

La escena virtual no debe ser silenciosa, debe incluir también sensaciones auditivas.

La sensación de presencia (o inmersión) se obtiene a través de la interactividad sensorial (visual, auditiva, táctil, muscular, etc). Cuanto más sentidos estén

implicados mayor es la sensación experiencia vivida que se consigue. Para que la inmersión sea verdaderamente realista el sistema debe ser capaz de crear una simulación sensorial completa o lo más próximo posible a ella. Es importante, además, que el usuario pueda ver en la imagen virtual una representación morfológica de alguna parte de su cuerpo (una o dos manos, brazos, cabeza, etc) para que le sirva como guía espacial dentro del entorno digital.

Presentamos un proyecto tecnológico con herramientas de modelado de bases de datos para simulación de conducción, incluyendo las especificaciones del trazado real de carreteras basadas en información independiente para la planta, el alzado y el perfil. Se explica la representación de esta información en una estructura de datos adecuada, la generación de tiras poligonales y la organización de esta información en una estructura jerárquica compatible con la visualización en tiempo real. Se discute igualmente, cuál puede ser el interfaz de edición adecuado para manipular interactivamente estos diseños por medio de puntos de control. El sistema de generación de estructuras visuales ha sido utilizado para crear los escenarios tridimensionales de varias aplicaciones de simulación de conducción.

Buena prueba de ello es la aparición en los últimos años de secciones especializadas dentro de los congresos tradicionales sobre simulación (ITEC - International Training Equipment Conference-, RTS -Real Time Systems-) y de seminarios y congresos propios como el DSC (Driving Simulation Conference). A este desarrollo ha contribuido la disminución de los costes de los equipos gráficos, plataformas móviles y otros componentes, pero también el avance en las técnicas de representación de los escenarios y el cálculo dinámico de movimientos y del tráfico, aspectos que hacen a la simulación de conducción más compleja que otros tipos de simulación, como la de vuelo.

Uno de los aspectos donde se ha avanzado grandemente, pero que todavía plantea desafíos importantes, es el de la generación de los datos referentes al escenario donde se va a desarrollar la simulación, incluyendo las carreteras, los elementos directamente relacionados con ellas (semáforos, señales de tráfico) y otros simplemente decorativos (edificios, terreno circundante). La generación de todos los datos necesarios es una tarea compleja, puesto que se trata de componentes heterogéneos que, sin embargo, deben compartir una estructura básica común y están ligados por multitud de relaciones.

## Software de desarrollo de realidad virtual UNITY 3D

### ¿Qué es Unity 3D?

Más que un motor, Unity es un ecosistema completo de herramientas y servicios diseñados para personas que desea desarrollar un negocio exitoso a través de la creación de juegos multiplataforma y contenido interactivo.

El ecosistema Unity está disponible para todos aquellos que descarguen el motor Unity que integra, en una plataforma sin paralelos, las herramientas que necesita para crear contenido interactivo 2D y 3D; soluciones de colaboración; despliegue rápido multiplataforma, así como servicios de retención, publicidad y análisis que harán crecer su negocio.

### Creación de contenido

Cada herramienta que necesita para crear cualquier tipo de contenido interactivo, directamente en el motor o desde la Tienda de Activos



Ilustración 3 Entorno de Unity

## **Unity free**

Descargue la versión gratuita del motor Unity en este momento. Sin limitaciones ni obligaciones de pago de regalías. La versión gratuita y totalmente funcional de nuestro software viene con una versión de prueba gratuita de 30 días de Unity Pro.

## **Unity pro**

Unity Pro pone a su disposición el conjunto de herramientas completo de Unity para juegos y contenido interactivo refinados, profesionales y de alto rendimiento. Compre Unity Pro o suscríbase por USD 75/mes durante 12 meses.

## **Industria**

Unity es una herramienta para desarrollar contenido interactivo y desplegarlo en cualquier parte. Ya sea que esté creando juegos, recorridos arquitectónicos, simulaciones de entrenamiento online o arte interactivo, Unity es la herramienta que le permitirá lograr resultados asombrosos en forma rápida.

## **Documentación**

La documentación de Unity le ofrece un listado completo de todos los componentes, sistemas y nuestra API de scripting.

Utilice el Manual para leer descripciones detalladas de los flujos de trabajo de Unity e investigar a fondo nuestra base de códigos con la referencia Scripting.

## **Manual de unity**

Unidad se hace para empoderar a crear la mejor experiencia de entretenimiento interactivo o multimedia que se puede. Este manual está diseñado para ayudarle a aprender cómo utilizar la unidad, desde el básico hasta técnicas avanzadas. Se puede leer de principio a fin o se utiliza como una referencia.

Si usted encuentra que cualquier pregunta que usted tiene no es contestada en este manual por favor pregunte en Unity respuestas o Unity foros . Usted será capaz de encontrar su respuesta allí.

### **Apl de scripting de unity.**

Esta sección de la documentación contiene detalles de la API de scripting que ofrece la Unidad. Para usar esta información, usted debe estar familiarizado con la teoría básica y la práctica de secuencias de comandos en la Unidad que se explica en la sección Automatización de nuestro manual.

La referencia de secuencias de comandos está organizado de acuerdo con las clases disponibles para las secuencias de comandos que se describen junto con sus métodos, propiedades y cualquier otra información pertinente a su uso.

Las páginas están ampliamente equipadas con código de ejemplo que usted es libre de utilizar para cualquier propósito sin acreditar Unidad. Los ejemplos se pueden ver en cualquiera de los tres idiomas soportados ( C # , JavaScript y Boo ) mediante el menú en la parte superior de cada página. Tenga en cuenta que la API es el mismo, independientemente de que se utiliza el lenguaje, por lo que la elección del lenguaje es puramente una cuestión de preferencia.

Subsecciones de la referencia se pueden seleccionar usando el menú de la izquierda. Para la mayoría de los usuarios, el tiempo de ejecución de las clases sección será el principal puerto de llamada. Otras secciones de la API, incluyendo la API de extensión Editor se pueden seleccionar en el menú desplegable en la parte superior de la lista de clases.

## Cuarenta y cinco por ciento de desarrolladores

El motor de juegos Unity es de lejos más popular entre los desarrolladores que cualquier otro software de desarrollo de juegos. La proporción de desarrolladores que confían en Unity como su herramienta de desarrollo principal y que utilizan Unity experimenta un crecimiento permanente.

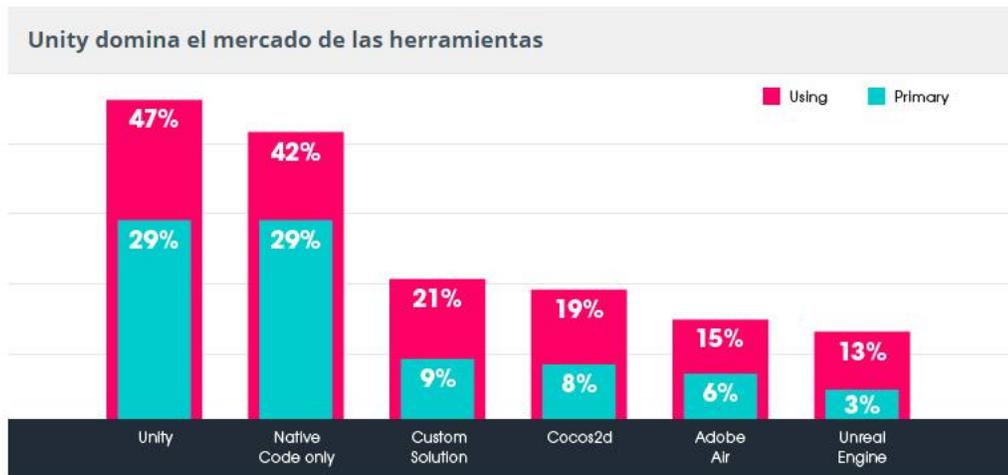
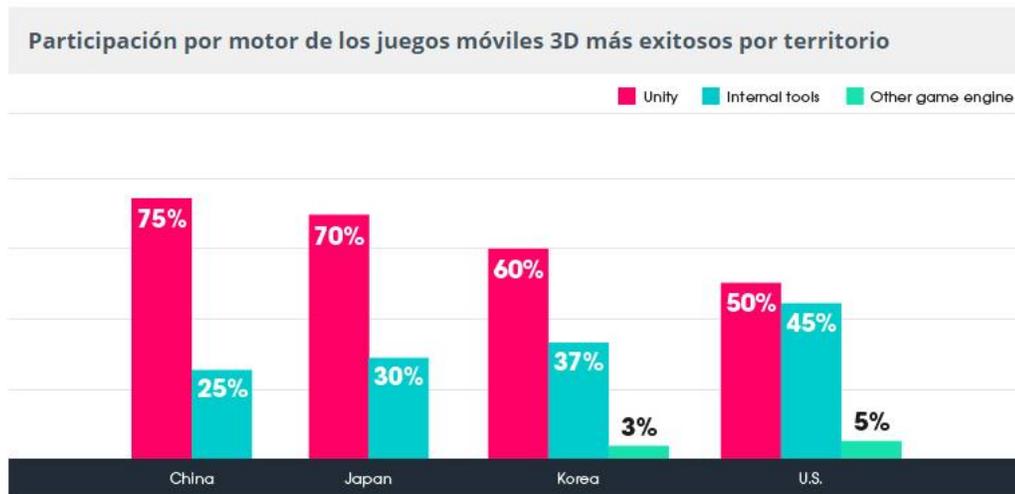


Ilustración 4 Unity domina el mercado

## Unity domina los juegos móviles 3d

En cada uno de los mercados importantes del mundo, la gran mayoría de los juegos móviles 3D más exitosos creados con herramientas de terceros se han creado con Unity.



*Ilustración 5 Participación por motor de los juegos móviles 3D*

### **Sdk utilizado en el desarrollo de la aplicación para realidad virtual**

Tecnología de seguimiento de buceo simplemente funciona.

Se ha desarrollado un plugin de vanguardia de baja latencia que trae de cabeza seguimiento a sus Aplicaciones y Juegos. Hemos tenido muchas solicitudes para también hacerlo iOS / iPhone / iPod.



Ahora el buceo Unidad Plugin es multiplataforma para iOS / Android.

Seguimiento de buceo es ahora para Android y iPhone / iPod. Su proyecto sigue siendo el mismo. Usted sólo tiene que arrastrar la configuración de la cámara de buceo a la escena y se obtiene un rayo Head Tracking rápido en ambas plataformas.

Descargue el paquete de plugins: Unidad de Buceo Plugin Paquete 2.0 para Android / iOS .

También puede descargar el proyecto de buceo Unidad Demo de 2.0 .

El uso básico es unir el guión sensor OpenDive a una cámara y compilar para el teléfono. Estén atentos para Videos Tutoriales y funciones ampliadas.

También puede descargar el buceo Unidad de demostración de la Google Play Store. Para obtener más consejos y preguntas por favor visite la Junta de buceo Durovis

## **Android Estudio general**

Android Studio es la IDE oficial para el desarrollo de aplicaciones para Android, basado en IntelliJ IDEA . Además de las capacidades que se esperan de IntelliJ, Android Studio ofrece:

- Sistema de construcción basado en Gradle Flexible
- Construir variantes y múltiples apk generación de archivos
- Plantillas de código para ayudarle a construir características de la aplicación comunes
- Editor de diseño Rich con soporte para la edición de arrastrar y soltar tema
- pelusa herramientas para la captura de rendimiento, facilidad de uso, compatibilidad de versiones, y otros problemas
- Capacidades ProGuard y aplicación de firma
- El soporte integrado para Google Cloud Platform , por lo que es fácil de integrar Google Cloud Mensajería y App Engine.

## Manager SDK

El SDK de Android separa herramientas, plataformas y otros componentes en paquetes que se pueden descargar mediante el Administrador de SDK. Por ejemplo, cuando las herramientas de SDK se actualizan o una nueva versión de la plataforma Android es liberado, se puede utilizar el Gestor de SDK para ellos descargar rápidamente a su entorno.

Puede lanzar el SDK Manager en una de las siguientes maneras:

- De Eclipse (con ADT ), seleccione Ventana > Administrador de Android SDK .
- Desde Android de estudio, seleccione Herramientas > Android > Manager SDK .
- En Windows, haga doble clic en el SDK Manager.exe archivo en la raíz del directorio del SDK de Android.
- En Mac o Linux, abra un terminal y vaya a la / herramientas de directorio en la ubicación donde está instalado el SDK de Android, a continuación, ejecutar Android SDK .
- Puede seleccionar los paquetes que desea descargar activando las casillas de verificación a la izquierda, a continuación, haga clic en Instalar para instalar los paquetes seleccionados.

## 1.1.2 Bases teóricas

### Introducción a la Realidad Virtual

Empezaremos por un apartado donde se explican los tres ingredientes elementales que tiene que tener un sistema de realidad virtual para ser considerado como tal: *simulación interactiva, interacción implícita e inmersión sensorial*.

A pesar de que en la literatura se pueden encontrar muchas definiciones de realidad virtual (en inglés, virtual reality VR), seguramente una de las más completas es la que propuso A. Rowell:

*“La Realidad Virtual es una simulación interactiva por computador desde el punto de vista del participante, en la cual se sustituye o se aumenta la información sensorial que recibe”.*

En esta definición aparecen los elementos básicos que tienen que estar presentes en todo sistema de realidad virtual y que se discuten a continuación:

- Simulación interactiva
- Interacción implícita
- Inmersión sensorial

#### ▪ **Simulación interactiva**

Una aplicación de realidad virtual es una simulación en el sentido de que se recrea un mundo virtual que sólo existe como una representación digital 3 en la memoria de un ordenador. El hecho de que la simulación sea interactiva es lo que distingue la realidad virtual de una animación. En una animación, al igual que en el cine, los espectadores son individuos pasivos en el sentido que no pueden alterar el contenido de las imágenes que ven: éstas han sido grabadas previamente desde un determinado punto de vista o siguiendo una determinada trayectoria. En cambio, en un sistema de realidad virtual, el usuario puede escoger libremente su movimiento por la escena y, por tanto, sus acciones afectan de forma directa a las imágenes que verá. Además, el usuario puede

improvisar su movimiento sobre la marcha, sin necesidad de tener que establecer un guión previo, pues el sistema de realidad virtual responderá en tiempo real (es decir, con un tiempo de respuesta despreciable) a sus acciones.

Dado que la trayectoria que seguirá el participante es imprevisible, y que hay infinitos puntos de vista diferentes, los sistemas de realidad virtual requieren tener una representación geométrica 3D del entorno virtual, de forma que puedan calcular la imagen correspondiente a un punto de vista arbitrario. Además de este modelo geométrico, los sistemas de realidad virtual también requieren utilizar algoritmos de visualización realista (algoritmos de síntesis de imágenes) para poder generar las imágenes a partir de estas representaciones digitales de la escena.

- **Interacción implícita**

La realidad virtual utiliza la interacción implícita en contraposición a la interacción explícita o interacción clásica.

En la interacción clásica, cuando un usuario quiere llevar a cabo una determinada acción (pongamos que quiere ver la parte del mundo virtual que tiene detrás) tiene que comunicar de forma explícita su voluntad al computador.

Para esta tarea, el usuario utiliza el esquema de comunicación determinado por la interfaz de la aplicación, ya sea una interfaz basada en comandos o una interfaz gráfica de tipo W. I. M. P. (window-icon-menu-pointing device), que es como se denominan los sistemas basados en ventanas e íconos. En cualquiera de estos casos, el usuario tiene que realizar un pequeño esfuerzo, para recordar o buscar el comando o el elemento gráfico asociado a la acción que quiere realizar, y debe hacer llegar al ordenador esta orden a través de los dispositivos de interacción clásica, fundamentalmente el teclado y el ratón.

A pesar que dé en los últimos años las interfaces gráficas en mejorado mucho la comunicación del hombre con el computador, la comunicación de acciones repetidas (como la trayectoria que quiere seguir el usuario) no deja de suponer un esfuerzo, y normalmente requiere de un cierto tiempo de entrenamiento o formación.

En cambio, en la realidad virtual el sistema captura la voluntad del usuario implícita en sus movimientos naturales. El ejemplo más claro es el control de la cámara virtual: en una sistema de realidad virtual, la cámara se actualiza en función de los movimientos de la cabeza del usuario. Si el usuario quiere ver la parte de mundo virtual que tiene detrás, no tiene que utilizar ningún comando ni mover el ratón, sino que simplemente debe hacer el mismo gesto natural (girar la cabeza) que haría en el mundo real: Otro ejemplo es la interacción con los objetos de la escena. Si el usuario quiere abrir una puerta o mover un objeto, lo único que tiene que hacer es coger la puerta o el objeto y llevar a cabo con la mano los mismos movimientos que haría con un objeto real.

Como podemos ver, tanto en interacción clásica como en interacción implícita se utilizan dispositivos periféricos para la entrada de datos, pero la diferencia fundamental es la percepción que tiene el usuario de estos dispositivos.

La revolución de la realidad virtual en la interacción hombre-máquina radica en que el usuario deja de percibir los dispositivos (incluso deja de percibir el propio ordenador) para pasar a interactuar directamente con los objetos de la escena. La diferencia psicológica es que el usuario deja de mirar por una ventana y pasa a estar dentro del entorno virtual.

- **Inmersión sensorial**

El tercer elemento fundamental de todo sistema de realidad virtual es la inmersión sensorial.

La palabra inmersión tiene significados muy variados según el contexto en que se utilice, pero en realidad virtual es un concepto muy claro. Podemos definir la inmersión sensorial como la desconexión de los sentidos del mundo real y la conexión al mundo virtual. Como consecuencia, el usuario deja de percibir el entorno que le rodea y pasa a estar inmerso dentro del mundo virtual que recrea el computador en tiempo real. De los diferentes órganos de los sentidos, la vista es el que nos proporciona una mayor cantidad de información y a la vez es la que da una mayor sensación de presencia. Por este motivo, todo sistema de realidad virtual debe proporcionar estímulos adecuados como mínimo para el sentido de la vista (es decir, debe generar las imágenes correspondientes al mundo virtual), a pesar de que a menudo los sistemas de realidad virtual proporcionan también inmersión acústica.

A menudo se dice que un sistema de realidad virtual crea el efecto de que los objetos del entorno virtual existen con independencia del dispositivo de visualización. Cuando miramos un modelo en una pantalla de un computador (sin realidad virtual), claramente distinguimos que los objetos están proyectados en la superficie de la pantalla, y no tenemos ninguna sensación que estos objetos tengan una existencia material, por más realista que sean las imágenes.

En cambio, la utilización de los dispositivos de visualización de realidad virtual crea en el usuario el efecto que los objetos no están proyectados en ninguna superficie, sino que se encuentran a diferentes distancias “flotando” en el espacio que envuelve al observador, como si tuviesen existencia material propia. La clave de este proceso es la visión estereoscópica. La visión estereoscópica se basa en proporcionar dos imágenes ligeramente diferentes del mundo virtual, una por cada ojo, de forma que el sistema visual humano deduce la profundidad de los objetos a partir de las diferencias en las imágenes.

La visión estereoscópica es un elemento imprescindible en todo sistema de realidad virtual, pues es el único medio para hacer que los objetos tengan una fuerte presencia espacial. La visión estereoscópica también es una herramienta muy útil para aplicaciones que no se pueden considerar de realidad virtual, ya

que permite ubicar espacialmente los objetos de forma mucho más precisa, gracias a que la escena se percibe en relieve.

### **1.1.3 Componentes de un sistema de realidad virtual**

En un sistema de realidad virtual se pueden distinguir elementos hardware y elementos software.

Los componentes hardware más importantes son el computador, los periféricos de entrada y los periféricos de salida.

Los componentes software más importantes son el modelo geométrico 3D y los programas de simulación sensorial (simulación visual, auditiva, táctil), simulación física (movimiento de la cámara virtual, detección de colisiones, cálculo de deformaciones), y recogida de datos. La siguiente figura ilustra los componentes de un sistema típico de realidad virtual:

*A continuación describiremos brevemente cada uno de estos componentes:*

#### **Periféricos de entrada (sensores)**

Los periféricos de entrada se encargan de capturar las acciones del participante y enviar esta información al computador. Los periféricos de entrada más frecuentes en realidad virtual son los posicionadores (que permiten al sistema conocer en tiempo real la posición y la orientación de la cabeza, de la mano, o de todo el cuerpo del usuario), los guantes (que permiten detectar movimiento de los dedos de la mano) y los micrófonos (que graban la voz del participante).

#### **Periféricos de salida (efectores)**

Los periféricos de salida se encargan de traducir las señales de audio, video, etc. generados por el computador en estímulos para los órganos de los sentidos (sonido, imágenes, . . . ). Los efectores se clasifican según el sentido al que va

dirigido: existen efectores visuales (cascos estereoscópicos, pantallas de proyección...), y de audio (sistemas de sonido, altavoces...) de fuerza y tacto (dispositivos táctiles), y del sentido del equilibrio (plataformas móviles).

## **Computador**

El computador se encarga de llevar a cabo la simulación de forma interactiva, basándose en el modelo geométrico 3D y en el software de recogida de datos, simulación física y simulación sensorial. Debido a que el proceso más crítico en realidad virtual es la simulación visual (síntesis de imágenes a partir de modelos 3D), los computadores que se utilizan para realidad virtual son estaciones de trabajo con prestaciones gráficas avanzadas, donde la mayor parte de las etapas del proceso de visualización están implementadas por hardware.

## **Modelo geométrico 3D**

Dado que un sistema de realidad virtual tiene que permitir explorar la escena de forma interactiva y ver el mundo virtual desde cualquier punto de vista, es necesario disponer de una representación geométrica 3D de este mundo, que permita hacer los cálculos de imágenes, generación de sonido espacial, cálculo de colisiones, etc. a los módulos que describiremos más adelante.

## **Software de tratamiento de datos de entrada**

Los módulos de recogida y tratamiento de datos se encargan de leer y procesar la información que proporcionan los sensores. Esto incluye los controladores de los dispositivos físicos, así como los módulos para el primer tratamiento de los datos suministrados. Por ejemplo, los datos de posición y orientación de la cabeza del usuario normalmente se tienen que transformar para expresarlas en un sistema de coordenadas de la aplicación y se deben filtrar para evitar saltos repentinos como consecuencia de lecturas erróneas de los valores de posición. Los sistemas que permiten la comunicación con el ordenador mediante órdenes orales requieren un sistema de reconocimiento de voz. Otros sistemas utilizan un esquema de comunicación basado en gestos de la mano (una especie de

lenguaje de sordo-mudos pero más sencillo) y que requiere el reconocimiento de gestos a partir de una secuencia de movimientos.

### **Software de simulación física**

Los módulos de simulación física se encargan de llevar a cabo las modificaciones pertinentes en la representación digital de la escena, a partir de las acciones del usuario y de la evolución interna del sistema. Por ejemplo, si el módulo de recogida de datos indica que el usuario tiene que hacer el gesto.

## **1.2 Determinación del problema**

Se evidencia un problema en el olvido de las señales de tránsito no muy comunes en las personas al momento de conducir un auto o moto en la ciudad. Con esto podemos darnos cuenta que es cierto, debido a las últimas estadísticas sobre accidentes de tránsito.

Y así se pretende con este proyecto incentivar a la gente a informarse acerca de lo que es la educación vial para evitar en un futuro, accidentes de tránsito, multas y sanciones al no respetar alguna señal por falta de conocimiento, motivando a la gente el aprender por medio de una herramienta tecnológica e interactiva los reglamentos y señales de tránsito que nos topamos al salir a las calles.

## **1.3 Pregunta/Problema de Investigación**

*¿Qué impacto causaría en las personas el aprender educación vial por medio de la realidad virtual?*

*¿El índice de accidentes disminuiría según la enseñanza que se les dé a las personas?*

## 1.4 Objetivo General y objetivos Específicos

<i>OBJETIVO GENERAL</i>	<i>OBJETIVOS ESPECÍFICOS</i>
Desarrollar una aplicación interactiva para facilitar el aprendizaje sobre educación vial a personas que poseen licencia de conducir	Reducir el desconocimiento u olvido sobre las señales de tránsito en los conductores entre 18 a 35 años de la ciudad de Guayaquil a pesar de haber realizado varios exámenes previos a la obtención de la licencia y con ello, las tasas de accidentes.
	Extender por medio de nuestro programa las buenas prácticas a sus amigos y familiares para que así participen y ser más interactivo y dinámico.
	Reforzar las señales de tránsito a los conductores.
	Crear mediante nuestro programa un proceso educativo y de concientización que así permita el buen uso de la vía pública.
	Influir en las tasas de accidentes.

*Tabla 1 Objetivos generales y específicos*

## **2. CAPITULO II**

### **2.1 Instrumentos de Investigación**

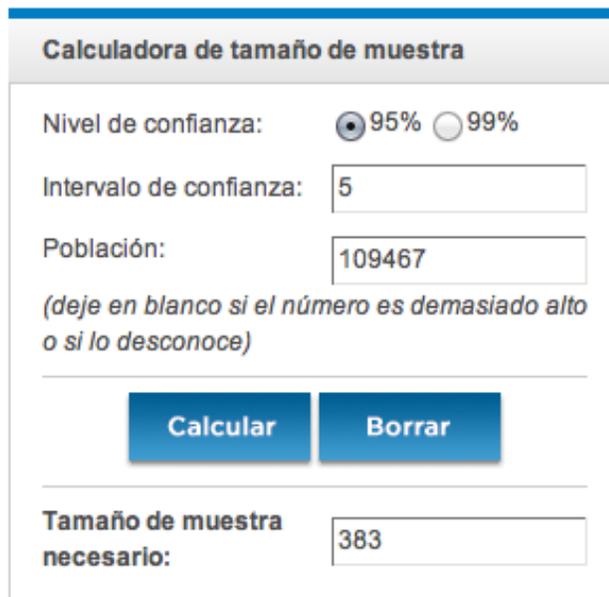
En este proyecto de titulación se ha realizado como instrumento de investigación dos métodos:

- Consultas bibliográficas (revistas indexadas, libros y páginas web certificadas)  
(Observar en las referencias bibliográficas)
- Encuestas a personas en la ciudad de Guayaquil (para poder comprobar si nuestro proyecto tiene viabilidad)
- Hablar con un experto de la Comisión de tránsito.- le preguntamos el total de personas que poseen licencias de conducir tipo B

## 2.2 Resultados de la Investigación

Realizamos una encuesta a un total de 383 personas en la ciudad de Guayaquil en la Universidad Católica entre docentes y estudiantes, en los centros comerciales y vía internet por Facebook. Para obtener ese número de encuestados hemos realizado una fórmula para poder calcular la muestra de una población.

La población exacta de personas en la ciudad de Guayaquil que tienen licencias de conducir son de 109.467 personas según Israel Eduardo Campuzano Saenz con el cargo de Vigilante en la Comisión de Tránsito.



The image shows a web-based calculator titled "Calculadora de tamaño de muestra". It features the following elements:

- Nivel de confianza:** Two radio buttons are present, with "95%" selected and "99%" unselected.
- Intervalo de confianza:** A text input field containing the number "5".
- Población:** A text input field containing the number "109467".
- (deje en blanco si el número es demasiado alto o si lo desconoce)*
- Buttons:** Two blue buttons labeled "Calcular" and "Borrar".
- Result:** Below the buttons, the text "Tamaño de muestra necesario:" is followed by a text input field containing the number "383".

Para así poder comprobar que muchos guayaquileños/as que poseen licencia de conducir, después de haber aprobado sus respectivos exámenes de conducción para obtener su licencia, desconocen ciertas señales de tránsito que deberían saberlas tras haber aprobado su examen.

Los resultados obtenidos en esta encuesta nos dan a conocer que:

## El genero

Sobre el total de 383 personas encuestadas existen un 66% de encuestados que son HOMBRES y un 34% que son MUJERES.

## Edades

Nuestra encuesta fue realizada a personas entre un rango de edades que lleva desde los 18 años, ya que es la mínima edad para poder obtener una licencia de conducir en Guayaquil, hasta los 34 años.

Encontramos conductores entre 18-20 años (6%), entre 21-23 años (32%), entre 24-26 años (27%), entre 27-30 años (15%), entre 31-34 años (25%).

## Posee auto

Entre ese total de personas encuestadas para nuestro proyecto encontramos que un 62% de personas entre hombres y mujeres poseen un auto activo en Guayaquil. Y un 38% de personas que no poseen licencia de conducir.

## ¿Qué tipo de licencia posee?

Entre las personas que tienen licencia su mayoría con un 77% poseen licencia tipo B (automotores de hasta 3500 kg y con remolque de hasta 750 kg, sin fines profesionales. Es decir conductores particulares). Y del tipo A un total del 8%.

## ¿Sabe cuántos puntos tiene su licencia de conducir?

Y el 57% de personas conocen el total de sus puntos actuales.

Identifique para que sirven estas señales

En las siguientes 7 preguntas de nuestra encuesta se encontraban imágenes de señales de tránsito que encontramos en la ANT (Agencia Nacional de Tránsito) junto con respuestas falsas para despistar a los encuestados. De las cuales 5 de las 7 preguntas nuestros encuestados respondieron incorrectamente y las 2 restantes respondieron correctamente. Dando así a entender que al haber obtenido ellos su licencia de conducir previamente, a mis encuestados no les aseguró el poder responder correctamente todas nuestras preguntas sobre el significado de algunas de las señales de tránsito y por ende mi proyecto serviría de mucho, para poder obtener de una manera más dinámica, interactiva y espontánea a los guayaquileños mejores resultados para conocer o recordar todas las señales de tránsito existentes.

¿Qué tipo de realidad virtual usted conoce?

En nuestra siguiente pregunta obtuvimos que la mayoría de nuestros encuestados tenía una idea de los tipos de realidad virtual dejando en primer lugar a los simuladores con un 57%, las gafas virtuales con un 52%, el xbox (kinect) con un 39%. Pero existen un grado de confusión al tener un porcentaje un poco alto 2 opciones que no son consideradas realidad virtual como la Televisión con 46% y el cine con 48%.

¿Usted posee un teléfono celular?

Del total de nuestros encuestados encontramos que un 98% de personas entre hombres y mujeres tienen al menos un teléfono celular y un 2% de personas que no lo poseen. Dejando como mayoría a las personas que si poseen, así podrán adquirir nuestro producto y tenerlo en sus hogares para poder practicar y hasta interactuar con amigos y familiares sobre ello, haciendo un tema como son las señaléticas de tránsito no tan interesantes a poder aprender mientras hacen su recorrido virtual de una manera más divertida.

¿Usted ha sufrido algún tipo de accidente automovilístico?

Otro punto importante que encontramos en nuestro proyecto es preguntar sobre si han tenido o no accidentes de tránsito. Y un 34% dijo que sí ha tenido por lo menos un accidente en su vida y un 66% no ha tenido.

¿Porque motivo fue su accidente?

También pudimos rescatar que un 28% de los accidentes que se han dado a mis encuestados a sido por causa de choques, un 4% por no respetar un pare y problemas mecánicos; y por último un 3% por pasarse una roja y no respetar la zona cebra.

Con todos estos resultados que hemos obtenido de nuestra encuesta, llegamos a la conclusión y confirmamos que los guayaquileños necesitan de algo para poder reforzar lo ya aprendido y por ello nosotros con nuestro proyecto queremos ayudar a implementar esta forma más dinámica para reforzar lo ya aprendido en los cursos de conducción existentes.

Aunque en este mes de Diciembre, jueves 4 del 2014, se aprobó en la Asamblea Nacional que las personas que desean obtener la licencia de conducir tipo B, no será necesario el realizar un curso para poder obtenerla sino que directamente deben acercarse a realizar "las pruebas psicométricas, teóricas y proactivas tomadas" por la Agencia Nacional de Tránsito (ANT). Más sino aprueba, deberá realizar el curso.

Con esta medida al no haber un curso de por medio como obligación, las personas no sabrán bien como empezar y solo aprenderían al cómo manejar un auto más no a conocer y aprenderse todas las señales de tránsito, ya que existen personas fuera de la CTE (Comisión de Tránsito del Ecuador) que venden libritos con las preguntas que serán tomadas en el examen teórico.

## **3. CAPITULO III**

### **3.1 Descripción del Proyecto**

Nuestro proyecto trata de una aplicación para teléfonos Smartphone en conjunto con unas gafas de realidad virtual incluidas, donde la pantalla es cualquier celular de pantalla grande que se agregue, dándole así, el toque innovador, divertido y que además, se lo puede realizar desde la comodidad del hogar las veces que el usuario desee para así poder reforzar lo ya aprendido.

Con nuestro programa hecho en Unity3D, basado en objetos tridimensionales cualquier persona ya habiendo obtenido su licencia de conducir puede ir practicando para que no se pueda ir olvidando esas señales que no son muy comunes verlas que existen registradas en el Ecuador en la ANT. Gracias a este programa que le simplificará las cosas y le será mucho más de su agrado recordar.

El usuario al abrir nuestra aplicación en su celular y al colocarse las gafas de realidad virtual, se observará que esa persona se encuentra dentro del programa siendo el conductor de un auto, en una calle, dentro de una ciudad.

Y así, una vez dentro de la aplicación, el usuario será guiado por un ayudante (oficial) en un recorrido virtual como si manejara dentro de la ciudad donde se presentarán en cada esquina de las calles, las distintas señales de tránsito existentes y el para qué sirve cada una de ellas.

Entre las más comunes señales de tránsito encontramos: el pare, la zona cebrada, el ceda el paso, el prohibido estacionar, etc. Y las no muy comunes como las señales de servicio y de atractivo turístico que son más vistas en carreteras. Dicha información sobre cada señal de tránsito encontrada y su utilidad se le mostrará en la práctica y en texto en su pantalla.

Al realizar esta aplicación obtendremos que varias personas que conocen las señales de tránsito, logren mantener lo aprendido en la realidad, reforzándolo en un entorno virtual y así seguir aplicándolo en la vida cotidiana, evitando así los diferentes tipos de accidentes, sanciones y faltas por motivo del olvido.

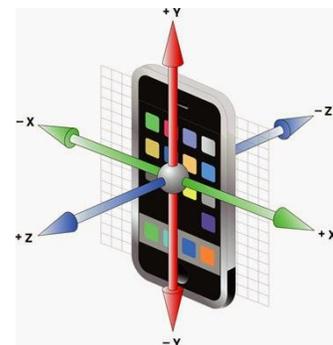
Al final del recorrido, el usuario tendrá una pequeña evaluación para así poder constatar que él aun recuerda y sigue apto para recorrer las calles habiendo aprobado el test final por nuestro programa.

## 3.2 Alcance

Se pretende con este proyecto desarrollar una aplicación con la tecnología de realidad virtual utilizando un casco o visor, el cual una vez adquirido la persona que lo esté utilizando aprenda de una manera más interactiva e inversiva, por lo cual se pretende que con esta aplicación llegue a quedarse en la mente de los usuarios.

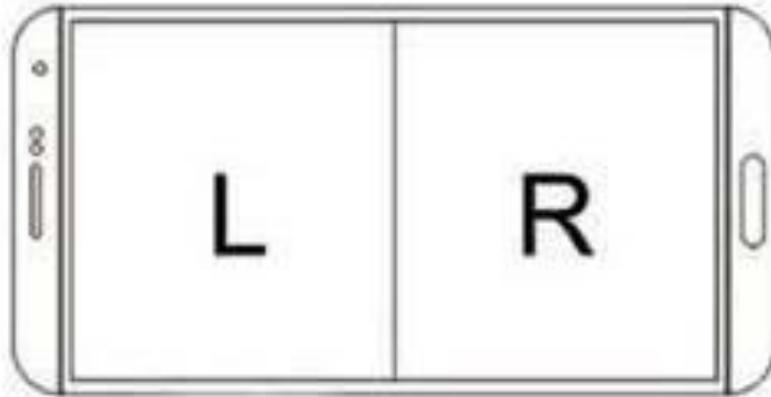
La aplicación trae consigo un sin número de elementos tanto audiovisuales como multimedia, siendo estos como imágenes, animaciones, gráficos en 3D, botones, sonidos, texto, etc. Tratando así, lograr una mejor recepción de información educacional, y como en nuestro caso hemos escogido el desarrollo de una aplicación para educación vial, también se lo puede implementar para diferentes áreas y temas que nuestra sociedad lo necesite.

Algo novedoso que nuestra aplicación contará, es con una tecnología que se llama “Giroscopio” donde el usuario al tener las gafas puestas y mirando para cualquier dirección, él se sentirá que está dentro de la aplicación porque al mover el celular y cambiar su orientación ya sea arriba, abajo, derecha o izquierda podrá observar todo el área en el que se encuentra como el cielo, suelo y los alrededores.



*Ilustración 6 Giroscopio*

Otra particularidad de nuestra aplicación, es que consta con una pantalla dividida en dos, que al poner el celular en las gafas y tenerlo a cierta distancia de nuestros ojos hace que podamos observar las dos pantallas por separado como si fueran una sola debido a que nuestro cerebro las une, generando una imagen tridimensional más creíble y por ende un grado de inmersión muy alto.



*Ilustración 7 Pantalla dividida*

Además, el recorrido virtual que hará la aplicación será guiado por un ayudante quien te explicará las características de la señal que estarás observando al pasar cerca de ella, ya sea dentro del vehículo o en una vista desde afuera, esto se dará en todo el recorrido de las calles, también se podrá escuchar música si lo desea con tan solo pasar el indicador de color rojo por encima del botón con el icono de música.

Al desarrollar esta aplicación se tendrá en cuenta algunos factores técnicos tales como:

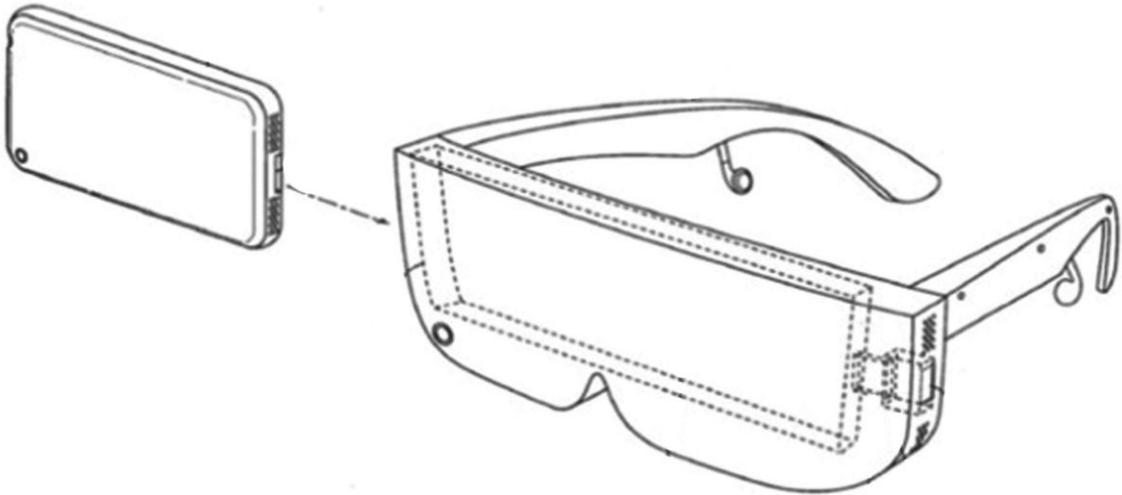
- Desarrollo de una aplicación de realidad virtual completa en Unity 3D
- Interfaces de todas las pantallas
- Ayudante, que en este caso será un oficial de tránsito
- Al final la aplicación será compilada para la plataforma Android (APK)
- Tal y como se ha descrito con anterioridad, los estadios de diseño de una interfaz se dividen en tres partes esenciales:

- Decidir qué canales externos se comunicarán con los internos y en qué forma lo harán: lo que se conoce como mapeo
- Establecer los elementos que actuarán de enlace en el exterior de la aplicación: las interfaces físicas o interfaces de hardware
- Determinar los elementos que actuarán de enlace en el interior de la aplicación: las interfaces lógicas o interfaces de software.

### 3.3 Especificaciones funcionales

Para poder hacer uso de las gafas con nuestra aplicación se debe seguir los siguientes pasos:

1. Coger las gafas de realidad aumentada y un celular Android que ya contenga la aplicación instalada.
2. Insertar el celular en el case de realidad virtual.



*Ilustración 8 Insertar celular en case*

3. Al abrir la aplicación, se observará la pantalla de menú con 3 botones: Play (para poder comenzar el juego), Test (será la evaluación) y Créditos



Ilustración 9 Menú aplicación

4. Luego se coloca las gafas en su cabeza y las ajusta de manera que se sienta cómodo.

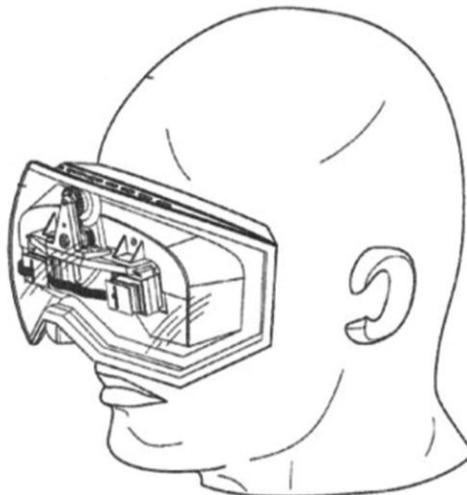


Ilustración 10 Case

5. El usuario podrá escoger el tipo de vista con el que desea realizar el recorrido ya sea desde la vista superior del auto o la vista dentro del auto



Ilustración 11 Selección de vistas de la aplicación

6. Podría observar el seguimiento desde afuera del auto mientras recorre por las diferentes señales de tránsito.



Ilustración 12 Vista Aérea

7. En la segunda vista el recorrido será desde el interior del auto



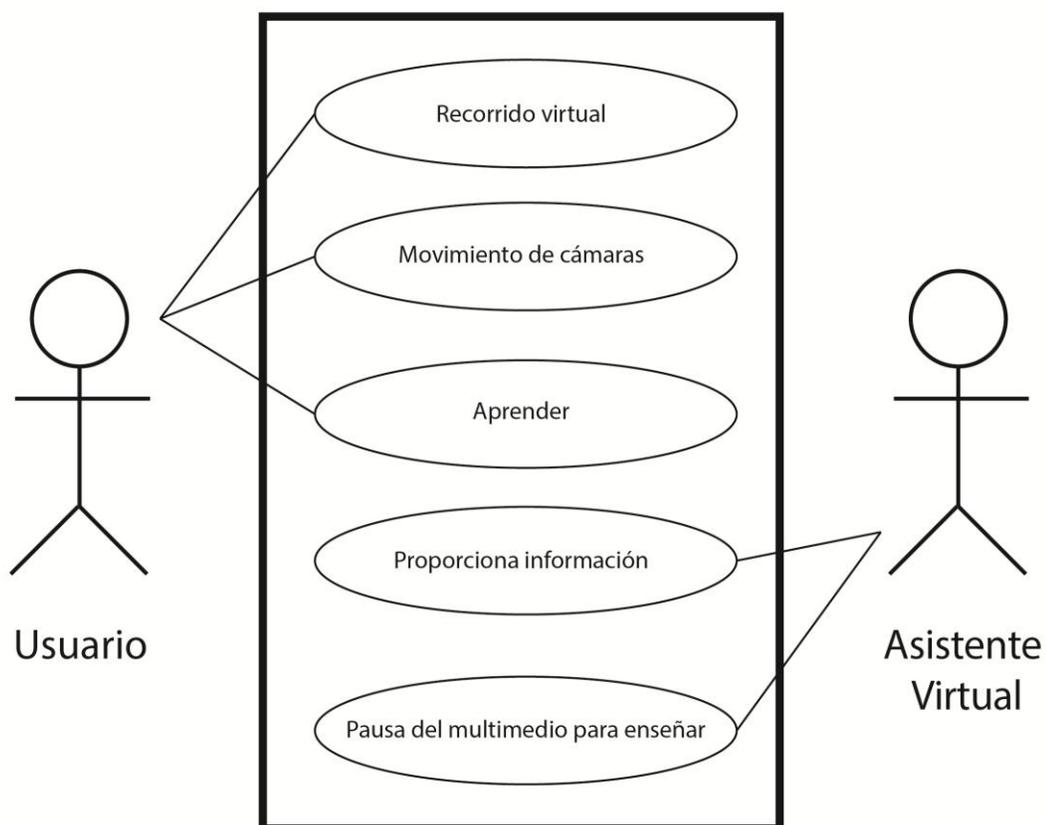
*Ilustración 13 Vista dentro del vehículo*

8. El asistente virtual le explicará en cada esquina lo que significa cada señal, usando el punto rojo que observa en la mitad de la pantalla llamado Raycast.



*Ilustración 14 Vista de la señal de tránsito a explicar*

## Sistema de enseñanza de las señales de tránsito con el uso de la realidad virtual



*Ilustración 15 Sistema de enseñanza*

## 3.4 Módulos de la Aplicación

### 3.4.1 Módulo del hardware

Este módulo sirve para preparar los componentes que se necesitan para poder visualizar la aplicación. En este proyecto solo utilizaremos los siguientes: un casco o gafas de realidad virtual y un teléfono celular que contenga el sistema operativo Android, de preferencia que el celular posea pantalla grande como los Samsung S4, S5, Note 3.

Ambos accesorios se los junta adaptando el casco al teléfono celular. Y antes de eso ya se debió de haber instalado nuestra aplicación en su celular.

#### *Descripción de las gafas virtuales:*

- Este artículo tiene las piezas magnéticas para que pueda controlar como el cartón 3D Google.
- En comparación con el producto en general, el tamaño de la caja de los vidrios es más grande, se aplica a la longitud total del teléfono es 16,5 cm, el ancho es de 9 cm
- Las gafas 3D se hace de materiales ABS y lente de resina esférica sin lámina de plástico estimulación que es amigable con el medio ambiente, completamente cerrado diseñado;
- Circunferencia principal ajustable;
- Conveniente para el teléfono móvil de la pantalla 3.5 ~ 6.5 pulgadas;
- Imágenes secretas, otras personas no pueden ver nada, sino para el propio usuario;
- Usted no sentirá fatiga visual y mareos incluso utiliza mucho tiempo w / la lente de la resina;
- Imagen 3D del efecto de pantalla es super bueno, sensación impresionante;
- Jefe circunferencia: 58cm (ajustable), conveniente para 3.5 ~ 6.5 "teléfonos inteligentes.
- Dimensiones: 6,8 x 4,72 en en x 3,74 en (17,3 cm x 12 cm x 9,5 cm)
- Peso: 225 g
- Color: Negro



*Ilustración 16 Hardware de case y teléfono celular*

### 3.4.2 Módulo del asistente virtual – Aprendizaje

Este módulo sirve para ayudar al usuario a poder mantenerse orientado durante el recorrido y que vaya recordando el significado de cada señal de tránsito que se la va presentando en cada esquina de la ciudad para que así se le pueda mantener en su mente y no olvidarlo.

El ayudante no es nada más que un Policía de Tránsito que les irá guiando, haciendo el papel de una simulación de un vigilante de tránsito que nos van guiando en nuestro camino y así poder obtener una calificación.

Luego, mediante vaya realizando el recorrido y avanzando dentro de la aplicación, el ayudante lo va deteniendo al usuario las veces necesarias cuando le aparezca una señal de tránsito en lo que dure hasta llegar al test.

Y al final de todo, el usuario deberá encontrarse preparado para el test final que se le mostrará al usuario al final del recorrido.



*Ilustración 17 Asistente virtual*

### 3.5 Especificaciones Técnicas

Para la elaboración de nuestra aplicación de educación vial, utilizamos el programa de escritorio llamado *Unity 3D: Game Engine*; pero ¿Qué es Unity 3D?

Unity es un motor para crear videojuegos: que contiene un poderoso motor de renderizado totalmente integrado con un conjunto completo de herramientas intuitivas y flujos de trabajo rápido para crear contenido 3D interactivo; y es utilizada para desarrolladores independientes quienes aman crear juegos.

Unity 3D soporta tres lenguajes de programación, que son: C#, JavaScript and Boo; aunque de los tres mas se utiliza *JavaScript* para así poder crear juegos que enganchan y deleitan a los jugadores en cualquier plataforma.

Una parte muy importante para poder realizar el recorrido virtual fueron los requerimientos del sistema que utilizamos:

- Modelados 3D ----- <http://tf3dm.com>

En esa página de internet pudimos descargar los recursos 3D que utilizamos para poder desarrollar el entorno de nuestra ciudad para el recorrido de nuestra aplicación.

- Transformar texto en audio ----- <http://vozme.com>

En esta página utilizamos su transformador de texto en audio para así poder obtener los audios que necesitaríamos en mp3 para nuestra aplicación cuando el ayudante le hable al usuario.

- Necesitaríamos un teléfono celular con sistema operativo Android.

Necesitaríamos un teléfono celular bajo ese sistema operativo ya que ese es el escogido para llevarse a cabo nuestra aplicación. Utilizaremos un Samsung S5.

Descargar una SDK para poder convertir la aplicación de Unity en el formato adecuado para los teléfonos android.

- Photoshop

Utilizamos ese programa para poder editar las texturas que necesitaríamos

- Cinema 4D

Utilizamos ese programa para poder realizar algunos modelados y para poder quitar en algunos modelados muy pesados algunos nodos demás.

Por último en las especificaciones técnicas utilizamos tres algoritmos especiales:

*(Véase el código en los Anexos)*

- *DOBLE PANTALLA*
- *GIROSCOPIO*
- *RAYCAST (LÁSER ROJO)*

### 3.6 Funciones del Aplicativo

La aplicación móvil consta con varios botones que sirven para distintas funciones como:

En el menú cuenta con 3 botones;

1. Es el botón PLAY , con el que el jugador está pidiendo comenzar el recorrido.



*Ilustración 18 Botón PLAY*

2. El botón de TEST, es la evaluación que se le hace al usuario al finalizar el recorrido.



*Ilustración 19 Botón TEST*

3. Es el botón de CRÉDITOS, en el que se observará los nombres de todas las partes involucradas en la realización del proyecto.



*Ilustración 20 Botón CRÉDITOS*

El usuario tendrá una luz roja parecida a un láser con él, al apuntar a objetos como las señales de tránsito se le aparecerá en la pantalla al usuario las indicaciones y para qué sirve de esa cada señal que apunte.

## **3.7 Descripción del usuario**

### **3.7.1 Usuario**

El rol que cumple el usuario es seguir un recorrido virtual donde se le mostrará las señales de tránsito, al hacer el recorrido el podrá hacer uso del giroscopio de la aplicación y podrá observar todo a su alrededor. Y con todo ello, el usuario podrá ir recordando las señales que se le van presentando seguido del asistente que lo guiará.

### **3.7.2 Asistente virtual**

El asistente virtual, tiene como roles principales el poder guiar e instruir al usuario una vez comenzado el recorrido y así poder proporcionar la información adecuada y pertinente.

Cada vez que proporciona la información el asistente puede detener al usuario para poder explicarle, habiendo el usuario apuntado con el raycast la señal que se va mostrando en cada esquina.

## Conclusiones y Recomendaciones

### Conclusiones

Para nuestras conclusiones en nuestro proyecto de titulación, primero tuvimos que acceder a las estadísticas de la Comisión de Tránsito para observar, que existen conductores con licencia de conducir que forman parte de las estadísticas de los accidentes de tránsito.

Con esto, hemos utilizado la problemática de las *señaléticas*, es decir, hemos detectado que estos conductores habiéndose preparado para la obtención de su licencia y haber aprendido todas las señales de tránsito que les proporcionan en todas las escuelas de conducir, existen conductores que con el tiempo tienden a olvidarse debido a que no existe un entrenamiento posterior.

A pesar de que existen señales muy comunes como el pare, el doble vía, aproximación de vigilante acostado o el prohibido estacionar; el usuario las tiene muy presente debido a que las ve todos los días mientras conducen, por el contrario, existen señaléticas que se encuentran en las zonas rurales o carreteras que no son tan comunes y por ello el usuario tienden a olvidarse.

Sin embargo, en nuestra primera fase del proyecto, involucró detectar el problema que queremos resolver y en la segunda fase se tuvo que realizar una encuesta que nos evidencia que efectivamente nuestra hipótesis, de que los usuarios que tienen licencia tienden a ir olvidando ciertas señales de tránsito. Eso se comprobó con una encuesta que realizamos a 100 personas de las cuales pudimos comprobar que de las 7 preguntas que realizamos en dicha encuesta, los usuarios respondieron 5 preguntas incorrectamente y 2 preguntas correctamente, con ello observamos que la mayoría de las señales que se encontraban en la encuesta desconocían para que servían dichas señales de tránsito.

Estas señales fueron seleccionadas por nosotros, en función de las señales que presumíamos el usuario iba a tender a olvidarlas, es decir en la encuesta no están las señales típicas del pare y gire en U, ya que es una señal muy común y obvia para cualquiera. Con el fin de probar si el usuario las reconocía o no,

nuestra encuesta arrojó a favor de nuestro proyecto que efectivamente hay un alto porcentaje de personas que no recordaban el porqué de dichas señales de tránsito.

Un tema importante sobre las señales es que éstas deberían ser obvias, muy fáciles para que las personas no necesiten estudiarlas y aprenderlas, debería ser tan obvias que incluso, para las personas que no tengan licencia deberían poder entenderlas a simple vista.

Partiendo con esa información, nosotros obtenemos nuestro planteamiento del problema teniendo en cuenta que para estos conductores no hay una forma de que se los entrene de forma posterior a la obtención de su licencia.

Una vez teniendo nuestra problemática, empezamos a llevar a cabo este proyecto, en el que hemos escogido un aspecto innovador y tecnológico que está en estos momentos en su gran acogida a nivel mundial que son: LOS CASCOS DE REALIDAD VIRTUAL.

Hoy en día existen muchas empresas que realizan estos cascos de RV, como: el Oculus Rift de Facebook, Samsung Gear de Samsung, Hololens de Microsoft y Morpheus de Sony. Dan evidencia de que el tema de la RV con estos cascos logran un alto grado de inmersión el cual es muy necesario.

Con todo esto, nosotros utilizaremos el uso de un teléfono celular con sistema operativo android y el desarrollo de una aplicación móvil. Para el proyecto no hemos pensado en desarrollar un casco sino la alternativa de conseguir un case para que el celular haga de pantalla del mismo y así, sea la aplicación móvil la que realmente produzca el efecto de dividir la pantalla distribuida para cada ojo para así, poder observar el mundo virtual en el que se va a envolver el usuario.

La aplicación móvil está hecha en Unity 3D, que es nuestra herramienta más importante y principal de las que hemos utilizado, con ese programa elaboramos todo el recorrido; aparte de los demás programas que utilizamos como: photoshop, para el retoque de imágenes y de las texturas; cinema 4D, para los modelados y también hemos utilizado una serie de algoritmos para resolver en Unity la doble pantalla, el giroscopio, etc.

Es importante recalcar, que desde el punto de vista del desarrollo de la aplicación móvil Unity no cuenta con las opciones para que se nos facilite el dividir las pantallas, vincular el giroscopio con el control del input de entrada que tendrá el usuario y demás algoritmos que se tuvo que ir investigando,

desarrollando y programando en Unity con su lenguaje predefinido que es Javascript. Y así, poder probarlo en el móvil.

Los usos de nuestra aplicación pueden ser: para que el usuario parcialmente realice esta dinámica para que no las vaya olvidando a lo que vaya transcurriendo el tiempo y también sirve para las personas que les vence su licencia y quieren prepararse de nuevo para poder renovarla.

Por otra parte hablando de la aplicación en sí, es muy importante hablar de la inmersión porque hablando de los primeros casco de realidad virtual el problema es que siempre tenían una proyección hacia delante en una pantalla, la misma que distribuye la misma imagen hacia los dos ojos, con ello el usuario al ver los bordes negros a los lados (izquierda, derecha, arriba y abajo) no logra un alto grado de inmersión que podamos evidenciar.

Hace dos años aproximadamente sale a la palestra otra vez, el tema de los cascos de realidad virtual y el abanderado Oculus Rift; cuando uno de los grandes programadores de videojuegos como John Carmack pasa a formar parte del proyecto de Oculus Rift creando Dumb 3 para que funcione con estos cascos.

Si se analiza el casco de Oculus Rift evidenciamos que ellos cubren toda el área periférica de los ojos y distribuyen dos imágenes; una para cada ojo. Con eso se logra una alta tasa de inmersión, en el que podemos observar en algunos videos de youtube donde vemos a usuarios probando este casco en las versiones beta que existen; donde los usuarios se marean, pierden el equilibrio, y hasta se evidencia que el usuario de verdad cree que se encuentra dentro del mundo virtual; mundos que son desde montañas rusas y hasta los mismo videojuegos que se han podido portar a estos cascos.

Algo interesante de todo esto, es cuando Samsung entra en escena y ellos le piden a la empresa de oculus Rift que les proporcionen no un casco sino un case para poder meter al Samsung Galaxy Note 3, Samsung aprovecha la pantalla, los giroscopio que tiene el celular y el mismo hecho de que se puede realizar la aplicación para el celular .

Y así, nosotros hemos decidido irnos por ese lado, usar un case, y ya que tenemos el conocimiento de realizar una aplicación móvil hemos seleccionado Unity 3D, logrando acabo usar el mismo efecto que Oculus Rift tiene de dividir la pantalla, también logramos el efecto de que los giroscopios sigan el

movimiento dentro del mundo 3D (arriba, abajo, izquierda, derecha) para construir un multimedia que enseñe a través de un asistente las señales que según nuestra encuesta los usuarios tienden a olvidarse; esto es un software no solo de entrenamiento sino de aprendizaje.

Y para todo esto hemos tenido en cuenta, que según los datos de la INEC, los celulares muestran que en el 2013 se registro un 86,4% de los hogares posee al menos un teléfono celular y el 16,9% son Smartphone. De tal manera que para los usuarios les es más asequible el poder comprar la aplicación con el case y con su celular poder ir entrenando en sus propias casas, sin tener que ir a ningún centro de entrenamiento.

Finalmente el multimedia trata básicamente en hacerles un recorrido virtual al usuario una vez que se coloca el case con el celular, les aparecerá un asistente virtual que es un policía le ira guiando y mostrando las señales de tránsito y el para que sirve; el usuario por medio del raycast (punto rojo) podrá seleccionar las señales o botones que se aparecerá y; junto con el case el usuario tendrá la inmersión de alto grado, que el usuario creará que se encuentra dentro del mundo virtual 3D.

## Recomendaciones

- Es necesario hacer una fase 2, fase que se realice otra encuesta a los usuarios que ya hayan utilizado la aplicación para poder medir el porcentaje de si las recuerdan o no.
- La aplicación se la aporte en los cascos ya brevemente explicado que son como el Oculus Rift, ya que tiene una pantalla mucho más grande logrando más inmersión.
- También sería importante recomendar que se aporte para los distintos sistemas operativos, ya que nuestra aplicación esta disponible solo para Android y sería bueno que también exista para iOS, Windows phone, Blackberry. O para Pc o Mac que son computadores de escritorio o portátiles.
- Finalmente sería bueno revisar el impacto en los accidentes de tránsito para así observar que hemos podido bajar algún porcentaje de las tasas.
- Otra recomendación, sería considerar un poco más el audio agregando audífonos.
- En la aplicación utilizamos para la voz del policía, una voz robotizada; sería bueno poder incorporar voces reales para así parecer más real aun.

## BIBLIOGRAFÍA

Baeza Valencia, D. F. (2011). Sistema Interactivo de Seguridad Vial. 7-8 págs.

Levis, D. (2006). ¿Qué es la realidad virtual. 1-3 págs.

Eastgate, R. (1996): Virtual Reality for Industrial Applications. Opportunities and Limitations. Nottingham University Press, Nottingham., 166 págs.

Volkswagen. (2009). Index: The Fun Theory. Retrieved Septiembre 2, 2010, from The Fun Theory

Lab, S. P. (2007). Index: Stanford Persuasive Technology Lab

Aukstalkanis, S. y Blatner, D. (1993) : El espejismo de silicio. Arte y ciencia de la realidad virtual. 282 págs.

Bayarri, S. (2005). Modelado de Carreteras. 1-6 págs.

Biocca, F. y Levy, M. (1995): "Virtual reality as a Communication System" en Communication in the age of Virtual Reality. 15-31 págs.

Pimentel, K, y Texeira, K. (1992) Virtual Reality. Through the New Looking Glass, Intel/McGraw-Hill New York.; 2ªEdic.1995, 438 págs.

Unity 3d. (2009). Motor gráfico para desarrollo de entornos 3d. ¿Que es unity ? , <http://unity3d.com/es>.

Dive. (2013). SDK para desarrollo de Realidad Virtual. What is Dive?, <http://www.durovis.com/index.html>.

## ANEXOS

### A) Encuesta realizadas a personas en la ciudad de Guayaquil

#### Encuesta

Esta encuesta trata sobre Realidad Virtual y señáleticas de tránsito

##### Género

- Femenino
- Masculino

##### Edad

##### ¿TIENE AUTO?

- Si
- No

##### ¿QUÉ TIPO DE LICENCIA POSEE?

##### ¿SABE CUANTOS PUNTOS TIENE EN SU LICENCIA DE CONDUCIR?

- Si
- No

##### INDIQUE PARA QUE SIRVEN ESTAS SEÑALÉTICAS.



- Proximidad de un resalto
- Proximidad de una irregularidad física
- Proximidad de una pendiente física

Ilustración 21 Encuesta



- Proximidad de calle que cruza
- Proximidad de una intersección
- Proximidad de cruces



- Prohibido estacionar
- Super prohibido estacionar
- Prohibición de estacionar y/o detener un vehículo



- Proximidad de una separación en un mismo sentido
- Proximidad de una separacion en dos sentidos



- Proximidad de vivienda
- Proximidad de ciudad
- Proximidad de refugio



- Sitio favorito
- Sitio histórico
- Proximidad de comisaria



- Retorno de carretera
- Girar en U

#### ¿QUÉ TIPO DE REALIDAD VIRTUAL UD CONOCE?

- Televisión
- Gafas virtuales
- Simuladores
- X-box/Kinect
- Cine

#### ¿UD POSEE UN TELÉFONO CELULAR?

- Si
- No

#### ¿UD HA SUFRIDO ALGUN TIPO DE ACCIDENTE AUTOMOVILISTICO?

- Si
- No

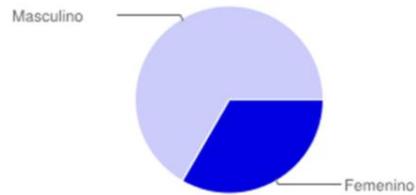
#### ¿PORQUE MOTIVO FUE SU ACCIDENTE?

- Pasarse una roja
- No respetar zona cebra
- No respetar un Pare
- Me chocaron
- Problemas mecánicos

Enviar

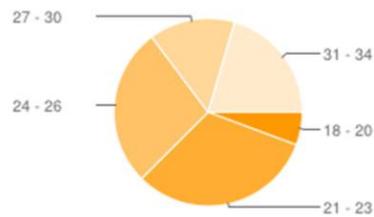
## B) Porcentajes de las respuestas de las encuestas

### Género



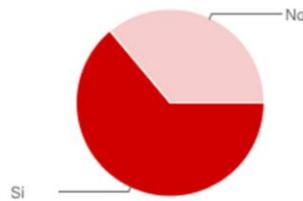
Femenino	131	34%
Masculino	252	66%

### Edad



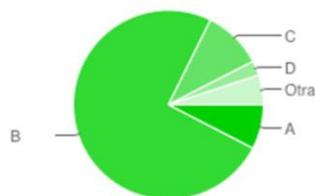
18 - 20	22	6%
21 - 23	122	32%
24 - 26	104	27%
27 - 30	57	15%
31 - 34	78	25%

### ¿TIENE AUTO?



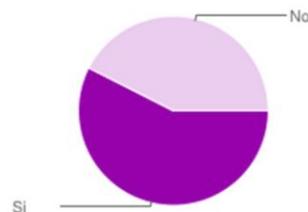
Si	239	62%
No	144	38%

### ¿QUÉ TIPO DE LICENCIA POSEE?



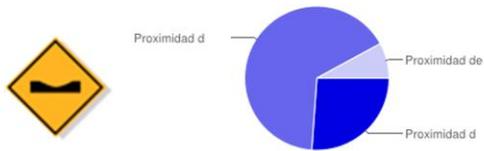
A	26	8%
B	296	77%
C	35	9%
D	9	2%
Otra	17	4%

### ¿SABE CUANTOS PUNTOS TIENE EN SU LICENCIA DE CONDUCIR?

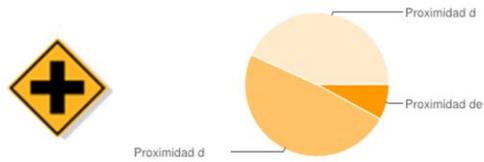


Si	218	57%
No	165	43%

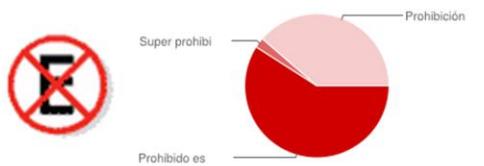
## INDIQUE PARA QUE SIRVEN ESTAS SEÑALES



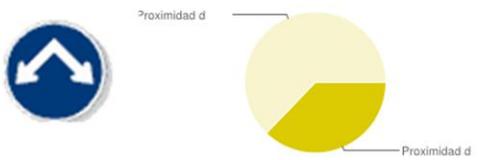
Proximidad de un resalto	100	26%
Proximidad de una irregularidad física	252	66%
Proximidad de una pendiente física	30	8%



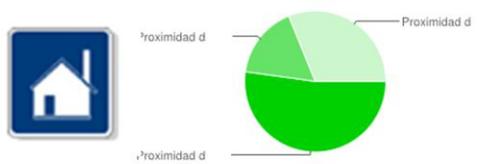
Proximidad de una calle que cruza	30	8%
Proximidad de una intersección	187	49%
Proximidad de cruces	165	43%



Prohibido estacionar	226	59%
Super prohibido estacionar	9	2%
Prohibición de estacionar y/o detener un vehículo	148	39%



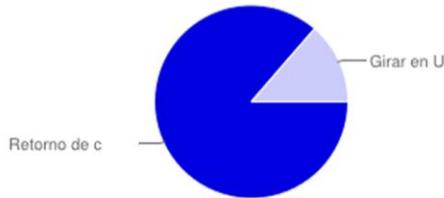
Proximidad de una separación en un mismo sentido	148	39%
Proximidad de una separación en dos sentidos	235	61%



Proximidad de vivienda	205	54%
Proximidad de ciudad	65	17%
Proximidad de refugio	113	30%

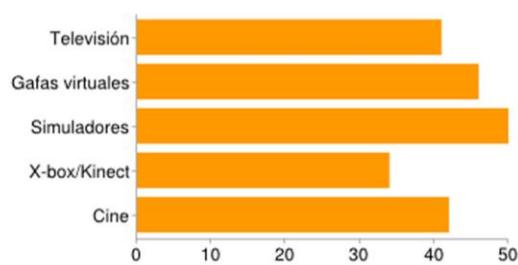


Sitio favorito	57	15%
Sitio histórico	118	31%
Proximidad de comisaria	209	55%



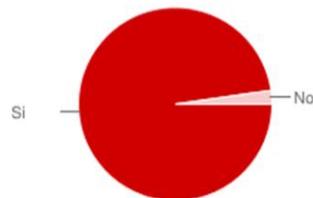
Retorno de carretera	<b>331</b>	<b>86%</b>
Girar en U	<b>52</b>	<b>14%</b>

### ¿QUÉ TIPO DE REALIDAD VIRTUAL UD CONOCE?



Televisión	<b>178</b>	<b>46%</b>
Gafas virtuales	<b>200</b>	<b>52%</b>
Simuladores	<b>218</b>	<b>57%</b>
X-box/Kinect	<b>148</b>	<b>39%</b>
Cine	<b>183</b>	<b>48%</b>

### ¿UD POSEE UN TELÉFONO CELULAR?



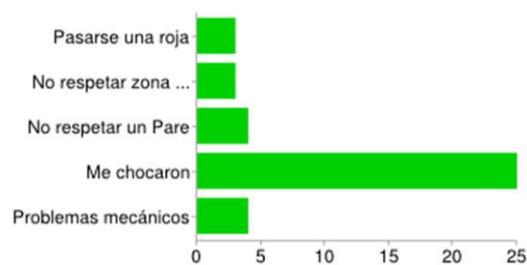
Si	<b>374</b>	<b>98%</b>
No	<b>9</b>	<b>2%</b>

### ¿UD HA SUFRIDO ALGUN TIPO DE ACCIDENTE AUTOMOVILISTICO?



Si	<b>131</b>	<b>34%</b>
No	<b>252</b>	<b>66%</b>

### ¿PORQUE MOTIVO FUE SU ACCIDENTE?



Pasarse una roja	<b>13</b>	<b>3%</b>
No respetar zona cebra	<b>13</b>	<b>3%</b>
No respetar un Pare	<b>17</b>	<b>4%</b>
Me chocaron	<b>109</b>	<b>28%</b>
Problemas mecánicos	<b>17</b>	<b>4%</b>

## C) Código que debimos resolver

### *DOBLE PANTALLA*

```
using UnityEngine;
using System.Collections;
using System.Runtime.InteropServices;
using System.IO;
using System;

//Dive Head Tracking
// copyright by Shoogee GmbH & Co. KG Refer to LICENCE.txt
//[ExecuteInEditMode]
public class OpenDiveSensor : MonoBehaviour {
    // This is used for rotating the camera with another object
    //for example tilting the camera while going along a racetrack or rollercoaster
    public bool add_rotation_gameobject=false;
    public GameObject rotation_gameobject;
    // mouse emulation
    public bool emulateMouseInEditor=true;
    public enum RotationAxes { MouseXAndY = 0, MouseX = 1, MouseY = 2 }
    public RotationAxes axes = RotationAxes.MouseXAndY;
    public Texture nogyrotexture;
    /// Offset projection for 2 cameras in VR
    private float offset =0.0f;
    private float max_offset=0.002f;
    //public float max_offcenter_warp=0.1f;
    public Camera cameraleft;
    public Camera cameraright;
```

```

public float zoom=0.1f;

private float IPDCorrection=0.0f;

private float aspectRatio;

public float znear=0.1f;

public float zfar=10000.0f;

private float time_since_last_fullscreen=0;

private int is_tablet;

AndroidJavaObject mConfig;

AndroidJavaObject mWindowManager;

private float q0,q1,q2,q3;

private float m0,m1,m2;

Quaternion rot;

private bool show_gyro_error_message=false;

string errormessage;

#if UNITY_EDITOR

private float sensitivityX = 15F;

private float sensitivityY = 15F;

private float minimumX = -360F;

private float maximumX = 360F;

private float minimumY = -90F;

private float maximumY = 90F;

float rotationY = 0F;

#elif UNITY_ANDROID

private static AndroidJavaClass javadivepluginclass;

private static AndroidJavaClass javaunityplayerclass;

private static AndroidJavaObject currentactivity;

private static AndroidJavaObject javadiveplugininstance;

[DllImport("divesensor")] private static extern void initialize_sensors();

```

```

    [DllImport("divesensor")] private static extern int get_q(ref float q0,ref float q1,ref float q2,ref float
q3);

    [DllImport("divesensor")] private static extern int get_m(ref float m0,ref float m1,ref float m2);

    [DllImport("divesensor")] private static extern int get_error();

    [DllImport("divesensor")] private static extern void dive_command(string command);

#ifdef UNITY_IPHONE

    [DllImport("__Internal")] private static extern void initialize_sensors();

    [DllImport("__Internal")] private static extern float get_q0();

    [DllImport("__Internal")] private static extern float get_q1();

    [DllImport("__Internal")] private static extern float get_q2();

    [DllImport("__Internal")] private static extern float get_q3();

    [DllImport("__Internal")] private static extern void DiveUpdateGyroData();

    [DllImport("__Internal")] private static extern int get_q(ref float q0,ref float q1,ref float q2,ref float q3)

#endif

    public static void divecommand(string command){

        #if UNITY_EDITOR

        #elif UNITY_ANDROID

        dive_command(command);

        #elif UNITY_IPHONE

        #endif }

    public static void setFullscreen(){

        #if UNITY_EDITOR

        #elif UNITY_ANDROID

        String answer;

        answer= javadiveplugininstance.Call<string>("setFullscreen");

        #elif UNITY_IPHONE

        #endif

        return;}

```

```

void Start () {

    rot=Quaternion.identity;

    // Disable screen dimming

    Screen.sleepTimeout = SleepTimeout.NeverSleep;

    Application.targetFrameRate = 60;

#if UNITY_EDITOR

    if (rigidbody)

        rigidbody.freezeRotation = true;

#elif UNITY_ANDROID

    // Java part

    javadivepluginclass = new AndroidJavaClass("com.shoogee.divejava.divejava");

    javaunityplayerclass= new AndroidJavaClass("com.unity3d.player.UnityPlayer");

    currentactivity = javaunityplayerclass.GetStatic<AndroidJavaObject>("currentActivity");

    javadiveplugininstance = javadivepluginclass.CallStatic<AndroidJavaObject>("instance");

    object[] args={currentactivity};

    javadiveplugininstance.Call<string>("set_activity",args);

    initialize_sensors ();

    String answer;

    answer= javadiveplugininstance.Call<string>("initializeDive");

    answer= javadiveplugininstance.Call<string>("getDeviceType");

    if (answer=="Tablet"){

        is_tablet=1;

        Debug.Log("Dive Unity Tablet Mode activated"); }

    else{

        Debug.Log("Dive Phone Mode activated "+answer); }

    answer= javadiveplugininstance.Call<string>("setFullscreen");

    show_gyro_error_message=true;

    Network.logLevel = NetworkLogLevel.Full;

```

```

int err = get_error();

if (err==0){ errormessage="";
            show_gyro_error_message=false;}

if (err==1){
            show_gyro_error_message=true;

            errormessage="ERROR: Dive needs a Gyroscope and your telephone has
none, we are trying to go to Accelerometer compatibility mode. Dont expect too much.";}

#ifdef UNITY_IPHONE
            initialize_sensors();
#endif

float tabletcorrection=-0.028f;

//is_tablet=0;

if (is_tablet==1){
            Debug.Log ("Is tablet, using tabletcorrection");
            IPDCorrection=tabletcorrection;}

else {
            IPDCorrection=IPDCorrection;}

//setIPDCorrection(IPDCorrection); }

void Update () {
            aspectRatio=(Screen.height*2.0f)/Screen.width;
            setIPDCorrection(IPDCorrection);

            //Debug.Log ("Divecamera"+cameraleft.aspect+"1/asp "+1/cameraleft.aspect+" Screen
Width/Height "+ aspectRatio);

#ifdef UNITY_EDITOR

#ifdef UNITY_ANDROID

            time_since_last_fullscreen+=Time.deltaTime;

            if (time_since_last_fullscreen >8){
                    setFullscreen ();

                    time_since_last_fullscreen=0;}

```

```

    get_q(ref q0,ref q1,ref q2,ref q3);

    //get_m(ref m0,ref m1,ref m2);

    rot.x=-q2;rot.y=q3;rot.z=-q1;rot.w=q0;

    if (add_rotation_gameobject){

        transform.rotation =rotation_gameobject.transform.rotation* rot;}

    else{

        transform.rotation = rot;

        if
(is_tablet==1)transform.rotation=rot*Quaternion.AngleAxis(90,Vector3.forward);}

#ifdef UNITY_IPHONE

    DiveUpdateGyroData();

    get_q(ref q0,ref q1,ref q2,ref q3);

    rot.x=-q2;

    rot.y=q3;

    rot.z=-q1;

    rot.w=q0;

    transform.rotation = rot;

    if (add_rotation_gameobject){

        transform.rotation =rotation_gameobject.transform.rotation* rot;}

    else{

        transform.rotation = rot;

        if
(is_tablet==1)transform.rotation=rot*Quaternion.AngleAxis(90,Vector3.forward); }

#endif

#ifdef UNITY_EDITOR

    if (emulateMouseInEditor){

        if (axes == RotationAxes.MouseXAndY){

            float rotationX = transform.localEulerAngles.y + Input.GetAxis("Mouse
X") * sensitivityX;

```

```

        rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
        rotationY = Mathf.Clamp (rotationY, minimumY, maximumY);
        transform.localEulerAngles = new Vector3(-rotationY, rotationX, 0);}
else if (axes == RotationAxes.MouseX){
    transform.Rotate(0, Input.GetAxis("Mouse X") * sensitivityX, 0);}
else{
    rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
    rotationY = Mathf.Clamp (rotationY, minimumY, maximumY);
    transform.localEulerAngles = new Vector3(-rotationY,
transform.localEulerAngles.y, 0);}}
#endif}

void OnGUI (){
    /*    if (GUI.Button(new Rect(Screen.width/4-150, Screen.height-100, 300,100), "+IPD")){
        Debug.Log("Clicked the button with an image");
        IPDCorrection=IPDCorrection+0.01f;
        setIPDCorrection(IPDCorrection);}
    if (GUI.Button(new Rect(Screen.width-Screen.width/4-150, Screen.height-100, 300,100),
"-IPD")){
        Debug.Log("Clicked the button with an image");
        IPDCorrection=IPDCorrection-0.01f;
        setIPDCorrection(IPDCorrection);}
    */

    if (show_gyro_error_message){
        if(GUI.Button(new Rect(0,0, Screen.width, Screen.height) , "Error: \n\n No Gyro
detected \n \n Touch screen to continue anyway")) {
            show_gyro_error_message=false;}

        GUI.DrawTexture(new Rect(Screen.width/2-320, Screen.height/2-240, 640,
480), nogyrotecture, ScaleMode.ScaleToFit, true, 0);

        return;}}

```

```

void setIPDCorrection(float correction) {

    // not using camera nearclipplane value because that leads to problems with field of view
    in different projects

    cameraleft.projectionMatrix = PerspectiveOffCenter((-zoom+correction)*(znear/0.1f),
(zoom+correction)*(znear/0.1f), -zoom*(znear/0.1f)*aspectRatio, zoom*(znear/0.1f)*aspectRatio, znear,
zfar);;

    cameraright.projectionMatrix = PerspectiveOffCenter((-zoom-correction)*(znear/0.1f),
(zoom-correction)*(znear/0.1f), -zoom*(znear/0.1f)*aspectRatio, zoom*(znear/0.1f)*aspectRatio, znear,
zfar);;

    static Matrix4x4 PerspectiveOffCenter(float left, float right, float bottom, float top, float near, float
far) {

        float x = 2.0F * near / (right - left);

        float y = 2.0F * near / (top - bottom);

        float a = (right + left) / (right - left);

        float b = (top + bottom) / (top - bottom);

        float c = -(far + near) / (far - near);

        float d = -(2.0F * far * near) / (far - near);

        float e = -1.0F;

        Matrix4x4 m = new Matrix4x4();

        m[0, 0] = x;

        m[0, 1] = 0;

        m[0, 2] = a;

        m[0, 3] = 0;

        m[1, 0] = 0;

        m[1, 1] = y;

        m[1, 2] = b;

        m[1, 3] = 0;

        m[2, 0] = 0;

        m[2, 1] = 0;

        m[2, 2] = c;

        m[2, 3] = d;

```

```

        m[3, 0] = 0;
        m[3, 1] = 0;
        m[3, 2] = e;
        m[3, 3] = 0;

return m; } }

```

## *GIROSCOPIO*

```

#pragma strict

:Vector3;

var inputJump=false;

var grounded=false;

var downmovement=0.1;

var velocity:Vector3;

public var max_speed=0.1;

public var max_speed_air=0.13;

public var max_speed_ground=0.1;

public var acceleration=10;

public var acceleration_air=10;

public var gravity=-0.18;

public var friction=0.8;

public var cameraObject:GameObject;

private var controller : CharacterController;

public var groundNormal:Vector3;

public var jumpspeed=0.16;

public var stopmovingup=false;

public var fallkillspeed=-0.38;

public var collisionFlags : CollisionFlags;

public var ground_gameobject:GameObject=null;

public var last_ground_pos:Vector3;

```

```

private var jumpcommand=0;

public var floating=false;

public var autowalk:int=0;

public var inhibit_autowalk=1;

public var reload_once:int=0;

function Awake () {

controller = GetComponent (CharacterController);

Debug.Log("Controller SlopeLimit"+controller.slopeLimit);}

function toggle_autowalk(){

if (autowalk==0)autowalk=1;

else autowalk=0;

function JumpUp(){

jumpcommand=1;}

function Start () {

reload_once=0;}

function OnControllerColliderHit (hit : ControllerColliderHit) {

if (hit.normal.y > 0 && hit.moveDirection.y < 0) {

groundNormal = hit.normal;

grounded=true;

ground_gameobject=hit.gameObject;

print("Landed on: ground"+ground_gameobject.name);

stopmovingup=false;}}

var fadeduration = 2.0; // fade duration in seconds

function Die(){

// create a GUITexture:

var fade: GameObject = new GameObject();

fade.AddComponent(GUITexture);

// and set it to the screen dimensions:

```

```

fade.guiTexture.pixelInset = Rect(0, 0, Screen.width, Screen.height);

// set its texture to a black pixel:
var tex = new Texture2D(1, 1);
tex.SetPixel(0, 0, Color.black);
tex.Apply();
fade.guiTexture.texture = tex;

// then fade it during duration seconds
for (var alpha:float = 0.0; alpha < 1.0; ){
    alpha += Time.deltaTime / fadeduration;
    fade.guiTexture.color.a = alpha;
    yield; }

// finally, reload the current level:
if (reload_once==0){
    reload_once =1;
    var async : AsyncOperation = Application.LoadLevelAsync(Application.loadedLevel);
    yield async;}}

function Update () {
if (velocity.y < fallkillspeed)Die();

Debug.DrawLine (transform.position, transform.position+groundNormal, Color.red);
//print("GroundNormal y" +groundNormal.y);

var directionVector = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));
if (autowalk==1)directionVector=Vector3(0,0,1*inhibit_autowalk);
if (directionVector != Vector3.zero) {
        // Get the length of the directon vector and then normalize it
        // Dividing by the length is cheaper than normalizing when we already have the length anyway
var directionLength = directionVector.magnitude;
        directionVector = directionVector / directionLength;
        // Make sure the length is no bigger than 1

```

```

directionLength = Mathf.Min(1, directionLength);

// Make the input vector more sensitive towards the extremes and less sensitive in the middle
// This makes it easier to control slow speeds when using analog sticks

directionLength = directionLength * directionLength;

// Multiply the normalized direction vector by the modified length
directionVector = directionVector * directionLength;}

// Apply the direction to the CharacterMotor
inputMoveDirection = directionVector;

inputJump = Input.GetButton("Jump");

if(jumpcommand==1){

    inputJump=true;

    jumpcommand=0;}

    grounded=(collisionFlags & CollisionFlags.Below);

    if (floating){

        velocity.y=0.1;}

if (inputJump&&grounded){

velocity.y+=jumpspeed;

velocity.z*=1.5;

    //print ("Jump");

    grounded=false;}

    if (grounded){

        velocity+=inputMoveDirection*acceleration*Time.deltaTime;}

    else{

        velocity+=inputMoveDirection*acceleration_air*Time.deltaTime;}

    var translation=Vector3(velocity.x,0,velocity.z);

    translation=Vector3.Lerp(Vector3(0,0,0),translation,friction);

    velocity.x=translation.x;

    velocity.z=translation.z;

```

```

velocity.y = velocity.y+gravity*Time.deltaTime;

//print ("Time deltatime "+Time.deltaTime);

if (grounded)velocity.y=0;

//print ("Vel y "+velocity.y);

if (grounded)max_speed=max_speed_ground;

if (!grounded)max_speed=max_speed_air;

if (translation.magnitude>max_speed){

translation=translation/translation.magnitude;

translation=translation*max_speed;}

velocity.x=translation.x;

velocity.z=translation.z;

translation.y=velocity.y;

var yrotation_camera= Quaternion.Euler(0, cameraObject.transform.rotation.eulerAngles.y, 0);

//transform.position+=yrotation_camera*translation;

var platformdelta:Vector3;

if(ground_gameobject!=null) {

//print("ground object ungleich null");

platformdelta=ground_gameobject.transform.position-last_ground_pos;}

//if (!grounded)platformdelta=Vector3.zero;

//MAKE A MOVE!

collisionFlags=controller.Move(yrotation_camera*translation+platformdelta);

if (collisionFlags & CollisionFlags.CollidedAbove){

if (stopmovingup==false){

print ("ControllerColliderHit");

velocity.y=0;

stopmovingup=true;}}

if (ground_gameobject!=null && !grounded){

ground_gameobject=null;}

```

```

        if(ground_gameobject!=null)last_ground_pos=ground_gameobject.transform.position;    }

function LateUpdate () {

//groundNormal=Vector3.zero;}

function OnGUI () {

//      GUI.Label (Rect (10, 10, 400, 20), "isgrounded "+grounded+      " velocity.y "+velocity.y+"
collisionbelow "+ (collisionFlags & CollisionFlags.Below) );

var e : Event = Event.current;

    if (e.isKey) {

        Debug.Log("Detected key code: " + e.keyCode);

            GUI.Label (Rect (10, 10, 400, 20), "Keycode =" + e.keyCode );} }

function OnTriggerEnter( other : Collider ) {

if (other.name == "Float") {

floating=true;

inhibit_autowalk=0.1;}}

function OnTriggerExit( other : Collider ) {

if (other.name == "Float") {

floating=false;

inhibit_autowalk=1;}

}@script RequireComponent (CharacterController)

```

## *PUNTO ROJO*

```

private var motor : CharacterMotor;

// Use this for initialization

function Awake () {

motor = GetComponent(CharacterMotor);

}

// Update is called once per frame

function Update () {

```

```

// Get the input vector from keyboard or analog stick
var directionVector = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));
if (directionVector != Vector3.zero) {
    // Get the length of the direction vector and then normalize it
    // Dividing by the length is cheaper than normalizing when we already have the length anyway
    var directionLength = directionVector.magnitude;
    directionVector = directionVector / directionLength;
    Make sure the length is no bigger than 1
    directionLength = Mathf.Min(1, directionLength);
    // Make the input vector more sensitive towards the extremes and less sensitive in the middle
    // This makes it easier to control slow speeds when using analog sticks
    directionLength = directionLength * directionLength;
    // Multiply the normalized direction vector by the modified length
    directionVector = directionVector * directionLength;
}
// Apply the direction to the CharacterMotor
motor.inputMoveDirection = transform.rotation * directionVector;
motor.inputJump = Input.GetButton("Jump");
}
// Require a character controller to be attached to the same game object
@script RequireComponent (CharacterMotor)
@script AddComponentMenu ("Character/FPS Input Controller")

```