



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

JUEGOS DIDÁCTICOS CON REALIDAD
AUMENTADA PARA MATEMÁTICAS UTILIZANDO
EL SISTEMA OPERATIVO ANDROID

TESIS

Que para obtener el título de
INGENIERO EN COMPUTACIÓN

PRESENTA:

DAVID RICARDO MONTALVÁN RODRÍGUEZ

DIRECTOR DE TESIS:

M.E. ROSALBA RODRÍGUEZ CHÁVEZ



CIUDAD UNIVERSITARIA, CD. MX., 2016



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**JUEGOS DIDÁCTICOS CON REALIDAD
AUMENTADA PARA MATEMÁTICAS UTILIZANDO
EL SISTEMA OPERATIVO ANDROID**

Agradecimientos

A mi familia.

A la Universidad Nacional Autónoma de México que me ha abierto las puertas del conocimiento y me ha otorgado el placer de formarme como estudiante y como persona, dentro de sus aulas.

Reconocimiento

Este trabajo se realizó gracias al apoyo de la Dirección General de Asuntos del Personal Académico (DGAPA) de la Universidad Nacional Autónoma de México (UNAM), a través del Programa de Apoyo a Proyectos para la Innovación y Mejoramiento de la Enseñanza (PAPIME) por medio del proyecto PE104416 “Ambientes Virtuales y Herramientas Digitales para Neurociencias”.

Se agradece también al proyecto PE104415 “Drupal Camp y Web Academy”, del mismo Programa de Apoyo a Proyectos para la Innovación y Mejoramiento de la Enseñanza (PAPIME) por el apoyo brindado para el desarrollo de esta tesis.

Resumen

En la Facultad de Ingeniería de la UNAM se ha registrado en los últimos años, un rezago estudiantil dentro de los primeros semestres de la carrera, así como un déficit de conocimientos previos en Matemáticas. Es por ello que la misma Facultad, a través de la División de Ciencias Básicas, se ha dado a la tarea de proponer soluciones a este problema, una de esas propuestas es la integración de juegos didácticos que ayuden al alumno a entender de una manera más amena los temas que se han visto en clases.

Contenido

1. Introducción.....	9
1.1 Las TIC como apoyo educativo.....	11
1.2 Objetivo de la propuesta	13
1.3 Definición del problema	13
1.4 Rezago estudiantil en la FI	17
1.5 Propuestas de la DCB para solucionar el rezago	20
2 Las TIC en la educación.....	23
2.1 Juegos didácticos en las Matemáticas	24
2.2 M-learning en la educación.....	26
2.2.1 El rol del profesor y el rol del alumno en M-learning	27
2.3 Plataformas Educativas.....	29
2.4 Software educativo.....	30
2.5 Aplicaciones Android con realidad aumentada para la enseñanza- aprendizaje de Matemáticas	32
2.5.1 Aplicaciones Android para el apoyo de Matemáticas.....	33
2.5.2 Realidad Aumentada con Android.....	36
2.5.3 Elementos necesarios para la Realidad Aumentada.....	37
3. Desarrollo de una aplicación de Realidad Aumentada para Android	40
3.1 Requerimientos.....	41
3.2 Metodologías para la Ingeniería de software.....	41
3.3 Herramientas de software para la Realidad Aumentada	47
3.3.1 Realidad Aumentada	47
3.3.2 Vuforia y el reconocimiento de imágenes.....	49
3.3.3 Unity y Realidad Aumentada	50
3.3.4 Blender y el modelado en 3D	51
3.3.5 XAMPP para la base de datos	53
3.4 Construcción del Juego de Realidad Aumentada.....	54

3.4.1 Escena uno: Menú de inicio.....	56
3.4.2 Escena dos: Juego de Superficies	64
3.4.3 Escena tres: Juego de Secciones Cónicas	73
3.4.4 Escena cuatro: Juego de Cálculo de Volumen Máximo.....	81
3.5 Base de datos del Juego de Realidad Aumentada.....	88
3.6 Conexión con la Base de Datos.....	92
3.7 Estrategias para el uso de plantillas de juegos lúdicos.....	100
3.7.1 Estrategia para el juego de superficies.....	101
3.7.2 Estrategia para el juego de secciones cónicas	102
3.7.3 Estrategia para el juego de cálculo de volumen máximo.....	103
4. Resultados.....	105
5. Conclusiones.....	108
6. Referencias Bibliográficas	112

1. Introducción

La Facultad de Ingeniería promueve la creación de diversos recursos para apoyar el proceso enseñanza-aprendizaje, dichos recursos ayudarán a reforzar los temas vistos en diversas asignaturas. Algunos de los recursos didácticos que se desarrollan en la Facultad han encontrado un gran apoyo en las nuevas Tecnologías de la Información y la Comunicación, TIC, ya que permiten al estudiante adquirir conocimientos y habilidades de manera más práctica y, de alguna forma, abren el panorama hacia los nuevos métodos de aprendizaje basados en el surgimiento de estas tecnologías.

La Organización de las Naciones para la Educación, la Ciencia y la Cultura, UNESCO, señala que las TIC pueden incrementar las oportunidades de educación al permitir superar las barreras geográficas; apoyando sobre todo la educación a distancia, reformando el proceso de enseñanza-aprendizaje y desarrollando en la población habilidades tecnológicas tales como: acceso, evaluación y organización de la información en entornos digitales, así como su modelación y transformación en nuevo conocimiento o su uso como fuente de nuevas ideas, entre otros. (Rangel, Revista Digital Universitaria, 2013)

Las TIC tienen el potencial de transformar los procesos enseñanza-aprendizaje de manera innovadora como apoyo de las formas tradicionales y no tradicionales, además de fomentar un modelo centrado en el estudiante, y el apoyo en diversas estrategias como las de trabajo colaborativo, también favorecen el desarrollo de proyectos de investigación, los cuales se derivan en aprendizajes más reflexivos, profundos y participativos.

Durante este trabajo se tratan temas como el rezago escolar en la Facultad de Ingeniería y las soluciones que propone la misma Facultad para afrontar este problema, también se ve la integración de las nuevas TIC en distintos campos de la sociedad pero sobre todo, lo importante que ha resultado el uso de estas tecnologías en el campo educativo, por último se describe el desarrollo de plantillas para juegos de realidad aumentada como apoyo a las Matemáticas y a las propuestas de la Facultad para tales problemas.

1.1 Las TIC como apoyo educativo

En el estudio de la ingeniería existe una variedad enorme de herramientas para su estudio y comprensión, y una de las más importantes son las Matemáticas. En la Facultad de Ingeniería de la Universidad Nacional Autónoma de México existe una preocupación por el nivel de conocimientos y habilidades en Matemáticas de sus alumnos. Y es que en los últimos años ha aumentado el índice de rezago y reprobación en las generaciones de primer ingreso de esta Facultad. (Gómez, 2016)

Dentro de la División de Ciencias Básicas, DCB, de la misma Facultad, se han planteado diversos métodos y estrategias didácticas de apoyo para el estudiante, métodos que buscan reforzar y nivelar el conocimiento y, al mismo tiempo, disminuir el rezago estudiantil dentro de los primeros semestres. (Gómez, 2016)

Existen diversas formas de ayudar a reforzar y nivelar el conocimiento adquirido en clase, desde series de ejercicios hasta laboratorios virtuales, estos últimos, son parte de las nuevas tecnologías de la información y la comunicación, TIC, las cuales han venido ganando terreno en el campo de la enseñanza en los últimos años (Rangel, 2013).

Como parte de las TIC, *M-learning*, *movil learning*, o aprendizaje móvil, surge como una de las nuevas herramientas digitales, la cual permite que el estudiante desarrolle habilidades cognitivas, ya que este tipo de herramientas permite que el conocimiento tenga mayor alcance en generaciones que han adoptado la tecnología móvil como una forma de vida (UPM, 2013)

Con ayuda de las nuevas TIC, se pueden crear aplicaciones móviles portátiles, las cuales permiten una comunicación más cercana entre estudiantes, contando también con una realimentación inmediata, y en varias ocasiones, el aprendizaje con ayuda de los dispositivos móviles es más efectivo y se tiene la gran ventaja de que dichas aplicaciones pueden ser más personalizadas, y tener la información en tiempo y forma.

El presente trabajo busca apoyarse de estas nuevas tecnologías digitales para desarrollar herramientas que ayuden al alumno a tener una mejor retención de conocimientos y mejorar sus habilidades Matemáticas; basados en la idea de que el juego es la manera más natural de aprender nuevo conocimiento (González, 2008), de ahí que se pretenda crear una aplicación en forma de juego didáctico, en donde el estudiante pueda aprender, o reforzar, temas necesarios para su formación como estudiante de ingeniería.

Con los juegos desarrollados se pretende apoyar al alumno a mejorar sus conocimientos y habilidades de las materias de ciencias básicas, así como

también ayudar al profesor brindándole una herramienta que pueda utilizar para mejorar tanto la enseñanza como la administración de la misma, en cuanto a evaluaciones se refiere, de sus grupos.

Más allá de que este material didáctico ayude a disminuir los índices de rezago y reprobación en la Facultad de ingeniería, se busca que el alumno se apoye en otra herramienta más para reforzar sus conocimientos, y al mismo tiempo, hacerle saber que las nuevas Tecnologías de la Información y la Comunicación pueden ser utilizadas para la ayuda y la mejora del proceso de enseñanza-aprendizaje.

1.2 Objetivo de la propuesta

Implementar diversas plantillas para juegos didácticos en donde los profesores puedan incorporar diversos ejercicios de Matemáticas para que los alumnos puedan competir entre ellos. Dichas plantillas serán utilizadas en el desarrollo de juegos para dispositivos móviles.

1.3 Definición del problema

En la Facultad de Ingeniería de la UNAM, se tiene la tarea de que los alumnos obtengan una formación integral, la cual incluye la asimilación de conocimientos científicos y técnicos basada en el dominio de herramientas Matemáticas y de fenómenos físicos y químicos (COPADI- FI, 2016). Es por lo anterior que el

estudiante de ingeniería necesita, en todo momento, enriquecer su abanico de conocimientos.

La capacidad con la que contará el estudiante de ingeniería para poder afrontar, manejar y solucionar problemas, la adquirirá, en gran parte, en el estudio de fenómenos físicos y químicos desde una perspectiva matemática; hay que tomar en cuenta que las Matemáticas son una base fundamental en la formación de ingenieros. (Balderrama, 2003)

Además las Matemáticas aplicadas son las que han permitido lograr el desarrollo que ha alcanzado la ingeniería, éstas son el medio más poderoso que tiene el hombre para comprender al mundo que le rodea (Balderrama, 2003), y la comprensión es una habilidad de la cual un ingeniero no puede prescindir, hay que recalcar que es importante comprender para resolver.

Constantemente, el alumno, debe identificar y aplicar diferentes estrategias de aprendizaje, las cuales tendrán que facilitar la manera en la que el estudiante obtiene el conocimiento, de lo contrario será complicada dicha adquisición y eso reducirá la capacidad que tenga el estudiante para resolver los problemas.

Un déficit de conocimientos previos en los estudiantes es un problema grave, sobre todo si ese problema se viene acarreado de niveles anteriores, en nuestro caso, si existe una mala calidad de conocimientos a nivel preparatoria, al entrar a

la universidad se vuelve un problema mayor, el cual desemboca en un alto índice de reprobación y rezago.

La deserción y el rezago de los alumnos de Ingeniería, en particular, aquellos alumnos que cursan las Ciencias Básicas, se ha vuelto uno de los problemas más significativos de la misma Facultad. Y es que, el bajo nivel de conocimientos previos a la carrera, aunado a una serie de problemas, hace que al alumno se le complique su paso por los primeros semestres de la carrera.

De acuerdo con el diagnóstico de conocimientos aplicado por la DCB, de la Facultad de Ingeniería UNAM, dos de cada diez estudiantes obtienen una calificación aprobatoria, mayor o igual a 6, sin embargo, se observa que la calificación regular de los estudiantes está alrededor de 4.5, en una escala de 0 a 10. Si lo dividimos por áreas de conocimiento, en el caso de Matemáticas obtienen en promedio una calificación de 4.6, en Física general de 3.9 y en Química de 5.0 (Escobar, 2012).

Más allá de los datos que arroja el diagnóstico que aplica la DCB, debemos pensar que existen diferentes elementos que causan el alto índice de reprobación y deserción, entre ellos, están:

- Deficiencia de conocimientos antecedentes. Algo que se presenta con mayor frecuencia, es el bajo nivel de conocimientos antecedentes con los

que cuenta el alumno de los primeros semestres de la carrera, tanto de física como de Matemáticas, materias indispensables para el estudio de la ingeniería desde sus primeros semestres. (García, 2008)

- Alto número de estudiantes. Según datos consultados en la página de la Facultad de ingeniería de la UNAM, en los ciclos correspondientes a los años 2013-2014, la matrícula oscila entre 12,000 y 13,000 estudiantes (Facultad de Ingeniería, 2015). Esto se vuelve un problema cuando existen grupos numerosos, en donde se vuelve difícil identificar a los alumnos con problemas para entender la clase y que tengan una atención personalizada.

Por otro lado, no solo los alumnos son los que tienen que cumplir con sortear retos y problemas, también los profesores de la Facultad de Ingeniería de la UNAM deben enfrentar de manera simultánea diferentes aspectos, entre los que destacan la transmisión eficiente de conocimientos y habilidades matemáticas básicos, la utilización de métodos y estrategias de enseñanza modernos y la búsqueda de mecanismos que permitan maximizar el aprendizaje. Todo lo anterior en un contexto en donde destaca la complejidad que implica el tener que trabajar con un número muy alto de estudiantes inscritos en la institución.

Por ello, la Facultad contempla algunos proyectos, los cuales, en términos generales, tratan de mejorar el proceso enseñanza-aprendizaje ante una amplia inscripción de alumnos que deben ser atendidos, así como disminuir el rezago

estudiantil ante el alto número de estudiantes irregulares con que se cuenta actualmente y, por último y quizás lo más importante, se busca que exista una formación y actualización continua por parte del profesorado ante las nuevas metodologías, estrategias didácticas y herramientas tecnológicas con las que se cuenta actualmente. (Gómez, 2016)

Es claro que la manera de aprendizaje así como las metodologías y estrategias de enseñanza se van modificando con el paso del tiempo, por ejemplo, hoy en día contamos con más y mejores herramientas que nuestros antepasados, herramientas que logran mejorar significativamente la manera en que aprendemos nuevo conocimiento. Tanto el estudiante como el profesor, deben aprovechar al máximo la utilización de dichas herramientas ya que potencializarán su capacidad de aprendizaje significativamente.

El estudiante tiene que tomar en cuenta que su conocimiento y sus estrategias de aprendizaje deberían ser ilimitados y al mismo tiempo lo más preciso posibles, esto con la finalidad de tener una mejor preparación y un mejor criterio al momento de enfrentarse con un problema.

1.4 Rezago estudiantil en la FI

En los últimos años, la Facultad de Ingeniería ha tenido que lidiar con nuevos y diversos problemas, uno de los más importantes es el rezago que los alumnos

tienen con respecto a los conocimientos y materias que han de cursar durante los primeros semestres de la carrera.

Con el claro objetivo de identificar el nivel de conocimientos y habilidades con los que ingresan alumnos de nuevas generaciones, la Facultad de Ingeniería a través de la DCB, practica desde hace ya varios años un examen diagnóstico, o sea un examen enfocado a temas como Matemáticas, Física y Química, el cual ha dejado ver que existe una estrecha relación entre los bajos resultados del mismo y las calificaciones reprobatorias que obtienen los mismos alumnos al finalizar su primer semestre de licenciatura (Rodríguez, 2015).

El examen diagnóstico califica cinco áreas dentro de las Matemáticas, las cuales son: Álgebra, Cálculo, Geometría Analítica, Geometría Euclidiana y Trigonometría. Cada una de esas áreas es evaluada independientemente dentro del mismo examen, eso le permite tener resultados parciales y observar cuáles áreas son las que necesitan más atención y, por otro lado, se tiene un resultado general que permite evaluar conocimientos generales en Matemáticas. (Rodríguez, 2015)

Las cifras indican que al menos en los últimos años, ha ido incrementando el valor de la calificación general en Matemáticas (Rodríguez, 2015).

Al menos en 2008, las calificaciones para cada materia, y en general, estuvieron por debajo de la calificación aprobatoria. Ver Tabla 1.

Tabla 1: Tabla de calificaciones promedio que arrojó el examen diagnóstico aplicado en el año 2008 (Gómez, 2016).

Materia	Calificación
Álgebra	4.82
Trigonometría	4.84
Geometría Euclidiana	4.55
Geometría Analítica	3.72
Cálculo I	3.72
General	4.41

Siete años después, en el 2015, el examen diagnóstico se siguió aplicando, y aunque algunas materias tuvieron un incremento en la calificación, otras obtuvieron una calificación menor a la de años pasados, pero eso no impidió que la calificación general incrementara. Ver Tabla 2.

Tabla 2: Tabla de calificaciones promedio que arrojó el examen diagnóstico aplicado en el año 2015 (Gómez, 2016).

Materia	Calificación
Álgebra	5.89
Trigonometría	3.88
Geometría Euclidiana	5.53
Geometría Analítica	4.08
Cálculo	4.98
General	5.04

Cabe mencionar que existen fluctuaciones año con año en las calificaciones, tanto de las materias, cada una por separado, como de la calificación promedio general de estas asignaturas, todas ellas, correspondientes al Departamento de Matemáticas. Y aunque el promedio general incrementa año con año, sigue siendo una calificación reprobatoria.

Las asignaturas que presentan un mayor número de alumnos reprobados son: Cálculo I, Álgebra y Geometría Analítica, las cuales son asignaturas de primer semestre, seguidas por: Álgebra Lineal y Cálculo II, pertenecientes al segundo semestre de la carrera (Gómez, 2016).

Los resultados muestran que dentro de los cuatro primeros semestres de la carrera, aquellos que pertenecen a las Ciencias Básicas, en particular Matemáticas y Física, es en dónde se encuentra el mayor índice de reprobados de toda la carrera.

1.5 Propuestas de la DCB para solucionar el rezago

Consciente del problema, la DCB se ha planteado diversas metas a cumplir, teniendo como principales objetivos:

- Dar al estudiante herramientas que le permitan afrontar problemas relacionados con el aprendizaje de alguna asignatura. Tales herramientas deberán ser de fácil acceso al estudiante.
- Implementar medidas que permitan al estudiante regularizar su situación académica. Estas medidas, deberán contemplar y atender a todos aquellos alumnos que tengan problemas de rezago (División de Ciencias Básicas, 2015).

Con tales objetivos, se busca lograr un contacto más personal con cada estudiante, y aunque es evidente que un trato individual es casi imposible, es necesario buscar mecanismos que atiendan a estudiantes en situaciones de aprendizaje más complicadas. En general, se busca que el alumno tenga una formación más integral.

Como apoyo a la formación académica de los estudiantes, la DCB ofrece diversas herramientas que van desde asesorías presenciales hasta material didáctico en Internet (CENTRO DE RECURSOS DE APRENDIZAJE PARA LAS CIENCIAS BASICAS, 2011).

En los últimos años, el material didáctico que ofrece la DCB, ha ido evolucionando junto con la tecnología, hoy en día ese material tiende a ser multimedia: documentos, fotografías, sonidos, videos, páginas web y aplicaciones móviles,

etc., integrando también los laboratorios virtuales, los cuales hacen uso de la realidad virtual.

La DCB ha integrado este tipo de recursos multimedia debido a las ventajas con las que cuenta tanto en evaluación como en la difusión de la información y un mejor entendimiento de temas que requieren un amplio razonamiento.

Materias como Geometría Analítica, Álgebra o Cálculo se ven beneficiadas gracias a este tipo de recursos multimedia, específicamente por la creación de aplicaciones móviles que logran ejemplificar desde espacios vectoriales en tres dimensiones, 3D, hasta superficies, también vistas en 3D (CENTRO DE RECURSOS DE APRENDIZAJE PARA LAS CIENCIAS BASICAS, 2011).

2 Las TIC en la educación

En los últimos años las Tecnologías de la Información y Comunicación se han vuelto cada vez más importantes en el desarrollo de las clases en diferentes niveles educativos.

Estas tecnologías cuentan con la gran ventaja o desventaja según se vea, de que la información que se maneja es mucha y se genera de forma muy rápida, eso hace que el profesor pueda detectar de manera más rápida y precisa el avance académico de los estudiantes, así como sus capacidades y sus fallas.

El uso de las nuevas TIC como apoyo a la docencia proporciona un ambiente de aprendizaje colaborativo entre estudiantes y profesores, y esto permite la creación, y adopción, de un entorno de aprendizaje basado en la tecnología. (INÉS, 2016)

Es importante destacar que el desarrollo de las nuevas TIC como son los recursos didácticos, deberán contar con estrategias didácticas de tal manera que puedan generar un aprendizaje significativo en el estudiante.

También se debe resaltar que el estudiante debe sentirse acompañado en el proceso enseñanza-aprendizaje, ya que la asimilación y acomodación de los nuevos conocimientos y habilidades en los estudiantes son de suma importancia, es por ello que los recursos didácticos generados a partir de las nuevas TIC deberán apoyar a que el alumno pueda repasar una y otra vez el nuevo aprendizaje.

2.1 Juegos didácticos en las Matemáticas

La actividad lúdica es el conjunto de estrategias que buscan llamar la atención de los alumnos, activar sus sentidos, su imaginación y, lo más importante, influir para sientan una motivación por aprender. También propician una interacción constante entre los alumnos y gracias a los intercambios en la discusión con el profesor, los

mismos alumnos formulan hipótesis, realizan análisis y obtienen conclusiones (Roberto, 2016).

El juego, como método de enseñanza, ha sido utilizado desde hace ya bastante tiempo, y en la era de la tecnología sigue siendo parte importante del aprendizaje. Es por ello que el desarrollo de juegos lúdicos, utilizados como herramienta para adquirir nuevo conocimiento, permitirá aportar valor a los métodos de enseñanza-aprendizaje con los que cuentan hoy en día los profesores.

Con juegos lúdicos se pretende apoyar tanto a los profesores como a los alumnos, en el proceso de enseñanza-aprendizaje, y es que, por una parte, el alumno tendrá oportunidad de aprender, o reforzar, de una manera más relajada y divertida el conocimiento que el profesor transmite y, por otra parte, el profesor observará de manera más detallada el avance que llevan sus alumnos respecto a la adquisición y comprensión de dicho conocimiento y reforzará el tema en cuestión.

Es por ello que se ha pensado en el juego como manera de aprender, reforzar y retener información de una forma más amigable, ya que aprender jugando puede llegar a disminuir la presión de tener que recordar y aprender muchas cosas, y al mismo tiempo el alumno se sentirá motivado para seguir aprendiendo.

Lo lúdico como apoyo a la enseñanza, permite que exista una cierta interacción entre creatividad, competición, entretenimiento y participación, lo cual favorece ampliamente, y de forma amena, al proceso enseñanza-aprendizaje (González, 2008).

2.2 M-learning en la educación

El acelerado desarrollo de la tecnología que se crea en estos tiempos, propicia una nueva generación de seres humanos, la generación TIC, gracias a que los dispositivos móviles que se manejan día con día contienen telefonía celular y se puede acceder a Internet casi desde cualquier lugar, además de que se cuenta con la gran ventaja, entre otras cosas, de tener información almacenada en memorias cada vez más pequeñas y además extraíbles.

La tecnología móvil favorece a que el proceso enseñanza-aprendizaje se dé en cualquier lugar y cualquier momento. Las redes inalámbricas hacen que sea posible la comunicación, y que ésta se pueda dar de manera síncrona o asíncrona.

M-learning, o aprendizaje móvil, como parte de esta nueva ola de tecnología nació a finales de los 90's, en PEP, *Palm Education Pioneers*, una asociación de profesores de Estados Unidos, en donde se hizo la invitación a participar en un concurso para el desarrollo de proyectos para la enseñanza donde se utilizarán las

agendas electrónicas personales. Para el 2001, en Europa, se adopta ésta tecnología con la finalidad de apoyar a los jóvenes que estaban desertando en materias enfocadas a la lectura y Matemáticas.

El aprendizaje móvil favorece al estudiante en el desarrollo de habilidades cognitivas puesto que ellos hacen suyo el conocimiento al explorar nuevos retos, además de que pueden trabajar de forma individual o colaborativa.

El aprendizaje móvil es por definición personal y social. Además, aprovecha las habilidades propias de los jóvenes de la era digital. A medida que la penetración de los *Smartphones*, *PocketPC* y el incremento de ancho de banda y conectividad están disponibles, el uso de dispositivos móviles para el aprendizaje será una progresión natural en las actividades educativas (Sánchez, 2010).

Por otro lado, las aplicaciones móviles portátiles permiten la comunicación de los estudiantes, existe realimentación inmediata en algunas ocasiones y la entrega de trabajos es instantánea. Sumado a esto el aprendizaje en los dispositivos móviles es efectivo y puede ser personalizado.

2.2.1 El rol del profesor y el rol del alumno en M-learning

El rol del profesor en el aprendizaje móvil es difundir su experiencia y sus habilidades en las aplicaciones móviles, apps, así como comunicarse con sus

pares para intercambiar sus conocimientos y habilidades, discutir sobre los temas, plantear, resolver y realimentar las actividades que realizan sus estudiantes. También es el encargado de planear, organizar y llevar la logística de las actividades que va incluir en su curso o clase, y de organizar los módulos y actividades de aprendizaje como considere que sea más conveniente. Con el uso del aprendizaje móvil, se pretende apoyar al profesor en el proceso de enseñanza-aprendizaje.

El rol del alumno en el aprendizaje móvil es: interactuar con los dispositivos móviles, mantenerse informado de lo que envía el profesor si es que el alumno se encuentra fuera del aula, informarse de las actividades que deja el profesor, así como de información sensible, debe estar atento ante los diversos recursos que tiene dentro del aula, etc. Con toda esa información que tiene el alumno a su alcance, podrá investigar más allá de lo que se ve en el aula.

Con base a lo anterior es importante que tanto alumnos como profesores evalúen la utilidad de las apps como herramientas de apoyo en el proceso enseñanza-aprendizaje.

Es importante señalar que la tarea del desarrollador de las apps para el aprendizaje móvil, es recabar la información de los expertos en la materia y después concentrarla en un recurso didáctico que pueda ser utilizado por los estudiantes.

2.3 Plataformas Educativas

Como apoyo al aprendizaje móvil, surge la idea de desarrollar plataformas web en dónde se pudiesen compartir recursos, crear tareas, y hasta evaluar el conocimiento de los alumnos en un solo lugar, es así que surgen las plataformas educativas.

Una de las plataformas más usadas a nivel mundial es Moodle, ya que cuenta con diversas actividades que refuerzan el proceso de aprendizaje de los alumnos, muchas de las cuales sirven para publicar materiales, o bien para que los alumnos realicen trabajos de forma individual o en grupo, llámese: wikis, bases de datos, talleres, foros, etc., y otras que pueden ser útiles para realizar exámenes. (Cubillo, Recursos TIC en Educacion, 2012)

Edmodo es otra plataforma educativa a nivel mundial que ayuda a conectar a los estudiantes con la gente y recursos necesarios para desarrollar todo su potencial (Edmodo, 2016), su desarrollo basado en microblogging proporciona al docente un espacio virtual y privado en el cual puede compartir mensajes, archivos, tareas y actividades y gestionarlas de manera sencilla (González M. P., 2013).

Localmente, en la UNAM, se cuentan con sistemas similares a los antes mencionados, la Red Universitaria de Aprendizaje, RUA, es una plataforma que

ofrece, a la comunidad universitaria y la sociedad en general, recursos educativos tales como: textos, videos, contenidos multimedia, etc, asociados a los planes de estudio vigentes de la UNAM, la mayoría de estos recursos son generados por miembros de la misma comunidad universitaria (RUA, 2016).

La Facultad de Ingeniería UNAM, también cuenta con plataformas educativas, por un lado el Sistema Integral de Apoyo al Proceso de Enseñanza-Aprendizaje en la Facultad de Ingeniería, SIAEFI, surge con la finalidad de ayudar a profesores y alumnos a organizar las clases y compartir material didáctico: temarios, tutoriales, artículos, manuales, etc., así como llevar un control de las calificaciones de los alumnos de cada grupo.

Por otro lado, existe el Centro de Recursos de Aprendizaje para las Ciencias Básicas, CERAFIN, que en general contiene material didáctico digital, así como diversos recursos de consulta y ejercicios didácticos disponibles para toda persona interesada en temas de áreas o asignaturas impartidas en la División de Ciencias Básicas de la Facultad de Ingeniería, UNAM.

2.4 Software educativo

Existen programas informáticos, además que se crearon con la finalidad de ser utilizados por estudiantes como medios didácticos, a este tipo de programas se les

conoce como: software educativo, y surgen como apoyo a los procesos de enseñanza-aprendizaje.

Actualmente, el software educativo que se desarrolla promueve un aprendizaje interactivo a través de recursos multimedia, esto despierta el interés y la participación del alumno, lo cual hace que incremente su motivación por aprender. La interactividad que brinda este tipo de software favorece a una realimentación inmediata y así el alumno puede observar el progreso de su aprendizaje.

Este tipo de software, además de acercar a las nuevas TIC a la educación, hace que el estudiante adquiera competencias digitales necesarias para su formación, al introducir un elemento innovador en el proceso enseñanza-aprendizaje se fomenta la ejercitación y el refuerzo de contenidos específicos (Caccuri, 2013).

Existe una gran variedad de este tipo de software y cada uno ha sido desarrollado para ciertos niveles escolares que van desde la educación básica hasta la educación superior; también se desarrollan para diferentes materias, que van desde el civismo hasta las Matemáticas, respecto al presente trabajo, se centró la atención en aquellos que apoyan la enseñanza de las Matemáticas. Entre los más conocidos se encuentran: *Geogebra*, un referente en la Didáctica de las Matemáticas desde la educación primaria hasta el ámbito universitario (Cubillo, 2012), y *WolframAlpha*, el cual es capaz de generar algoritmos, gráficos en 3D,

simulaciones y modelos (Dirección General de Tecnologías de la Información y la Comunicación, 2015).

Como los anteriores, podemos citar muchos tipos de software dedicados a la enseñanza y mejora de la educación, pero no es el objetivo final del presente trabajo. Sin embargo, se puede decir que la variedad del software educativo desarrollado es bastante y de buena calidad, y aunque mucho de este software es gratuito, para el Sistema Operativo Android, hay versiones que tienen un costo pero permiten una gama mayor de herramientas a ser utilizadas por el usuario.

2.5 Aplicaciones Android con realidad aumentada para la enseñanza-aprendizaje de Matemáticas

Las Matemáticas en la ingeniería son fundamentales, y el estudio de ellas muchas veces se vuelve difícil, debido a la gran cantidad de información que se requiere comprender en un tiempo corto, y la gran presión de retener y comprender dicha información.

El desarrollo de estos juegos está enfocado al apoyo de materias básicas en la formación de ingenieros, que en su mayoría son materias de la rama de la Física y las Matemáticas, en las cuales, se ha visto un claro rezago y falta de conocimientos previos por parte de los alumnos. Por lo anterior, los juegos didácticos a desarrollar serán exclusivamente matemáticos, como apoyo a los

planes que tiene la misma Facultad para afrontar el rezago estudiantil en ésta materia.

2.5.1 Aplicaciones Android para el apoyo de Matemáticas

Sin duda alguna, en el tiempo presente, estamos viviendo una revolución de tecnología móvil y en todo lo relativo al acceso a la información, y la aparición de los dispositivos móviles, tablets y smartphones, son quizá la base de este cambio en la nueva era de la tecnología (Gaspar, 2012).

El hecho de estar conectados en casi cualquier sitio y a cualquier hora, ha dado pie al desarrollo de miles de aplicaciones que ofrecen un aumento de conocimiento y una reducción de distancias al obtener información de manera más rápida y precisa (Gaspar, 2012). Es por eso que empresas como Google, han apostado por el cambio y han desarrollado Sistemas Operativos, SO, como Android, para cubrir la gran cantidad de dispositivos que se lanzan al mercado y que cada día están más optimizados en cuanto a nivel de procesamiento.

Además de ser uno de los SO más utilizados por los dispositivos móviles actuales, Android cuenta con una tienda en línea en la cual se pueden encontrar miles de aplicaciones divididas en diversas categorías. En este caso son las aplicaciones educativas las que interesan, específicamente, las de contenido matemático, las

cuales existen actualmente para todos los niveles educativos, y van desde calculadoras básicas hasta la enseñanza de geometría en tres dimensiones.

Y es que la necesidad de simplificar procedimientos complejos, de calcular rápidamente ciertas operaciones o simplemente de entender el mundo que nos rodea basándonos en la ciencia, ha hecho que el método de enseñanza adopte, y al mismo tiempo se adapte, al surgimiento de estas nuevas aplicaciones.

Existe un gran número de aplicaciones que se enfocan al apoyo de las Matemáticas, como se ha mencionado, las hay para distintos niveles de educación, pero las que interesan aquí son las desarrolladas para el apoyo de Matemáticas a nivel superior. Aplicaciones como Geogebra o WolframAlpha, son de las más conocidas y usadas por la comunidad universitaria para la resolución de problemas complejos. Pero existen muchas más que son de gran ayuda para la comprensión de las Matemáticas.

Un ejemplo sería: *MyScript Calculator*. Es una aplicación que realiza operaciones Matemáticas de forma natural a través de la escritura del usuario. Especialmente diseñado para los dispositivos Android, en esta aplicación el usuario escribe la expresión matemática en la pantalla y el software convierte esos símbolos y números en texto digital, todo en tiempo real, además de reconocer una amplia gama de operadores matemáticos, este software es fácil, sencillo e intuitivo (MyScript, 2016).

Otra aplicación parecida es: *Photomath - Cámara calculadora*. Como su nombre lo indica, esta aplicación utiliza la cámara del dispositivo, con la cual el usuario debe apuntar a un problema matemático y el software mostrará el resultado con una explicación paso a paso detallada. Esta aplicación abarca temas que van desde la aritmética, la trigonometría, las raíces, hasta las ecuaciones, las funciones de distintos tipos y derivadas e integrales (PhotoMath, 2016).

Otras aplicaciones que sirven de apoyo al estudio de las Matemáticas son: *Desmos*, *Andie Graph*, *Algeo Graphing Calculator* y la Calculadora Gráfica de Matlab, todas ellas podrían englobarse como calculadoras científicas dedicadas a las gráficas, ya que la mayoría de ellas actúa como una calculadora científica bastante completa y además incluyen la generación de gráficas, en dos dimensiones, 2D, a partir de ecuaciones o funciones que el usuario proporciona.

Todas las anteriores son ejemplos de aplicaciones que se han desarrollado para el SO Android. Es importante mencionar que no todas ellas son nativas de este SO, es decir, en un principio no fueron desarrolladas para Android, pero gracias a que tuvieron buena respuesta de aceptación en su SO original, se lograron desarrollar versiones para Android, colocándose dentro de la lista de las mejores aplicaciones para Matemáticas de este SO.

2.5.2 Realidad Aumentada con Android

Desde hace tiempo, la Realidad Aumentada surge como una de las tecnologías con más popularidad y poco a poco se está incorporando a diferentes ámbitos. Aunque en su origen ha tenido más impacto en áreas como: publicidad, marketing e investigación, en el ámbito educativo esta tecnología resulta especialmente valiosa (Educa con TIC, 2014). Y es que cada vez son más las aplicaciones que se desarrollan aplicando esta tecnología debido a las grandes ventajas que pueden ser explotadas en el ramo educativo.

La Realidad Aumentada se define como una tecnología que se basa en combinar una visión de la realidad con algo que es virtual: información, vídeo, volúmenes, etc., a través de un dispositivo. Es decir, se crea un conjunto entre la información virtual y la información física ya existente (Educa con TIC, 2014), no sustituyendo la realidad física, sino que se sobreponen los datos virtuales al mundo real.

Por ejemplo, al mirar un edificio singular de cualquier ciudad a través de nuestro teléfono inteligente, la tarea de la Realidad Aumentada sería agregar información sobre la historia, características, etc. del edificio en nuestra pantalla. Combinando así el mundo real, en este caso el edificio frente a nosotros, con el mundo virtual, que sería la información que se desplegaría en nuestro dispositivo móvil.

2.5.3 Elementos necesarios para la Realidad Aumentada

Como cualquier otro sistema computacional, los sistemas de realidad aumentada requieren de componentes hardware y software sobre los que se vayan a llevar a cabo las acciones deseadas.

Los elementos de hardware necesarios para cualquier sistema de realidad aumentada son: una cámara, una pantalla de visualización o equipos holográficos en 2D o 3D, un equipo de procesamiento y cálculo, componentes de almacenamiento y, por último, dispositivos de transmisión y recepción de datos en caso de que sean necesarios. La tabla 3 muestra un resumen de los componentes habituales de hardware en un sistema de realidad aumentada.

Tabla 3: Resumen de componentes de hardware de un sistema de realidad aumentada (Pombo, 2014).

Técnica	Componente Hardware
Captura video-through	Todo tipo de cámaras de vídeo
Captura see-through	Cámaras integradas de vídeo, HMD (Head Mounted Display)
Reconocimiento geoposicionamiento	Antena GPS, sensores de movimiento
Reconocimiento híbrido	Antena GPS, sensores de movimiento, dispositivos de captura de imágenes
Tratamiento imágenes	Procesadores con velocidad de procesamiento de, al menos, 2 GHz
Almacenamiento	Disco duro convencional
Comunicaciones locales	Tarjeta de red, conectores RJ45, antena wireless, punto de acceso
Comunicaciones móviles	Equipamiento GSM
Visualización video-through	Pantallas de vídeo, monitores, proyectores
Visualización see-through	HMD (Head Mounted Display), teléfonos móviles

Al igual que sucede a la hora de elegir el hardware adecuado, la decisión de utilizar un software u otro viene determinado por las prestaciones y características del sistema.

En términos generales, un sistema de realidad aumentada necesita software de reconocimiento, ya sea por posicionamiento o por imágenes (con o sin marcadores), librerías de procesamiento de imágenes (si es que lo necesitase), al menos una base de datos y, si fuese necesario, software de comunicaciones. La tabla 4 muestra un resumen del software genérico necesario para cada etapa.

Tabla 4: Resumen de los requisitos software de un sistema de realidad aumentada (Pombo, 2014).

Técnica	Componente software
Captura escena	Controladores de cámara de vídeo
Reconocimiento visual	Librerías de reconocimiento de imágenes
Reconocimiento geoposicionamiento	Brújula digital, software GPS
Reconocimiento híbrido	Brújula digital, librerías de reconocimiento de imágenes, software GPS
Tratamiento imágenes	Librerías de tratamiento de imágenes
Almacenamiento	Base de datos
Comunicaciones locales	Controlador de red
Comunicaciones móviles	Controlador GSM
Visualización contenidos	Software de reproducción de contenidos multimedia, librerías de tratamiento de imágenes

En términos más sencillos y generales, para poder implementar sistemas de realidad aumentada, son necesarios cuatro elementos básicos, los cuales se enlistan a continuación:

- Cámara. Este elemento captura la información del mundo real y la transmite al software encargado de mostrar la realidad aumentada.
- Pantalla. En este elemento se refleja la fusión de las imágenes reales y las imágenes virtuales que, en suma, conforman la realidad aumentada.
- Software. Su objetivo es interpretar la información del mundo real, generar la información virtual y fusionarla de forma adecuada para obtener una realidad aumentada.
- Marcadores. Elemento también denominado: activador, este sería la imagen real que visualiza el usuario y a partir de la cual el sistema debe mostrar las imágenes generadas por el software

3. Desarrollo de una aplicación de Realidad Aumentada para Android

Como se ha venido mencionando en este trabajo de tesis, la tecnología está evolucionado de manera considerable, y es gracias a eso que ahora podemos disfrutar de un modo de vida que antes era casi imposible imaginar. El avance en el desarrollo de hardware hace posible que las aplicaciones y recursos digitales creados sean más robustos, aprovechando al máximo el potencial del que los procesadores de hoy en día son capaces de lograr.

3.1 Requerimientos

El hardware es un factor importante a considerar en el desarrollo de una aplicación, no solo de realidad aumentada, sino en general, hay que tomar en cuenta la capacidad que tendrá un dispositivo para lograr procesar toda la información necesaria para poder ejecutar una aplicación de manera adecuada.

Si bien las primeras generaciones de celulares contaban con un gran avance tecnológico para su época, no fue hasta la tercera generación de móviles, 3G, que surgieron los primeros dispositivos capaces de conectarse a la red de Internet y ofrecer servicios multimedia (Rábanos, 2015), así como también velocidades más grandes de transferencia de datos, comparado con las primeras generaciones.

Por lo anterior, es necesario contar con un dispositivo móvil de tercera generación para poder ejecutar una aplicación cualquiera, básicamente las características mínimas que debería tener un dispositivo móvil para soportar aplicaciones de todo tipo serían: la capacidad para ejecutar tareas multimedia, voz, datos, audio, video, imágenes, etc., y una interconexión con servicios móviles por satélite (García J. L., 2005).

3.2 Metodologías para la Ingeniería de software

Para el desarrollo de cualquier sistema computacional se debe pasar por un proceso intensivo de recopilación de conocimiento, requerimientos, diseño,

desarrollo, pruebas, implementación y mantenimiento. Todo ello, en conjunto, se define como Ingeniería de software, y su objetivo principal es buscar convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de calidad y que satisfaga las necesidades y expectativas del cliente o usuario final.

Los sistemas computacionales van madurando y los profesionales de las nuevas Tecnologías de la Información y Comunicación se van dando cuenta de que es necesario seguir ciertas pautas predefinidas en el desarrollo del software de calidad, es decir, llevar un comportamiento metódico: seguir una metodología.

Una metodología impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Ésta define una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema.

Existen diferentes metodologías que han sido en los últimos años herramientas de apoyo para el desarrollo del software, las cuales han evolucionado de manera significativa en las últimas décadas. Hoy en día existe una amplia variedad de metodologías para el desarrollo de software, las cuales se engloban en tres categorías: tradicionales, ágiles e híbridas.

Se suele llamar tradicionales a todas aquellas surgidas antes de los años 90, las cuales pretendían ayudar a los profesionales indicando pautas para realizar y documentar cada una de las tareas del desarrollo del software. Sin embargo, casi todas ellas asumen que un proyecto de desarrollo es casi una extensión de un proyecto burocrático tradicional, no suelen tomar en cuenta cosas como la calidad, la satisfacción, la competitividad, los beneficios, etc. Fueron metodologías creadas en los años 70-80 pensando en los negocios de los años 50.

A partir de los años 90, surgen las metodologías ágiles, tomando en cuenta lo cambiante y amplio que es el mundo del software, se pensó en que una metodología debía ser lo suficientemente precisa como para que todo el mundo la pudiera seguir, suficientemente adaptable como para poder aplicarse en distintos proyectos, y suficientemente sencilla como para que no resultara muy costosa su utilización. En otras palabras, esta metodología debía permitir desarrollar software rápidamente y responder a los cambios que podrían surgir a lo largo de los proyectos. Y aunque ya estaban siendo utilizadas con éxito en proyectos reales, aún faltaba una mayor difusión y reconocimiento.

Según Ivar Jacobson, ingeniero sueco en Ciencias de la computación, la nueva tendencia en ingeniería de software es diseñar metodologías híbridas, las cuales retoman ventajas de los dos tipos de metodologías anteriores, sin embargo, antes de crear una nueva metodología híbrida para desarrollar proyectos de software, es

necesario detenerse a pensar si realmente es una opción factible e ideal para el desarrollo de proyectos.

En la tabla 5 se ven las tres metodologías poniendo una breve descripción así como algunos ejemplos de cada una de ellas.

Tabla 5: Tabla que sintetiza información sobre cada metodología (Hernández, 2012).

Metodología	Descripción	Ejemplo
Tradicionales	<ul style="list-style-type: none"> • Propician las buenas prácticas de la Ingeniería de Software, tales como llevar un orden y tener una estructura bien definida para su buen desarrollo • Se considera de gran importancia la documentación, lo cual permite entender, extender y darle un mejor mantenimiento al software. • Se requiere una gran disciplina de los desarrolladores, por ser un proceso riguroso. • Se genera documentación innecesaria • No tiene respuesta rápida a cambios. • Alta inversión de tiempo en el modelado del sistema 	<p>RUP (<i>Rational Unified Process</i>, Proceso Racional Unificado, IBM, 2011)</p> <p>OMT (<i>Object Modeling Technique</i>, Técnica de Modelado de Objetos, Rumbaugh, 1990 y Yourdon, 1976)</p>
Ágiles	<ul style="list-style-type: none"> • Presentan respuestas rápidas y efectivas al cambio. • El plan del proyecto es 	<p>XP (<i>eXtreme Programming</i>, Programación Extrema, Beck, 2010)</p>

	flexible. <ul style="list-style-type: none"> • Muestra simplicidad en el desarrollo. • Genera poca documentación. • No hace uso de métodos formales. 	SCRUM (Schwaber & Shuterland, 2011) DAS (Desarrollo Adaptativo de Software, Jim Highsmith, 1998)
Híbridas	<ul style="list-style-type: none"> • Retoman ventajas de las metodologías anteriores. • Combina las mejores prácticas que existen en metodologías ágiles y tradicionales. 	EssUP (<i>Essential Unified Process</i> , Proceso Esencial Unificado, Jacobson, 2011)

Si bien es cierto que las metodologías ágiles son las más convenientes para usar en el desarrollo de proyectos de nuevas tecnologías, también es cierto que existen diferencias entre las mismas, por ejemplo SCRUM cuenta con algunas ventajas contra otras metodologías ágiles, entre ellas están:

- Cumplimiento de expectativas: El cliente establece sus expectativas indicando el valor que aporta a cada requisito del proyecto.
- Flexibilidad a cambios. La metodología está diseñada para adaptarse a los cambios de requerimientos de conllevan los proyectos complejos.
- Reducción de tiempo. El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté terminado por completo.

- Mayor calidad de software. La forma de trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
- Mayor productividad. Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
- Predicciones de tiempos. Mediante esta metodología se conoce la velocidad del equipo por sprint (los llamados puntos historia), con lo que consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está retrasada.
- Reducción de riesgos. El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.

Por lo anterior, SCRUM se consideró como la metodología más adecuada a seguir, ya que varios aspectos del sistema de juegos didácticos a desarrollar encajan con dicha metodología. Siendo la flexibilidad a cambios, la reducción de tiempos y la mayor calidad de software las principales razones por las cuáles se siguió dicha metodología de trabajo.

3.3 Herramientas de software para la Realidad Aumentada

Más allá de la metodología usada para un mejor desarrollo, un aspecto importante a considerar al momento de desarrollar un sistema o aplicación, es el software o herramientas que se han de utilizar para lograr obtener el producto deseado. El objetivo de este proyecto es desarrollar una aplicación de realidad aumentada, por lo tanto se debe tomar en cuenta que se necesita un software capaz de generar gráficos en 3D y al mismo tiempo poder reconocer imágenes y generar contenido relacionado a esas imágenes, todo esto en tiempo real.

Actualmente se pueden distinguir básicamente dos tipos de Realidad Aumentada, la que emplea reconocimiento de marcadores o imágenes y la Realidad Aumentada basada en la posición del usuario de la aplicación. El presente trabajo se enfoca a desarrollar una aplicación de Realidad Aumentada basada en reconocimiento de imágenes o marcadores.

3.3.1 Realidad Aumentada

La Realidad Aumentada basada en reconocimiento de imágenes emplea, como bien lo indica, marcadores o imágenes, en los que se superpone algún tipo de información: imágenes, objetos 3D, vídeo, etc., cuando estos objetos son reconocidos por un software determinado. El software en ejecución debe ser capaz de realizar un seguimiento del marcador de tal manera que si el usuario lo mueve, el objeto 3D superpuesto también siga ese movimiento, si se gira el

marcador se puede observar el objeto 3D desde diferentes ángulos y si se acerca o se aleja, el tamaño del objeto aumenta o se reduce respectivamente. También se puede mostrar información adicional de una forma atractiva para el usuario una vez ha sido detectada una imagen o un menú que permita una nueva iteración con el usuario en función del objeto reconocido.

Existen diferentes librerías que permiten la programación de este tipo de aplicaciones, entre ellas: *Metaio*, *Layar*, *Vuforia*, *Total Immersion*, *ARTool Kit*, etc. Todas permiten la creación de aplicaciones de realidad aumentada. Para ello, por medio del seguimiento de vídeo, calculan en tiempo real, la posición de la cámara y la orientación relativa a la posición de los marcadores físicos. Una vez que la posición de la cámara real es conocida, la cámara virtual coloca el nuevo contenido en el mismo punto, superponiendo los modelos 3D exactamente sobre el marcador real. Así, este tipo de software resuelve dos de los principales problemas en la realidad aumentada: el seguimiento del punto de vista y la interacción con el nuevo objeto virtual.

Durante la última década se han desarrollado librerías que permiten al programador explotar al máximo el motor de reconocimiento de imágenes y marcadores. Entre ellas destaca el Kit de Desarrollo de Software, SDK por sus siglas en inglés, de Vuforia, esto debido a que sienta las bases de muchos otros SDKs como Wikitude o Layar, por mencionar algunos. El SDK de Vuforia permite desarrollar en múltiples plataformas móviles como Android o iOS, lo cual posibilita

al desarrollador llegar a un amplio conjunto de usuarios. Por este motivo y por su versatilidad, adaptabilidad y amplia documentación parece la elección más adecuada a tener en cuenta.

3.3.2 Vuforia y el reconocimiento de imágenes

Un sistema de Realidad Aumentada requiere de capacidad para conocer el entorno, o mundo real, a través de las imágenes que recibe el dispositivo, para poder añadir contenidos virtuales. Como ya se mencionó este entorno es reconocido a través del uso de marcadores, que bien pueden ser imágenes, datos de posicionamiento como: latitud y longitud, o cualquier tipo de dato que pueda ser obtenido y procesado por el sistema. A través de la cámara del dispositivo se rastrea, o escanea, constantemente el mundo real en busca de un patrón conocido. Para reconocer la imagen que está recogiendo la cámara del dispositivo, éste debe tener un patrón con el cuál comparar, es decir, conocer qué está buscando.

Las técnicas de reconocimiento de imagen se basan en la búsqueda de patrones de formas conocidas, detección de colores, geometrías y patrones repetidos. Para este proyecto se ha elegido Vuforia por su potencial para el reconocimiento de imágenes y la facilidad con que se integra a los diferentes tipos de software capaces de renderizar en tiempo real los complejos modelos en 3D necesarios

para las plantillas, además de que permite un gran número de marcadores y la posibilidad de crearlos a partir de imágenes proporcionadas por el desarrollador.

Estas características, aunadas a la posibilidad de trabajar con Unity, una potente herramienta de renderizado, fueron las que ayudaron a tomar la decisión de utilizar Vuforia.

3.3.3 Unity y Realidad Aumentada

Debido a la facilidad y flexibilidad con la que se pueden crear juegos, aplicaciones interactivas, visualizaciones y animaciones en 3D tiempo real, el motor gráfico Unity es uno de los software más completos con los que se puede desarrollar juegos, no sólo para Android, también permite desarrollar aplicaciones para diferentes plataformas tales como: *PC, Mac, Flash (Hasta la versión 4), XBox, PS2/3/4, PSVita y iPhone*, siendo ésta una de sus grandes ventajas, el desarrollo multiplataforma.

Como se ha mencionado antes, para el desarrollo de aplicaciones de realidad aumentada con elementos 3D, Unity y Vuforia son una excelente combinación. Por un lado, Unity nos permite obtener el punto de vista de la cámara del dispositivo y, por otro lado, Vuforia se encarga de relacionar la posición de los objetos virtuales basado en la posición en la que se encuentra el marcador físico, esto hace que se tenga una buena experiencia de Realidad Aumentada.

Unity puede trabajar con modelos 3D de cualquier tipo creado por un software de modelado. Sin embargo, hay ciertos tipos de objetos primitivos que pueden ser creados directamente dentro de Unity, principalmente están: Cube, Sphere, Capsule, Cylinder, Plane y Quad. Estos objetos ofrecen una manera rápida de crear marcadores de posición y prototipos para propósitos de pruebas.

Para modelos más elaborados existe una variedad de software dedicado al modelado de elementos en 3D, algunos de los más conocidos son: *AutoCAD*, *Maya3D*, *3ds Max*, *Sketchup*, *Blender*, entre otros.

3.3.4 Blender y el modelado en 3D

Las diferencias entre los diversos software para modelado en 3D, al menos a nivel aficionado, radican básicamente en su interfaz, ya que cuentan con herramientas muy parecidas; si se piensa en un nivel más avanzado, o profesional de modelado, debe mencionarse que esa variedad de ese software está enfocado a modelar distintas cosas, algunos están pensados para la arquitectura o para diseñadores ambientales o industriales, y otros para juegos de video, películas animadas o efectos visuales.

Pensando en que no es lo mismo hacer el modelo de un edificio a modo de maqueta en 3D con sus puertas, ventanas, muebles, etc., a tener que hacer la

animación de algún videojuego o alguna escena de una película animada, se debe identificar el tipo de software que mejor convenga a los objetivos de un proyecto. Como el presente trabajo está enfocado a desarrollar un juego de realidad aumentada, lo más acertado ha sido pensar en un software enfocado a las animaciones o a generar modelos 3D de personajes muy elaborados para trabajar junto con las herramientas ya mencionadas. Los más destacables en ese sentido serían: Maya3D, 3ds Max y Blender.

Tanto Maya3D como 3ds Max fueron creados por Autodesk, una compañía dedicada al software de diseño en 2D y 3D, pensados como herramientas de ayuda para las industrias de manufactura, infraestructura, construcción, etc. Blender, por el contrario, es un programa de código abierto y está dedicado especialmente al modelado, iluminación, renderizado y animación, además de desarrollar videojuegos gracias a su motor de juegos interno.

Por sus características, Blender pareció la mejor opción para el desarrollo de este proyecto, el buen desempeño que tiene al trabajar junto con Unity y el hecho de ser de código abierto, hacen que este software sea una herramienta poderosa y accesible, ya que no hay necesidad de pagar una licencia por su uso y cuenta con el respaldo de una comunidad enorme que aporta para su constante desarrollo.

3.3.5 XAMPP para la base de datos

Lo anterior es software para la construcción del juego de realidad aumentada, pero en todo juego es importante guardar datos, ya sea el avance del mismo, la puntuación generada por el jugador, o simplemente datos del mismo jugador. Este juego no es la excepción, más aún cuando se trata de un juego enfocado al aprendizaje, habrá que guardar datos del usuario: nombre, grupo, calificación obtenida, etc., en general todo dato que pudiera ser de utilidad para llevar un seguimiento de su avance en conocimientos y habilidades.

Todos esos datos deberán estar alojados en algún lugar en donde puedan ser consultados y evaluados. Para ello existen los llamados servidores, un tipo de software cuyo propósito es alojar y proveer datos de modo que otras máquinas puedan utilizar esos datos. Entre los más comunes están: Apache, Microsoft IIS, WAMP, XAMPP, LAMP, etc. Siendo los tres últimos considerados como paquetes de software más que servidores en sí, cuentan con la gran ventaja de tener en un sólo software una base de datos: MySQL, el servidor web: Apache, y los intérpretes para lenguajes de script: PHP y Perl.

Esta característica de contener varias aplicaciones en un solo software facilita enormemente tanto la instalación como la configuración de dichas aplicaciones. El nombre de cada paquete es un acrónimo de los programas que lo conforman y el Sistema Operativo, SO, para el cual están desarrollados, es decir, LAMP es el acrónimo de Linux, que es el SO, Apache, como servidor web, MySQL como

gestor de base de datos y Perl, PHP o Python como lenguajes de programación. WAMP contiene las mismas aplicaciones pero está dedicado a trabajar con Windows, y la 'X' de XAMPP, indica que el software puede trabajar con cualquiera de los diferentes sistemas operativos.

Pensando en que este proyecto pudiera ser adaptable a diferentes SO, se eligió XAMPP como software para el resguardo de la base de datos que contendrá los datos generados por el juego. Además de contar con esa virtud de ser adaptable, este software actúa como un servidor web libre, es fácil de usar gracias a su panel de control y la instalación del mismo no es complicada y es rápida.

3.4 Construcción del Juego de Realidad Aumentada

Ya que se dieron a conocer las herramientas de software que se eligieron para la construcción del juego se explicará cómo trabajan en conjunto para lograr hacer un juego de realidad aumentada. Este proyecto está pensado para el apoyo de las Matemáticas a nivel superior, en específico a las materias de Geometría Analítica y Cálculo Diferencial, en donde se busca reforzar los temas de superficies, secciones cónicas y cálculo de volumen. Dichos temas son los títulos de cada juego desarrollado.

A continuación se explicará la construcción de cada uno de los juegos a manera que puedan servir como plantilla para hacerlos más dinámicos, o mejor aún, construir nuevos juegos basados en éstos.

Como se ha mencionado, Vuforia será el software encargado de reconocer las imágenes del mundo real y posicionar la cámara para la correcta visualización de los modelos en 3D. Para ello será necesario descargar el SDK de Vuforia, disponible en su página oficial, el cual trabajará en conjunto con Unity. Bastará con importar a un proyecto, previamente creado en Unity, el paquete que se ha descargado de Vuforia para poder trabajar en el desarrollo de la aplicación.

Una vez que se cargaron todos los archivos al proyecto, habrá que identificar elementos, o prefabs, como: `VirtualButton`, `ImageTarget` y `ARCamera`, que serán necesarios para la Realidad Aumentada y forman parte del SDK de Vuforia. Para usar correctamente estos elementos se tendrán que configurar tanto la `ARCamera`, añadiendo un código otorgado por los desarrolladores de Vuforia, como el `ImageTarget`, para seleccionar la imagen que será reconocida por la cámara, todas esas indicaciones se encuentran en la página de Vuforia y pueden ser consultadas en cualquier momento. En el caso de los botones virtuales, no hay más que identificar el script asociado a dichos botones para poder trabajar con ellos.

Cabe aclarar que los botones que se agreguen tendrán que estar asociados a un ImageTarget, es decir, los botones tendrán que ser hijos de un ImageTarget, y el script para trabajar con ellos es: VirtualButtonEventHandler, también parte del SDK Vuforia, con este script se podrá controlar las acciones de cada botón virtual.

En un principio, dentro del proyecto de Unity, se han creado cuatro escenas, las cuales corresponden a: menú de inicio, superficies, secciones cónicas y volumen máximo. A continuación se explicarán cada una de las escenas, así como los componentes necesarios para: visualizar modelos, mostrar paneles de usuario, interactuar con los elementos, etc. La explicación se realizará a modo de que algún desarrollador con ciertos conocimientos del software Unity, pueda utilizar éstos como plantilla para desarrollar juegos similares, o simplemente cambiar o agregar modelos 3D, botones virtuales, o código necesario para mejoras en el juego.

3.4.1 Escena uno: Menú de inicio

Al momento de abrir la aplicación lo primero que se muestra es un panel que contiene el nombre de la misma, así como un párrafo explicando la finalidad de los juegos aquí desarrollados. Ver figura 3.1.

Juegos de Realidad Aumentada para el apoyo en Matemáticas

Con estos juegos se busca apoyar y reforzar los conocimientos adquiridos en materias como Geometría Analítica y Cálculo I, del departamento de Matemáticas de la División de Ciencias Básicas de la Facultad de Ingeniería.

Se recomienda leer con atención las instrucciones para cada juego.

¡SUERTE!

Entendido

Figura 3.1 Panel que muestra el título de la aplicación, la finalidad de los juegos y las materias a las que están enfocados.

Al presionar el botón de Entendido, desaparecerá el panel y se mostrará otro, el cual está destinado a que el usuario entienda el modo de interacción de esta aplicación de Realidad Aumentada. Ver figura 3.2.

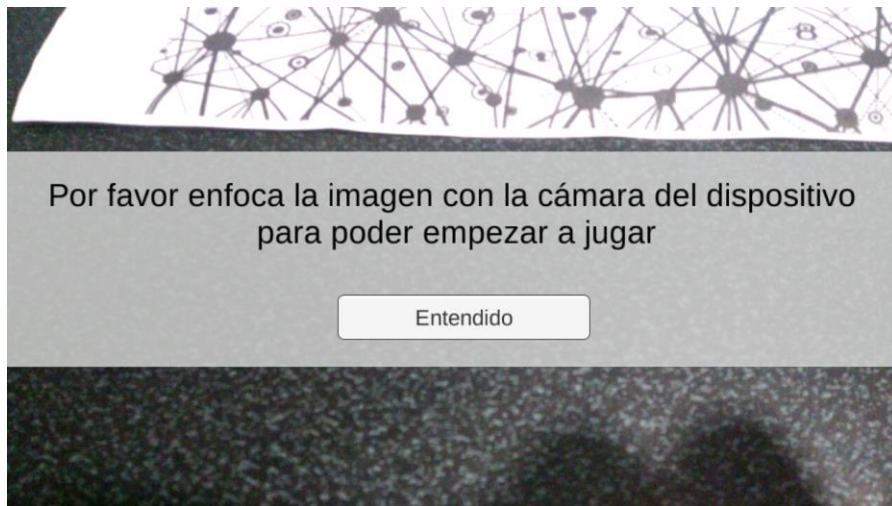


Figura 3.2 Se pide al usuario enfocar con la cámara del dispositivo la imagen que mostrará los objetos de Realidad Aumentada.

Una vez entendido el modo en que se pueden visualizar los modelos y botones virtuales se muestran textos indicando para qué sirve cada uno de los botones asociados a la imagen, o target, del menú de inicio. Ver figura 3.3.

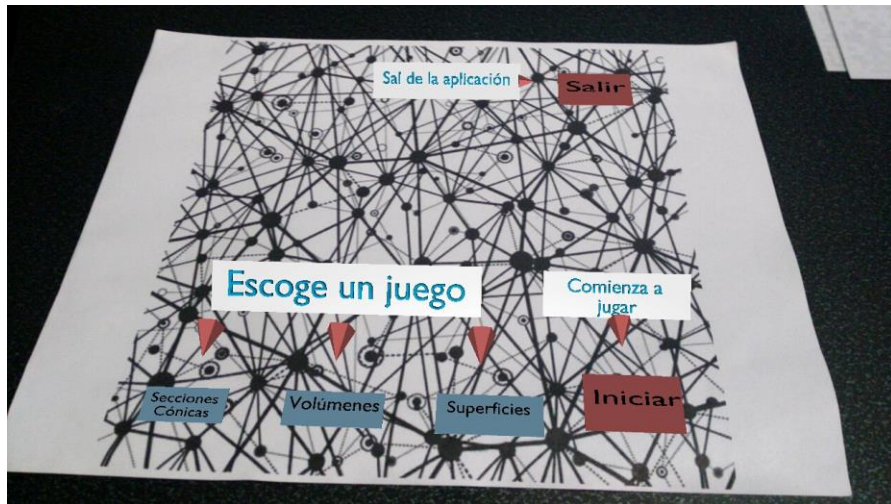


Figura 3.3 Indicaciones de funcionalidad de cada botón.

Las indicaciones de la función para cada botón son textos en 3D que se pueden cargar y posicionar fácilmente en cualquier punto de la tarjeta, claro está, siempre indicando el funcionamiento de algún objeto presente en la misma, éstas indicaciones desaparecerán al “presionar” alguno de los botones, dando por hecho que se ha comprendido el funcionamiento de los mismos.

Dentro del script existe la función llamada `OnButtonPressed`, la cual será llamada cada que un botón sea activado, la función recibe como parámetro el valor del botón y con un `switch`, obtenemos el nombre del botón y enseguida la acción asociada al botón seleccionado. A continuación se muestra el código usado para el desarrollo del juego.

Código

```
public void OnButtonPressed(VirtualButtonAbstractBehaviour vb
) {

    switch(vb.VirtualButtonName) {
        case "JuegoUno":
            //Code...
            break;
        case " JuegoDos":
            //Code...
            break;
        case " JuegoTres":
            //Code...
            break;
        case "Iniciar":
            //Code...
            break;
        case "Salir":
            //Code...
            break;
        default:
            throw new UnityException("Button not supported:");
            break;
    }
}
```

Dentro del switch, los primeros tres casos: JuegoUno, JuegoDos y JuegoTres, pertenecen a los juegos o escenas desarrollados, el cuarto caso: Iniciar, cargará la escena correspondiente al juego seleccionado previamente, y para el quinto caso: Salir, simplemente se cierra la aplicación.

De las figuras 3.4 a la 3.6 se muestran las acciones relacionadas a los tres primeros casos, a cada botón le corresponde un modelo 3D con un nombre y un objeto relacionado al juego mismo.

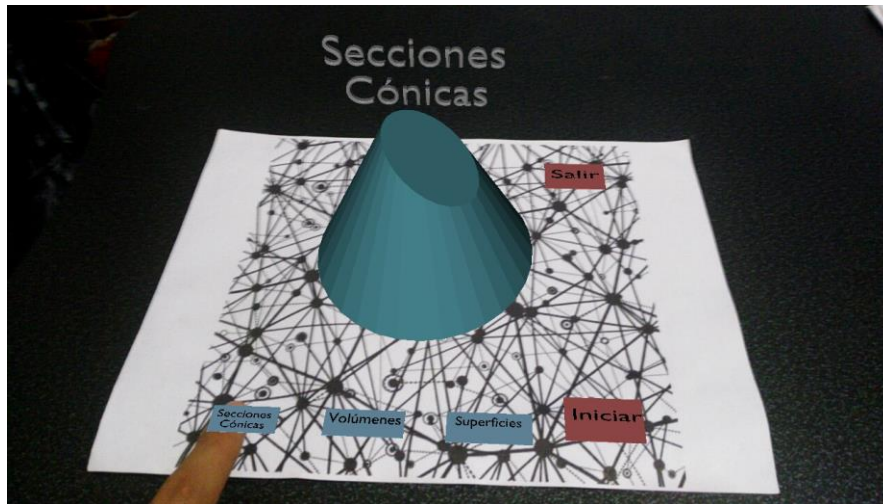


Figura 3.4 Primer caso del switch y el modelo 3D que se carga al presionar el primer botón.

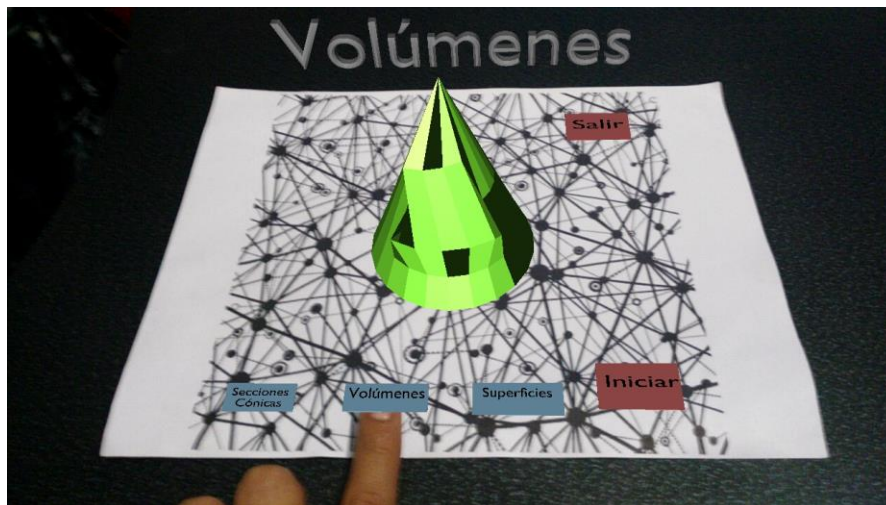


Figura 3.5 Segundo caso del switch y el modelo 3D que se carga al presionar el segundo botón.



Figura 3.6 Tercer caso del switch y el modelo 3D que se carga al presionar el tercer botón.

Ya que se tomó la decisión de la interacción con el juego, lo que sigue es ir a la escena del juego seleccionado, esto con la ayuda del botón virtual: Iniciar, ver figura 3.7, el cual se encargará de guardar el nombre correspondiente al juego seleccionado y mostrará un panel en el que se asegure la elección del usuario, con la opción de regresar a elegir otro juego o continuar con el ya seleccionado. Ver figura 3.8.

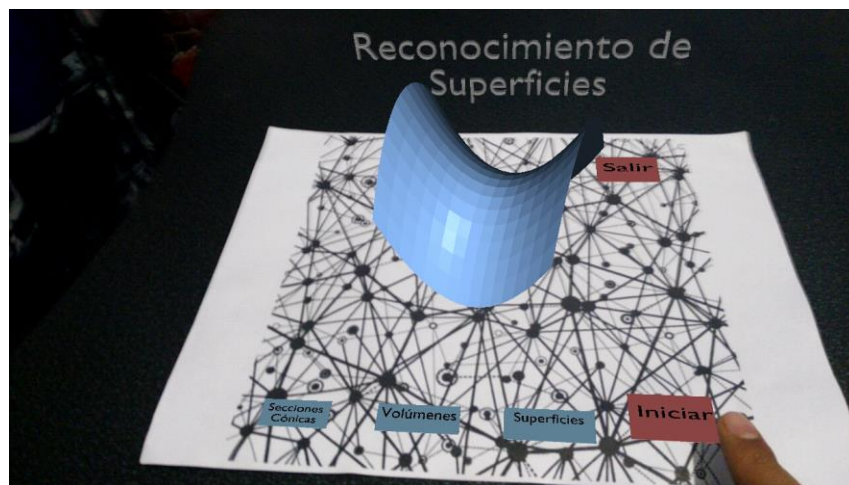


Figura 3.7 El caso de la elección del botón Iniciar.

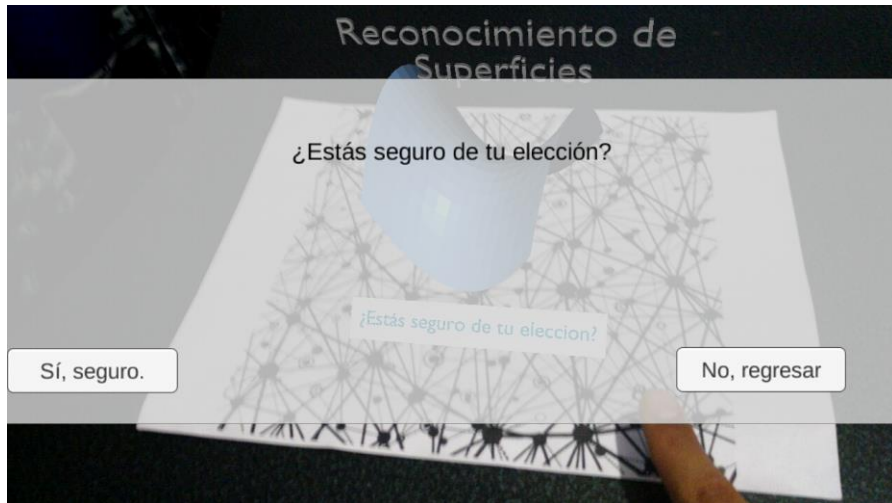


Figura 3.8 Panel para confirmar que el usuario esté totalmente seguro del juego que jugará.

Si el usuario está seguro de querer jugar la opción seleccionada, entonces se mostrará un panel en donde tendrá que ingresar los datos necesarios para poder jugar. Ver figura 3.9.

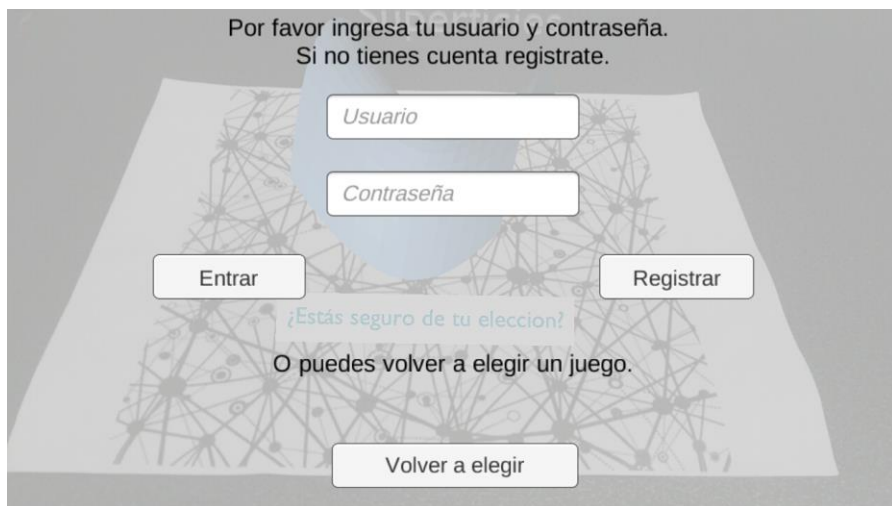
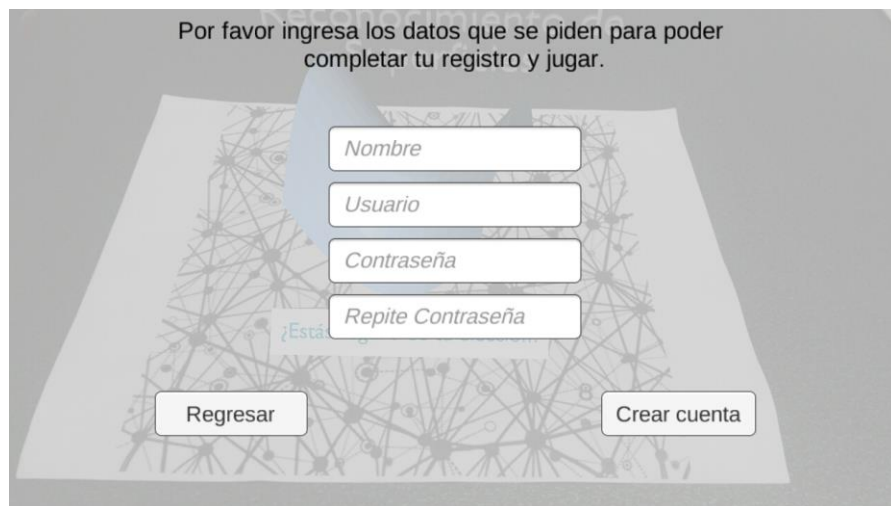


Figura 3.9 Panel para el ingreso de datos. Con la opción de volver a elegir el juego.

Los datos que aquí se piden son básicamente para tener un control de quién es la persona que juega, aquí existen dos alternativas: Entrar y Registrar. La primera es para usuarios que ya estén dados de alta en la base de datos y que, obviamente, ya pasaron por la opción de registro, en este caso solo es el nombre del alumno y su número de cuenta los datos que se piden para poder ingresar al juego. La segunda opción, Registrar, es para usuarios que juegan por primera vez en esta aplicación y que no tienen registro alguno en la base de datos del juego. Ver figura 3.10.



Por favor ingresa los datos que se piden para poder completar tu registro y jugar.

Nombre

Usuario

Contraseña

Repite Contraseña

Regresar

Crear cuenta

¿Está...?

Figura 3.10 Panel de registro para dar de alta al usuario en la base de datos.

La información de estos datos es necesaria para poder llevar un registro de los alumnos y las calificaciones que obtienen en los respectivos juegos, así como el día en que jugaron y el tiempo que tardaron en realizar el ejercicio. Lo anterior ayudará al profesor a tener una bitácora de cada alumno suyo que ingrese a la

aplicación a jugar y con ello poder evaluar el avance de conocimientos y habilidades de cada uno de sus alumnos.

Al llenar los datos necesarios ya sea para darse de alta o ingresar, se cargará la escena del juego seleccionado, cada juego cuenta con un panel al inicio de la escena, el cual tiene instrucciones específicas para jugar de manera correcta el juego en curso.

3.4.2 Escena dos: Juego de Superficies

Se deberá identificar la superficie que se indique en la pantalla basado en la ecuación de la misma, es decir, cada tarjeta contiene una ecuación asociada a una superficie, se deberá identificar la ecuación que corresponda a la superficie dada en el problema. Ver figura 3.11.

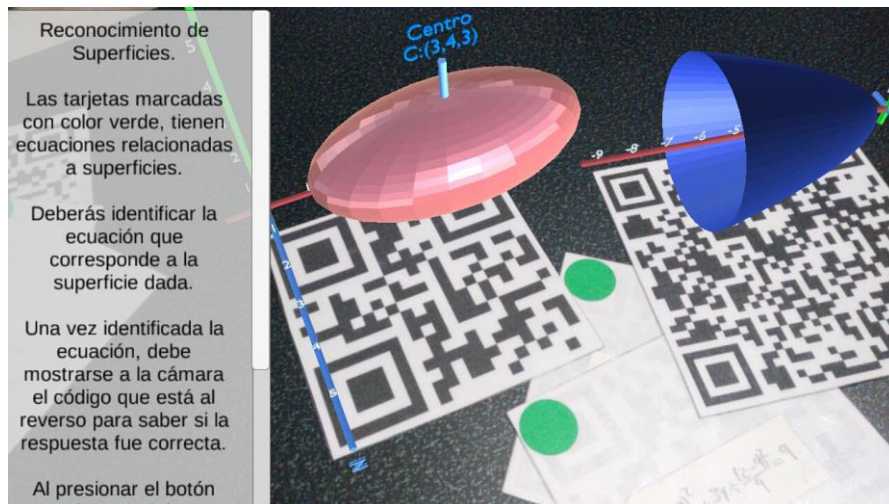


Figura 3.11 Panel de entrada del juego de Superficies, así como algunas de las superficies a reconocer.

Para este juego, se crearon diversos modelos 3D en el software Blender, modelos relacionados a superficies que son materia de estudio en la materia de Geometría Analítica. Tales superficies vienen dadas por ecuaciones, las cuales pueden ser introducidas tal cual a Blender y el mismo software genera la superficie relacionada a esa ecuación, existen superficies como la esfera o el elipsoide que vienen por defecto en el software en modo de objetos primitivos y no es necesaria su ecuación para poder generarlos.

Al exportar a Unity los modelos generados en Blender, se debe asociar cada modelo a un ImageTarget para poder visualizarlo en tiempo real, es decir, cada ImageTarget tendrá un objeto como hijo, en este caso una superficie, así como también un BoxCollider, elemento necesario para la detección de colisiones entre objetos.

La cámara de este juego cuenta con un objeto, como hijo, llamado Validador, al cual se le asocia un script llamado ObjectCollision, y también un BoxCollider. Dentro del script mencionado existe la función OnTriggerEnter, que es la encargada de detectar cuando existe una colisión entre objetos, uno de los cuales debe previamente ser definido como Trigger, en este caso será el objeto Validador el que espere a que algún otro colisione con él y active la función.

Código

```
void OnTriggerEnter(Collider other){

    NombreCurva = other.gameObject.name;
    switch (NombreCurva){
        case "Esfera":
            if(CurvaProblema == NombreCurva){
                Correcto();
            }else{
                Incorrecto ();}
            break;
        case "Elipsoide":
            if(CurvaProblema == NombreCurva){
                Correcto();
            }else{
                Incorrecto ();}
            break;
        case "Hiperboloide de doble manto":
            if(CurvaProblema == NombreCurva){
                Correcto();
            }else{
                Incorrecto ();}
            break;
        case "Paraboloide Circular":
            if(CurvaProblema == NombreCurva){
                Correcto();
            }else{
                Incorrecto ();}
            break;
        case "Hiperboloide un manto":
            if(CurvaProblema == NombreCurva){
                Correcto();
            }else{
                Incorrecto ();}
            break;
        case "Cilindro Eliptico":
            if(CurvaProblema == NombreCurva){
```

```

        Correcto();
    }else{
        Incorrecto ();}
    break;
case "Paraboloide Hiperbolico":
    if(CurvaProblema == NombreCurva){
        Correcto();
    }else{
        Incorrecto ();}
    break;
default:
    Debug.Log("Not Found.");
    break;
}
}

```

Con la función anterior se busca comparar el nombre de la curva que se ha dado en el problema, CurvaProblema, contra el nombre de la curva que ha elegido el usuario, NombreCurva, esto con la ayuda de un switch que ejecutará las instrucciones correspondientes al caso de la curva seleccionada por el usuario, y con la ayuda de un if sabremos si su respuesta fue correcta o no al comparar el nombre de las diferentes curvas. Con ello podemos llamar a diversas funciones dependiendo del resultado obtenido.

Para decidir la curva que ha de identificarse, es decir la CurvaProblema, se utilizó la clase random, que sirve para generar números aleatorios, y nuevamente con ayuda de un switch se elige el nombre de la curva dependiendo el número que arroje la clase random.

Código

```
int superficieEleccion = Random.Range (1,8);

switch(superficieEleccion){
    case 1:
        CurvaProblema = "Paraboloide Hiperbolico";
        break;
    case 2:
        CurvaProblema = "Elipsoide";
        break;
    case 3:
        CurvaProblema = "Paraboloide Circular";
        break;
    case 4:
        CurvaProblema = "Esfera";
        break;
    case 5:
        CurvaProblema = "Hiperboloide de doble manto";
        break;
    case 6:
        CurvaProblema = "Cilindro Eliptico";
        break;
    case 7:
        CurvaProblema = "Hiperboloide un manto";
        break;
    default:
        Debug.Log ("Not Found");
        break;
}
```

Para el código anterior, se ha declarado una variable entera, la cual será definida por la clase random antes mencionada, esta variable tendrá un valor definido entre 1 y 8, que son el número de curvas que se tienen para identificar, más el caso default del switch. El valor de superficieEleccion, será aleatorio desde que se carga la escena hasta que el usuario oprima el botón de Continuar para

68

comenzar a jugar, esto detendrá la clase `random` y el valor que se obtenga será lo que defina el nombre de la curva a reconocer por el usuario, la `CurvaProblema`.

Tanto el tiempo como el nombre de la curva del problema, serán mostrados en la pantalla del dispositivo una vez que el usuario leyó con atención las instrucciones y presionó el botón de `Continuar`. Con la ayuda de un `Canvas`, se define el panel que contendrá los datos, y bastará con asociar el tiempo y el nombre de la curva generados en el `script` con el panel que se mostrará en pantalla. Ver figura 3.12.

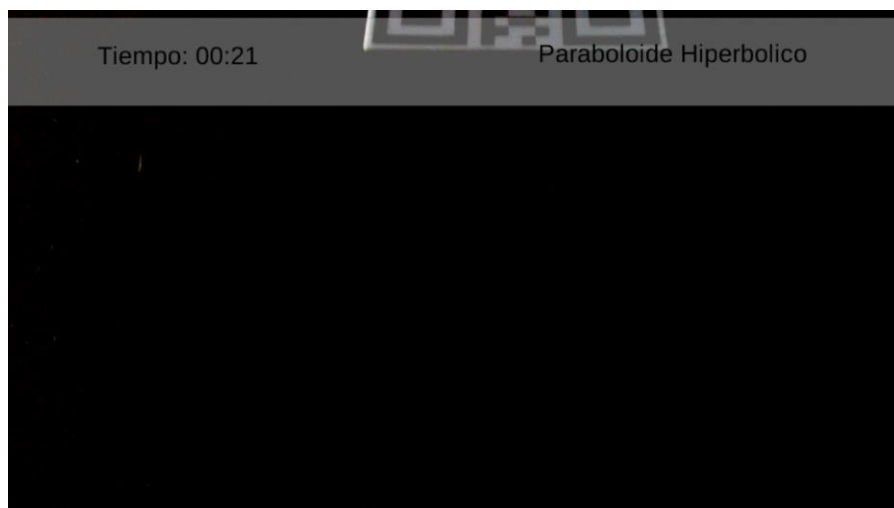


Figura 3.12 Panel que muestra el tiempo desde que se presionó `Continuar`, así como la superficie que tendrá que reconocer el usuario.

Para la parte de resultados, se tiene un panel que mostrará el texto correspondiente al resultado obtenido por el usuario. Como se mostró, el `switch` que detecta las colisiones, contiene instrucciones para cada caso del nombre de la curva, y dentro de cada caso se cuenta con llamadas a funciones que definen la respuesta correcta e incorrecta.

Código

```
public void Correcto(){

    resultado = 10;
    switch (NombreCurva){
        case "Esfera":
            Resultado.text ="Correcto.
            Tu calificación será:
            "+resultado+"Tu tiempo: "+textTime+" y califi
            cación quedarán guardados";
            break;
        case "Elipsoide":
            Resultado.text ="Correcto.
            Tu calificación será:
            "+resultado+"Tu tiempo: "+textTime+" y califi
            cación quedarán guardados";
            break;
        case "Hiperboloide de doble manto":
            Resultado.text ="Correcto.
            Tu calificación será:
            "+resultado+"Tu tiempo: "+textTime+" y califi
            cación quedarán guardados";
            break;
        case "Paraboloide Circular":
            Resultado.text ="Correcto.
            Tu calificación será:
            "+resultado+"Tu tiempo: "+textTime+" y califi
            cación quedarán guardados";
            break;
        case "Hiperboloide un manto":
            Resultado.text ="Correcto.
            Tu calificación será:
            "+resultado+"Tu tiempo: "+textTime+" y califi
            cación quedarán guardados";
            break;
        case "Cilindro Eliptico":
```

```

        Resultado.text ="Correcto.
        Tu calificación será:
        "+resultado+"Tu tiempo: "+textTime+" y califi
        cación quedarán guardados";
        break;
    case "Paraboloide Hiperbolico":
        Resultado.text ="Correcto.
        Tu calificación será:
        "+resultado+"Tu tiempo: "+textTime+" y califi
        cación quedarán guardados";
        break;
    default:
        Debug.Log("Correcto.");
        break;
    }
}

```

```

public void Incorrecto(){

    resultado = 5;
    switch (NombreCurva){
        case "Esfera":
            Resultado.text ="Incorrecto.
            Tu calificación será:
            "+resultado+"Tu tiempo: "+textTime+" y califi
            cación quedarán guardados";
            break;
        case "Elipsoide":
            Resultado.text ="Incorrecto.
            Tu calificación será:
            "+resultado+"Tu tiempo: "+textTime+" y califi
            cación quedarán guardados";
            break;
        case "Hiperboloide de doble manto":
            Resultado.text ="Incorrecto.
            Tu calificación será:
            "+resultado+"Tu tiempo: "+textTime+" y califi
            cación quedarán guardados";
            break;
    }
}

```



```

    case "Paraboloide Circular":
        Resultado.text ="Incorrecto.
        Tu calificación será:
        "+resultado+"Tu tiempo: "+textTime+" y califi
        cación quedarán guardados";
        break;
    case "Hiperboloide un manto":
        Resultado.text ="Incorrecto.
        Tu calificación será:
        "+resultado+"Tu tiempo: "+textTime+" y califi
        cación quedarán guardados";
        break;
    case "Cilindro Eliptico":
        Resultado.text ="Incorrecto.
        Tu calificación será:
        "+resultado+"Tu tiempo: "+textTime+" y califi
        cación quedarán guardados";
        break;
    case "Paraboloide Hiperbolico":
        Resultado.text ="Incorrecto.
        Tu calificación será:
        "+resultado+"Tu tiempo: "+textTime+" y califi
        cación quedarán guardados";
        break;
    default:
        Debug.Log("Incorrecto.");
        break;
}
}

```

Para cada función se define la calificación obtenida, 5 ó 10, así como el tiempo que ha tardado el usuario en identificar la curva, estos datos se anexan a un texto correspondiente al panel de resultados, presente en la escena. El texto de resultados puede ser modificado desde el script, y será visible al detectarse la colisión entre el objeto validador y la curva elegida por el usuario. Ver figura 3.13.

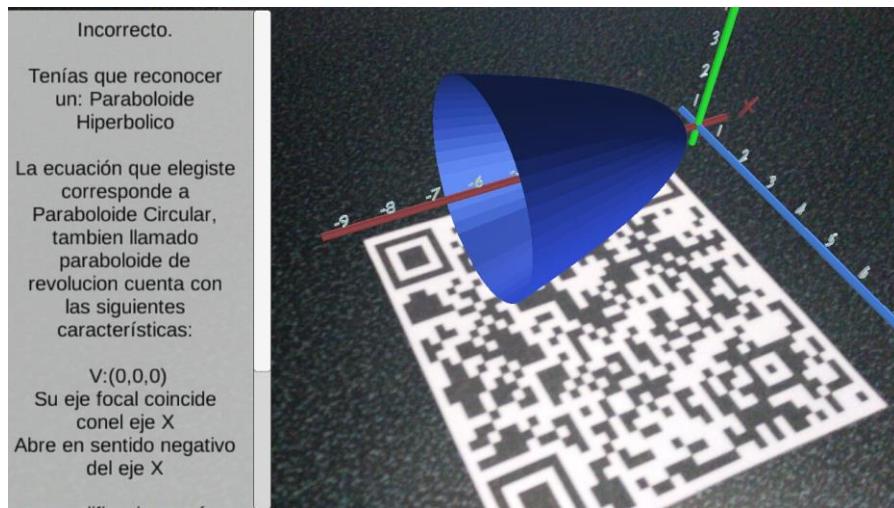


Figura 3.13 Panel de resultados, mostrando los datos de la superficie que eligió el usuario, así como el tiempo que tardó y el resultado obtenido.

3.4.3 Escena tres: Juego de Secciones Cónicas

En este juego se debe identificar el plano inclinado que cortará a un cono de tal forma que la superficie recortada del cono sea la que se indica en el problema. Al cargarse esta escena, lo primero que se muestra es un panel con indicaciones sobre el juego, y al igual que el juego anterior, se da por hecho que al presionar el botón de Continuar se ha entendido por completo las indicaciones. Ver figura 3.14.

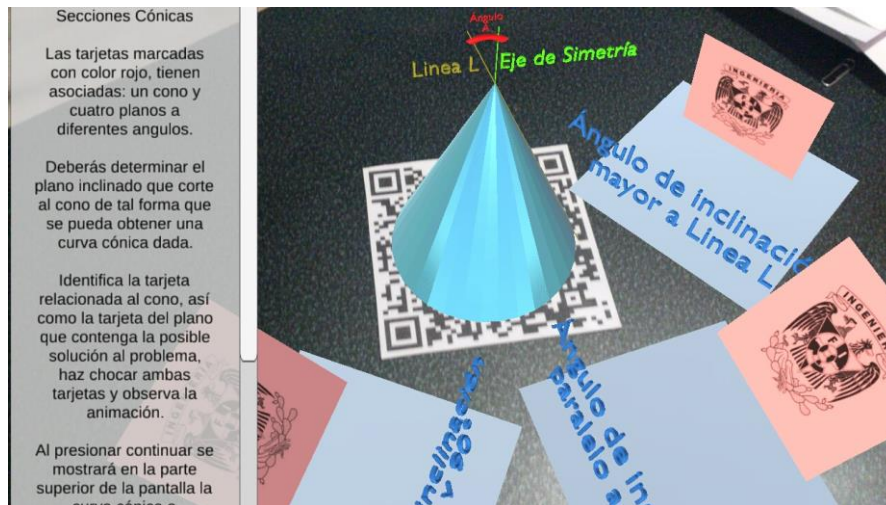


Figura 3.14 Panel de entrada del juego de Secciones Cónicas, así como los modelos 3D presentes en la escena.

En esta escena se han cargado nueve modelos 3D, que corresponden a: cuatro conos cortados a diferentes ángulos para observar las curvas cónicas, cuatro planos inclinados a diferentes ángulos, y un cono entero que será cortado por esos planos. Tanto el cono entero como los conos ya cortados, serán todos hijos de un ImageTarget, y los cuatro planos inclinados tendrán cada uno su propia ImageTarget.

Al igual que en el juego anterior se utilizarán Colliders para cada una de las ImageTarget, esto para detectar la colisión, en este caso, entre los diferentes planos y el cono completo, o simplemente, cono.

También se utiliza la clase random para decidir la curva cónica, CurvaConica, que hay que identificar basado en el plano inclinado, y nuevamente con ayuda de un

switch se elige el nombre de la curva cónica dependiendo el número que arroje la clase random.

Código

```
int ConicaAleatoria = Random.Range (1,4);
switch(ConicaAleatoria){
    case 1:
        CurvaConica = "Circunferencia";
        break;
    case 2:
        CurvaConica = "Elipse";
        break;
    case 3:
        CurvaConica = "Parábola";
        break;
    case 4:
        CurvaConica = "Hipérbola";
        break;
    default:
        Debug.Log ("Se define la curva problema");
        break;
}
```

Ya que se tiene definida la curva que tendrá que generar el usuario, se guarda en la variable CurvaConica y se muestra en un panel junto con el tiempo que correrá a partir de que el usuario presione Continuar. Ver figura 3.15.

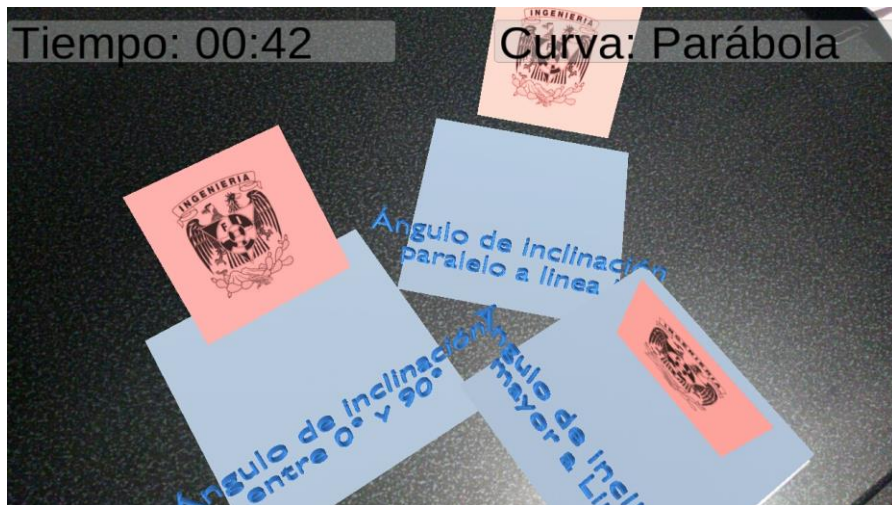


Figura 3.15 Panel que contiene el tiempo y la curva a generar, así como los planos que cortarán al cono para generar la curva cónica.

Por otro lado, el cono será el encargado de esperar a que algún plano colisione con él, esto para activar la función que mueva al respectivo plano y corte al cono de tal forma que mostrará una superficie cónica. El script asociado al cono es llamado MovePlane y contiene la función OnTriggerEnter.

Código

```
void OnTriggerEnter(Collider other){

    NombrePlano = other.gameObject.name;
    switch (NombrePlano) {
        case "PlanoUnoCircunf":
            Circunferencia.SetActive(true);
            muevePlanoCircunf = true;
            break;
        case "PlanoDosElipse":
            Elipse.SetActive(true);
            muevePlanoElipse = true;
            break;
        case "PlanoTresParabola":
            Parabola.SetActive(true);
    }
}
```

```

        muevePlanoParabola =true;
    break;
case "PlanoCuatroHiperbola":
    Hiperbola.SetActive(true);
    muevePlanoHiperbola =true;
    break;
default:
    break;
}
}

```

Con la función anterior, obtenemos el nombre del plano que ha colisionado con el cono, activamos el cono ya cortado correspondiente al plano seleccionado, y cambiamos el valor de la bandera muevePlanoX, esto para comenzar la animación del corte entre el plano seleccionado y el cono. Ver figura 3.16.

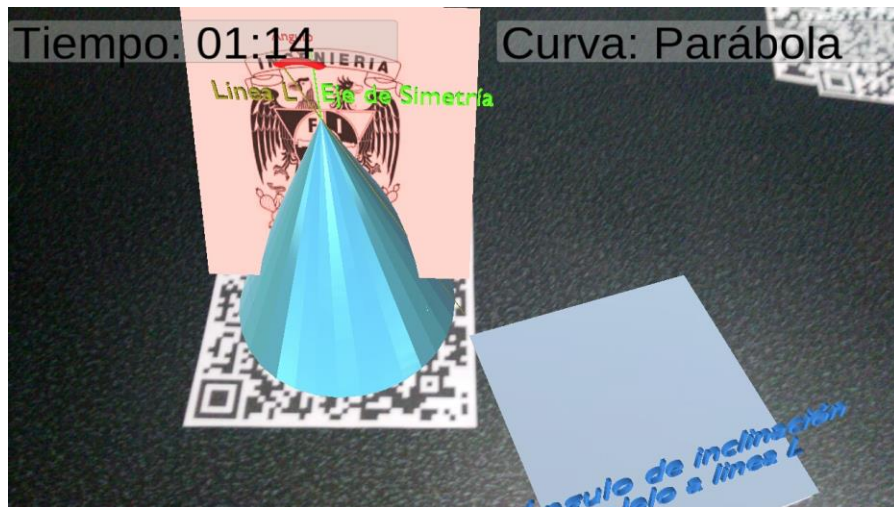


Figura 3.16 Parte de la animación entre el plano seleccionado y el cono que será cortado por ese plano.

En el mismo script usamos la función Update para estar leyendo constantemente el valor de la variable muevePlanoX.

Código

```
void Update () {  
  
    if (muevePlanoCircunf == true){  
        Plano.transform.Translate (200*Time.deltaTime, 0  
        , 0);  
    }  
    if (muevePlanoElipse == true){  
        PlanoElipse.transform.Translate (0,-  
        1*2*Time.deltaTime, -1*105*Time.deltaTime);  
    }  
    if (muevePlanoParabola == true) {  
        PlanoParabola.transform.Translate (0,-  
        1*2*Time.deltaTime, -1*105*Time.deltaTime);  
    }  
    if (muevePlanoHiperbola == true) {  
        PlanoHiperbola.transform.Translate (0, -  
        1*2*Time.deltaTime, -1*105*Time.deltaTime);  
    }  
}
```

Una vez que la función OnTriggerEnter ha cambiado el valor de muevePlanoX, dentro de la función Update se identificará de qué plano se trata con ayuda de un if, y se procede a moverlo en dirección al ángulo del mismo, de modo que pueda cortar al cono y mostrar la curva asociada al plano.

Terminada la animación del plano cortando al cono, se mostrará el resultado en un panel gracias a la función OnTriggerExit que, al contrario de los Triggers anteriores, éste detecta el término de la colisión entre objetos, es decir, se activa cuando los objetos dejan de colisionar entre sí.

Código

```
void OnTriggerExit ( Collider other){  
  
    Destroy(this.gameObject);  
    if (NombrePlano == CurvaConica) {  
        kalif = 10;  
        switch (NpmbrePlano){  
            case "Circunferencia":  
                RCorrectaTxt = "Correcto. Queríamos obtener una "+curvaProblema+"Y tu obtuviste una "+NombrePlano+" Tiempo"+CortaTiempoT  
xt.textTime+"Calificación"+kalif+"Tu tiempo y calificación quedarán guardados";  
                break;  
            case "Elipse":  
                RCorrectaTxt = "Correcto. Queríamos obtener una "+curvaProblema+"Y tu obtuviste una "+NombrePlano+" Tiempo"+CortaTiempoT  
xt.textTime+"Calificación"+kalif+"Tu tiempo y calificación quedarán guardados"  
                break;  
            case "Parábola":  
                RCorrectaTxt = "Correcto. Queríamos obtener una "+curvaProblema+"Y tu obtuviste una "+NombrePlano+" Tiempo"+CortaTiempoT  
xt.textTime+"Calificación"+kalif+"Tu tiempo y calificación quedarán guardados"  
                break;  
            case "Hipérbola":  
                RCorrectaTxt = "Correcto. Queríamos obtener una "+curvaProblema+"Y tu obtuviste una "+NombrePlano+" Tiempo"+CortaTiempoT  
xt.textTime+"Calificación"+kalif+"Tu tiempo y calificación quedarán guardados";  
                break;  
            default:
```



```

        break;
    }
} else {
    kalif = 5;
    RIncorrectaTxt = "Incorrecto. Queríamos obtener un
a "+curvaProblema+"Y tu obtuviste una "+OtherCurva
+"Tiempo"+CortaTiempoTxt.textTime+"Calificación"+k
alif+"Tu tiempo y calificación quedarán guardados"
;
}
}

```

Al activarse el OnTriggerExit lo primero es destruir el cono completo, dejando así al descubierto el cono cortado, activado anteriormente, correspondiente a ese plano y, de nuevo, con la ayuda de un if, se hace la comparación entre el nombre de las curvas.

Por un lado tenemos el nombre del plano que ha elegido el usuario, NombrePlano, mismo que será comparado con el nombre de la curva que se tiene que identificar, CurvaConica, dada aleatoriamente por la clase random.

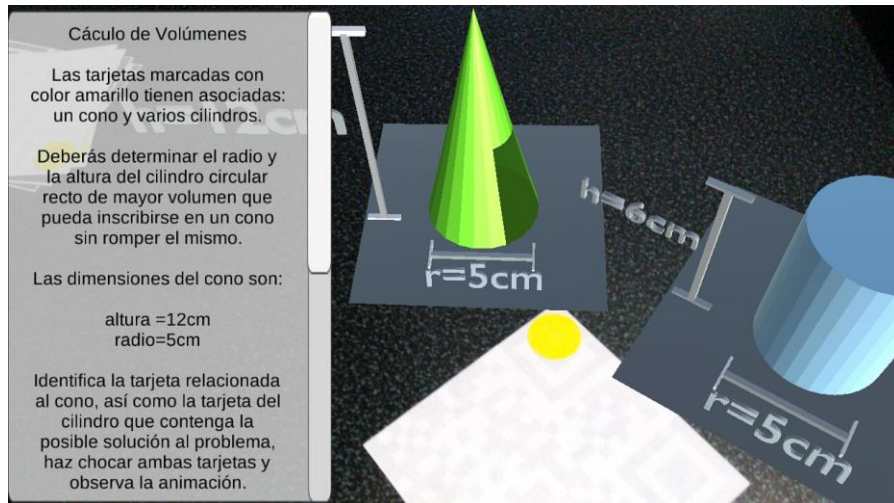
Si la comparación es cierta, se define el valor de la calificación, igual a 10, y se pasa a un switch, en donde para cada caso existen instrucciones específicas; de lo contrario, si la comparación es falsa, simplemente se define el valor de la calificación, igual a 5, y para ambos casos se genera un texto con la calificación, tiempo y resultado de la elección, todo esto será mostrado en el panel de resultados presente en la escena. Ver figura 3.17.



Figura 3.17 Panel de resultados y el fin de la animación de corte al cono.

3.4.4 Escena cuatro: Juego de Cálculo de Volumen Máximo

Al cargar esta última escena se muestra un panel con indicaciones para este juego, cuyo objetivo es calcular el máximo volumen de un cilindro que pueda estar inscrito en el cono del problema, si las dimensiones son más grandes de lo debido, el cono se romperá, si son más chicas, no se romperá pero tampoco será el volumen máximo que pide el problema. El alumno deberá efectuar sus cálculos de modo que pueda elegir la tarjeta que contiene las dimensiones exactas del cilindro. Ver figura 3.18.



3.18 Panel de entrada del juego de Cálculo de Volumen Máximo, así como algunos modelos 3D cargados en la escena.

Los modelos 3D utilizados para este juego son: dos conos con ciertas alteraciones en su estructura, uno de ellos con un hueco en uno de sus lados, y otro con el mismo hueco más algunas aristas fuera de lugar. Aunado a esos conos, se generaron seis cilindros con diferentes medidas en anchura y altura, y para todos los modelos se hicieron textos en 3D indicando tanto el radio como la altura cada uno de ellos.

Para este juego se repite el concepto de que si se presiona el botón Continuar, al final del panel de entrada, se entiende que el usuario ha leído y comprendido las instrucciones de juego, por lo tanto aparecerán los paneles que tienen: el tiempo y los datos del cono del problema. Ver figura 3.19.

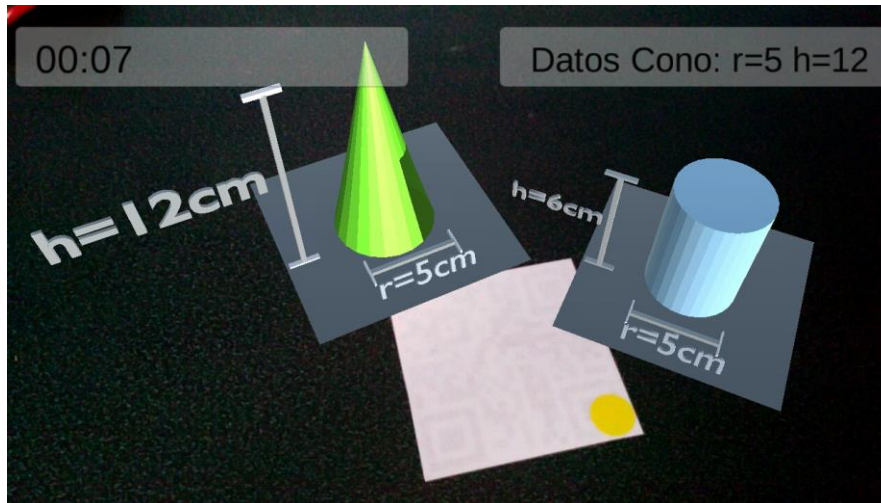


Figura 3.19 Panel que contiene el tiempo y los datos del cono, así como los modelos referentes al cono y a uno de los cilindros.

Lo mismo que en los juegos anteriores, utilizamos Triggers y Colliders para detectar la colisión de objetos y determinar el conocimiento del usuario. En este caso el cono con ciertas medidas de volumen: radio y altura, será el encargado de ejecutar las acciones correspondientes a la animación y mostrar el resultado en pantalla, todo esto a través de un script que hemos nombrado: ConoVolumen, el cual nos ayudará a identificar el cilindro que colisionará con el cono.

Código

```
void OnTriggerEnter(Collider other){

    nombre = other.gameObject.name;
    PanelDatos.SetActive (false);
    PanelTiempo.SetActive (false);
    PanelResultado.SetActive (true);

    switch (nombre) {
        case "Cilindro1":
```

```

    calif=5;
    DatosIncorrecto = "Incorrecto. El cilindro que escogiste rompió el cono"+Tiempo.text+"Los
    datos:
    "+DatosCono.text+"Calificación"+calif+"Tu tiempo y calificación serán guardados";
    PanelResultado.GetComponent<Text>().text = DatosIncorrecto;
    Cono.SetActive (false);
    conoRoto.SetActive (true);
    mueveCilindro=true;
break;
case "Cilindro2":
    calif=5;
    DatosIncorrecto = "Incorrecto. El cilindro que escogiste rompió el cono"+Tiempo.text+"Los
    datos:
    "+DatosCono.text+"Calificación"+calif+"Tu tiempo y calificación serán guardados";
    PanelResultado.GetComponent<Text>().text = DatosIncorrecto;
    Cono.SetActive (false);
    conoRoto.SetActive (true);
    mueveCilindro=true;
break;
case "Cilindro3":
    calif=5;
    DatosIncorrecto = "Incorrecto. El cilindro que escogiste rompió el cono"+Tiempo.text+"Los
    datos:
    "+DatosCono.text+"Calificación"+calif+"Tu tiempo y calificación serán guardados";
    PanelResultado.GetComponent<Text>().text = DatosIncorrecto;
    Cono.SetActive (false);
    conoRoto.SetActive (true);
    mueveCilindro=true;
break;
case "Cilindro4":

```

```

        calif=10;
        DatosCorrecto = "Correcto. El cilindro que es
        cogiste cabe          perfecto en          el cono
        "+Tiempo.text+"Los
        datos:"+DatosCono.text+"Calificación"+calif+"
        Tu tiempo y calificación serán guardados";
        PanelResultado.GetComponent<Text>().text = Da
        tosCorrecto;
        mueveCilindro=true;
    break;
case "Cilindro5":
    calif=5;
    DatosIncorrecto = "Incorrecto. El cilindro qu
    e escogiste rompió el cono"+Tiempo.text+"Los
    datos:
    "+DatosCono.text+"Calificación"+calif+"Tu tie
    mpo y calificación serán guardados";
    PanelResultado.GetComponent<Text>().text = Da
    tosIncorrecto;
    Cono.SetActive (false);
    conoRoto.SetActive (true);
    mueveCilindro=true;
    break;
case "Cilindro6":
    calif=5;
    DatosIncorrecto = "Incorrecto. El cilindro qu
    e escogiste rompió el cono"+Tiempo.text+"Los
    datos:
    "+DatosCono.text+"Calificación"+calif+"Tu tie
    mpo y calificación serán guardados";
    PanelResultado.GetComponent<Text>().text = Da
    tosIncorrecto;
    Cono.SetActive (false);
    conoRoto.SetActive (true);
    mueveCilindro=true;
    break;
}
}

```

De nuevo acudimos a un `OnTriggerEnter`, que identificará el nombre del cilindro que ha colisionado con el cono, al activarse la función lo primero es obtener el nombre del objeto que ha colisionado con el cono. Enseguida desactivamos los paneles respectivos al tiempo y a los datos del cono problema, y activamos el panel en donde se mostrarán los resultados.

Al entrar al `switch`, se evalúa el nombre obtenido anteriormente, y se ejecutan las instrucciones del caso correspondiente. En general, para cada caso, lo primero es definir la calificación obtenida por el usuario, previamente se ha decidido que la respuesta correcta será el objeto llamado `cilindro4`, que es el que contiene las medidas exactas para que se interseque dentro del cono sin romperlo, es por eso que para todos los casos la calificación será igual a 5, excepto el caso del `cilindro4` donde la calificación será de 10.

Ya definida la calificación, para cada caso se crea un mensaje que contendrá datos como la calificación, el tiempo de respuesta y los datos del cono del problema. Este mensaje será el que se muestre en el panel de resultados previamente activado en la escena. En el caso de una respuesta incorrecta, la siguiente instrucción es, desactivar el cono presente en la escena, y activar el cono que parece roto, esto para ejemplificar que el cilindro que se ha elegido no fue el correcto. Ver figura 3.20.

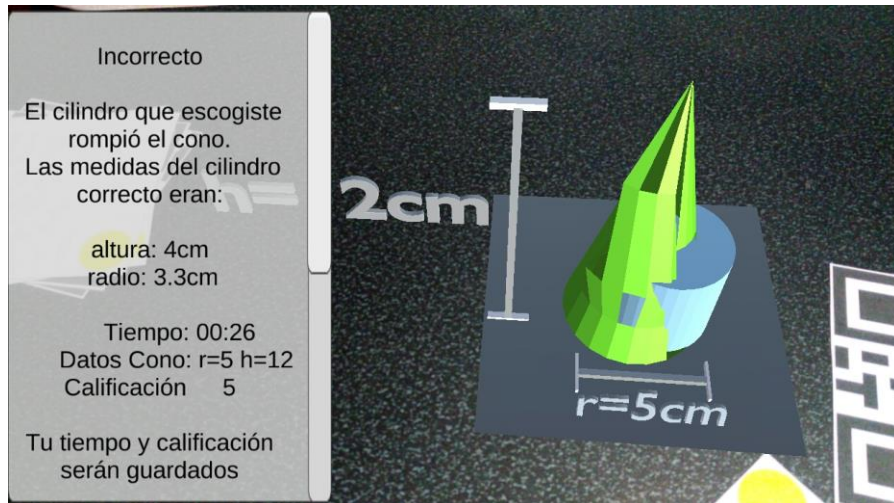


Figura 3.20 Panel de resultados, así como el fin de la animación entre el cono y el cilindro.

Por último, se cambia el estado de la bandera `mueveCilindro`, esto hará que en la función `Update` se ejecute la instrucción que haga mover al cilindro correspondiente hasta intersecar al cono problema.

Código

```

if (mueveCilindro == true) {
    Cilindro.transform.Translate (Vector3.left * 35 * Time.d
    eltaTime)
}

```

La acción se crea en la función `Update` ya que aquí las instrucciones se ejecutan una vez por frame, y dado que queremos simular el movimiento del cilindro hasta intersecar al cono, lo que se hace es dibujar el cilindro en una posición diferente en cada frame hasta llegar a incorporarse dentro del cono. Ver figura 3.21.

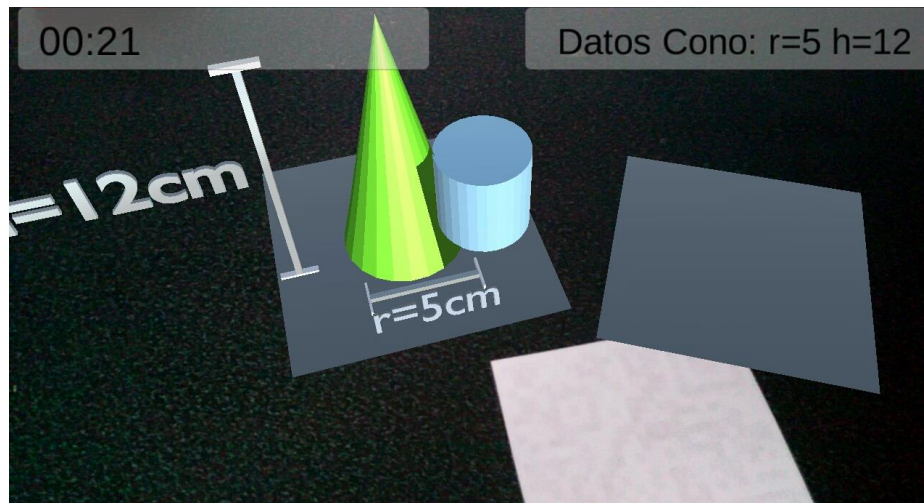


Figura 3.21 Parte de la animación al hacer colisionar el cono con el cilindro que ha escogido el usuario.

Hasta aquí la explicación del funcionamiento del último juego. Hay que tomar en cuenta que, aunque las mismas funciones existen en los tres juegos antes descritos, también existen diferencias que hacen que se deba explicar el funcionamiento de cada uno de ellos por separado.

Pero gracias a que los tres llegan al mismo punto que es el panel de resultados, a partir de ello podemos comenzar a explicar, de manera general, cómo insertar esos datos que fueron mostrados al usuario en la base de datos en la que se apoyará el profesor para evaluar el avance de los alumnos a su cargo.

3.5 Base de datos del Juego de Realidad Aumentada

Debido al tipo de juegos y el tipo de usuarios para los que se desarrollan éstos, se ha implementado una base de datos relacional, la cual se define básicamente

como un conjunto de tablas, formadas por filas, o registros, y columnas, o campos. En el modelo relacional de base de datos, las tablas comparten algún campo entre ellas. Estos campos compartidos van a servir para establecer relaciones entre las tablas que permitan consultas complejas (Sarría, 2006).

Por lo anterior se decidió utilizar una base de datos relacional, ya que se cuentan con diversas tablas que tienen campos compartidos, ver figura 3.22, y que serán el enlace entre ellas, a continuación se describe cada tabla, así como sus campos compartidos y su tarea dentro del juego.

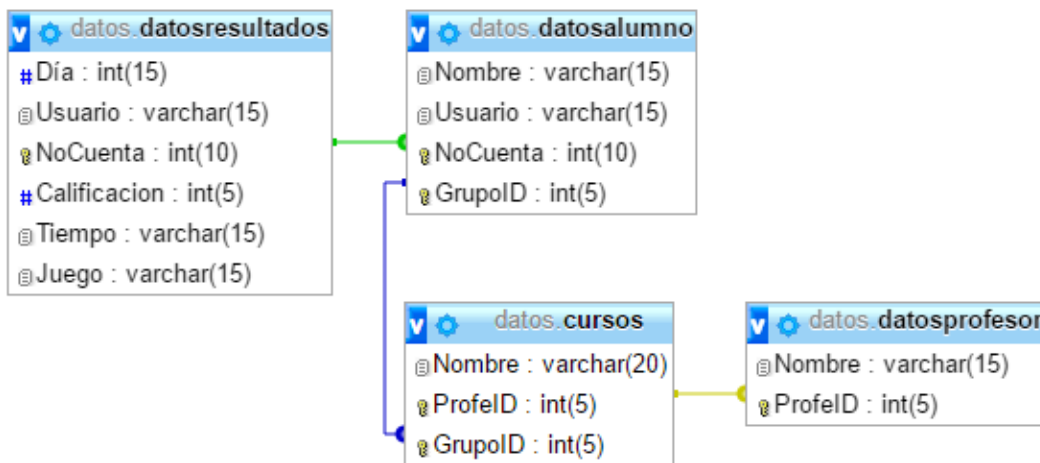
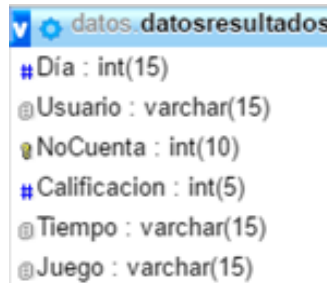


Figura 3.22 Diagrama entidad-relación de la base de datos.

Al final de cada juego se menciona que los datos quedarán guardados en la base de datos, los cuales corresponden a: el usuario, el número de cuenta del mismo usuario, la calificación que ha obtenido en el juego, el tiempo que tardó en resolverlo, el nombre del juego que ha jugado y la fecha en la que ha jugado.

Todos esos datos se almacenan en una tabla llamada DatosResultados. Ver figura 3.23.



The image shows a screenshot of a database interface displaying the structure of a table named 'datos.datosresultados'. The table has six columns with the following data types and constraints:

Column Name	Data Type	Constraint
Día	int(15)	Primary Key
Usuario	varchar(15)	Foreign Key
NoCuenta	int(10)	Index
Calificacion	int(5)	Primary Key
Tiempo	varchar(15)	Foreign Key
Juego	varchar(15)	Foreign Key

Figura 3.23 Tabla DatosResultados.

Al principio del juego se piden datos para el registro de usuarios, y es aquí donde viene la relación entre las diferentes tablas, por un lado tenemos la tabla DatosResultados antes mencionada, y por otro tenemos una tabla que corresponde al registro de los datos del alumno, la cual será llamada DatosAlumno, y esta cuenta con los siguientes campos: nombre del alumno, número de cuenta y grupo al que pertenece. Podemos observar que existe un campo idéntico en las dos tablas antes mencionadas, éste es el número de cuenta del alumno, o usuario. Este campo será nuestra primera llave primaria, ya que con ella podemos identificar fácilmente tanto los datos del alumno como sus resultados en los juegos. Ver figura 3.24.

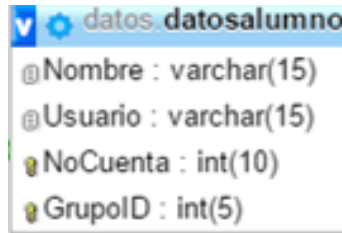


Figura 3.24 Tabla DatosAlumno.

Otra tabla importante es la de Cursos, en ella encontramos los diferentes cursos para los que se han desarrollado estos juegos, en éste caso: Geometría Analítica, en donde se encuentran los temas de Secciones Cónicas y Superficies Cuadráticas, y el curso de Cálculo Diferencial al que pertenece el tema de Volúmenes. En esta tabla se guardarán los datos del curso, tales como: Nombre del curso, el ID del grupo del curso, y el ID del profesor que imparte tal curso. Ver figura 3.25.

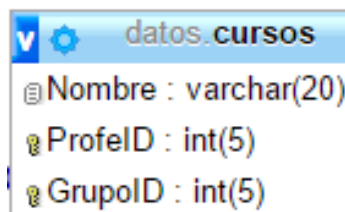


Figura 3.25 Tabla Cursos.

Esta última tabla, Cursos, cuenta con el dato de grupoID, el cual también está presente en la tabla DatosAlumno, esta será nuestra conexión entre esas dos tablas, por lo tanto, otra llave primaria es: gurpoID.

Por último, tenemos una tabla que hace referencia a los datos del profesor, llamada: DatosProfesor, y contiene los siguientes campos: nombre del profesor y

el IDProfesor, este último se asocia con la tabla cursos mediante un número que identifica al profesor, y ésta será entonces nuestra última llave primaria. Ver figura 3.26. Con ello podemos encontrar que todas las tablas están relacionadas entre sí por diversas llaves existentes en ellas.



Figura 3.26 Tabla DatosProfesor.

3.6 Conexión con la Base de Datos

Las tablas antes mencionadas están alojadas en una base de datos llamada cuentas, pero para poder acceder a ellas y guardar los datos que queremos de forma correcta, necesitamos de archivos que hagan una comunicación entre el dispositivo móvil, que es en dónde jugamos, y la base de datos que contiene nuestras tablas. Para ello se cuentan con tres archivos, todos escritos con el lenguaje PHP, y cada uno de ellos será llamado desde el juego y al mismo tiempo serán enviados los datos necesarios para guardar en las tablas correspondientes.

Primero, como se ha mencionado antes, al inicio del juego se cuenta con un panel en el que podemos registrarnos o ingresar, es aquí dónde comenzamos a recabar los datos y se puede comunicar con la base de datos. Se cuenta con un script llamado IngresoRegistro en donde existen las funciones Registrar y Entrar.

Código

```
public void Registrar(){

    message = "";
    if (CreaUser.text == "" || CreaName.text == "" || CreaPass.text == ""){
        message += "Faltan datos \n";
    }
    else{
        if (CreaPass.text == CreaRePass.text){
            WWWForm form = new WWWForm();
            form.AddField("user", CreaUser.text);
            form.AddField("name", CreaName.text);
            form.AddField("password", CreaPass.text);
            form.AddField("grupoID", CreaGpo.text);
            WWW w = new WWW("http://192.168.1.71:8080/register.php", form);
            StartCoroutine(registerFunc(w));
        }
        else
            message += "Tu contraseña no coincide \n";
    }
}

public void Entrar(){

    message = "";
    if (Usuario.text == "" || Contraseña.text == "")
        message += "Faltan datos \n";
    else{
        WWWForm form = new WWWForm();
        form.AddField("user", Usuario.text);
        form.AddField("password", Contraseña.text);
        WWW w = new WWW("http://192.168.1.71:8080/login.php", form);
        StartCoroutine(login(w));
    }
}
```

```
}  
}
```

Ambas funciones están ligadas a botones en la escena. Supongamos que es la primera vez que el usuario ingresa al juego y oprime el botón Registrar, esto activará su respectiva función y en ella encontramos un `if`, el cual verifica la validez de los campos que se piden para el registro, es decir, si de los datos que se piden hiciera falta definir alguno, la función mandará un mensaje respecto a ese problema. Si por el contrario, los datos que se piden han sido llenados completamente, pasamos al `else` de la función, en donde se vuelve a verificar que la contraseña sea la misma para ambos campos: contraseña y repetir contraseña. Si estos campos son diferentes se vuelve a mandar un mensaje de error.

Si todo va bien y las contraseñas coinciden, se define primero la forma para recabar los datos a enviar vía Internet, enseguida los datos que serán enviados, definiendo tanto el nombre del dato que será recibido por el archivo PHP, como el dato correspondiente a ese nombre, el cual es recabado desde el panel de registro del juego. Los datos que serán enviados en este mensaje serán: `usuario`, `nombre`, `password` y `grupoID`.

Ya definidos los campos que se enviarán, se declara una variable que será la encargada de mandar la forma antes generada, así como el medio por el cual se enviará. En este caso el medio será un protocolo de transferencia de hipertextos,

HTTP por sus siglas en inglés, y aunado a ello se indica la dirección IP a la cuál será enviada con su respectivo puerto de entrada y el nombre del archivo que recibirá los datos, llamado `register.php`, el otro argumento que se requiere es el que se mandará a ese archivo, es decir, la variable `form`, la cual ha sido definida previamente.

Como se puede observar, para la función Entrar se hace la misma validez de datos faltantes y se crea la misma variable `form` que encapsula los datos a enviar, solo que aquí los datos que se envían son diferentes, en este caso se supone que el usuario ya se dio de alta en la base de datos y solo debe ingresar su usuario y contraseña, estos datos son los que se mandarían en la forma correspondiente utilizando el mismo protocolo y dirección IP, pero modificando el archivo que recibirá esos datos, el nombre de ese archivo es: `login.php`.

Para ambas funciones se utilizará una *corrutina* para la comunicación con los archivos PHP, esto para asegurar una verdadera comunicación con los archivos dentro de la base: `login.php` y `register.php`, y saber que los datos se han enviado con éxito. Siendo que las llamadas a *corrutinas* se ejecutarán en su totalidad antes de retornar y permitirá tener esa seguridad mencionada.

Como se observa, cada función llama a una *corrutina* diferente, a continuación se muestra cada una de ellas.

Código

```
IEnumerator login(WWW w){

    yield return w;
    if (w.error == null)
    {
        if (w.text == "login-SUCCESS")
        {
            if (Cono.activeSelf) {
                Application.LoadLevel ("Conicas");
            } else if (Volumenes.activeSelf) {
                Application.LoadLevel ("Volumenes");
            } else if (Superficies.activeSelf) {
                Application.LoadLevel ("Superficies");
            }
        }
        else
            message += w.text;
    }
    else
    {
        message += "ERROR: " + w.error + "\n";
    }
}

IEnumerator registerFunc(WWW w){

    yield return w;
    if (w.error == null)
    {
        if (w.text == "Successfully Created User")
        {
            InicioRegistro.GetComponent<Canvas>().enabled =
                false;
            InicioEntrar.GetComponent<Canvas>().enabled =
                true;
        }else
            message += w.text;
    }
}
```

```

        }else
            message += "ERROR: " + w.error + "\n";
    }

```

En el caso de la función login, se verifica que la conexión con la base de datos sea correcta y que los campos que se han enviado son correctos también, esto evaluando el mensaje que regrese el archivo PHP correspondiente. Si no existe error con esas dos cosas, se procede a cargar la escena correspondiente al juego seleccionado por el usuario, esto se hace preguntando qué modelo es el que está activo en la escena, estos pueden ser: el modelo de volúmenes, o el de superficies, o el de secciones cónicas.

Para la otra función: registerFunc, se verificará también la conexión a la base de datos y que los campos enviados estén correctos, si no hay error, se procederá a desactivar el panel de registro y activar el panel de ingreso, en donde el usuario tendrá que ingresar los respectivos datos para cargar la nueva escena.

Ahora bien, los archivos PHP que se usan son:

Código

```

<?PHP // login.php
    $user = $_POST['user'];
    $pass = $_POST['password'];
    $con = mysql_connect("localhost","root","") or ("Cannot connect!"
        .mysql_error());
    if (!$con){
        die('Could not connect: ' . mysql_error());

```

```

}
mysql_select_db("cuentas" , $con) or die ("could not load the
        database" . mysql_error());
$check = mysql_query("SELECT * FROM DatosAlumnos WHERE
        `user`='".$user."'");
$numrows = mysql_num_rows($check);
if ($numrows == 0){
        die ("Username does not exist \n");
}
else{
        $pass = md5($pass);
        while($row = mysql_fetch_assoc($check)){
                if ($pass == $row['pass'])
                        die("login-SUCCESS");
                else
                        die("Password does not match \n");
        }
}
?>

```

El script login.php se llama cuando el usuario ya se dio de alta en la base y quiere ingresar al juego. En primer lugar se definirán las variables que recibirán los datos correspondientes, en este caso user y pass recibirán, respectivamente, los datos user y password que son los recabados y enviados por el juego desde el script IngresoRegistro. Enseguida se definirá la variable que hará la conexión a nuestra base, indicando tanto la dirección en dónde está la base, localhost, como el nombre de usuario, root, y contraseña para ingresar, para este caso, la contraseña es nula y solo se definirá el nombre de usuario.

Se evalúa si hubo conexión o no con la base, si no se logró simplemente se manda un mensaje de error, de lo contrario se ejecuta el comando de selección,

indicando el nombre de la base, cuentas en este caso, y enseguida se buscan los datos que coincidan con el nombre de usuario que se recibió. Si las columnas que arroja esa búsqueda son nulas, significará que no hay usuario registrado con ese nombre, de lo contrario, se verifica que la contraseña para ese usuario es la correcta y se enviará un mensaje de un login exitoso. Mismo que recibirá el script del juego y procederá a lo que corresponda ese mensaje.

Código

```
<?PHP // register.php
    $user = $_POST['user'];
    $name = $_POST['name'];
    $pass = $_POST['password'];
    $gpo = $_POST['grupoID'];
    $con = mysql_connect("localhost","root","") or ("Cannot connect!" .
        mysql_error());
    if (!$con){
        die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("cuentas" , $con) or die ("could not load the
        database" . mysql_error());
    $check = mysql_query("SELECT * FROM DatosAlumnos WHERE
        `user`='".$user."'");
    $numrows = mysql_num_rows($check);
    if ($numrows == 0){
        $pass = md5($pass);
        $ins = mysql_query("INSERT INTO `DatosAlumnos` ( `id`
            ,`user` , `name` , `pass` , `grupoID`) VALUES (" ,
            ".$user." , ".$name." , ".$pass." , ".$gpo." ) ;");
        if ($ins)
            die ("Succesfully Created User!");
        else
            die ("Error: " . mysql_error());
    }
    else{
```

```
        die("User already exists!");  
    }  
?>
```

Para el script que se encarga del registro de usuarios en la base de datos, se definirán y recibirán los datos de la misma manera que en el anterior, al igual que la verificación de la conexión a la base de datos y de igual modo se verifica la existencia del usuario en la misma, si el usuario ya existe en la base, entonces se enviará un mensaje con el error y el script del juego que ha llamado a éste, decidirá qué hacer en ese caso. Si por el contrario, el usuario no existe en esa base se procederá a crear al mismo con los datos recabados y para tal caso, se hace un query de inserción definiendo el nombre de la tabla, DatosAlumno, con sus respectivos campos y los valores que tomaran dichos campos.

Lo mismo pasará con los datos del profesor, los datos de los cursos, y los datos de las calificaciones que ha obtenido el usuario al terminar el juego, primero se deberán definir las variables que se insertarán en los campos, así como la base de datos a la cual se conectará, y la tabla a la cual se insertarán esos datos.

3.7 Estrategias para el uso de plantillas de juegos lúdicos

Hasta el momento se ha explicado el funcionamiento interno de las plantillas de juegos desarrollados, así como la forma en que se guardan los datos en la base

para su uso posterior. A continuación se proponen estrategias de juego para la utilización del software desarrollado.

3.7.1 Estrategia para el juego de superficies

Para el juego de reconocimiento de superficies el profesor podrá formar equipos de alumnos, y se propone que uno de ellos tenga a su cargo el dispositivo móvil que contiene la aplicación instalada, y los demás tendrán una tarjeta que asocia una ecuación con una superficie.

Al momento que el tiempo comience a correr, el alumno con el dispositivo móvil sabrá el nombre de la superficie que tiene que reconocer, éste dará a conocer a su equipo el nombre de la misma y el integrante que piensa tener la respuesta correcta deberá comunicarlo a sus compañeros, así como las razones del porqué piensa que tiene la tarjeta correcta. En tal caso, el alumno con el dispositivo móvil deberá decidir su respuesta basado en sus propios conocimientos y las razones explicadas por sus compañeros. Una vez seguro de su decisión, el alumno deberá mostrar a la cámara la tarjeta respectiva a la ecuación que haya elegido.

Si el profesor lo desea, antes de que los alumnos muestren la tarjeta a la cámara, podrá hacer pasar al frente a dos integrantes de cada equipo: uno el encargado del dispositivo móvil, y el otro, el que tenga la tarjeta con la respuesta correcta.

Esto para poder comparar las respuestas entre equipos y hacer más interactivo el ejercicio.

Para asegurar que todos los integrantes del equipo obtengan una calificación en este juego, el dispositivo móvil podrá rotarse entre ellos de tal modo que todos participen.

3.7.2 Estrategia para el juego de secciones cónicas

Para el juego de secciones cónicas se propone una estrategia similar a la anterior, en donde se formen equipos de alumnos y entre todos decidan cuál podría ser la tarjeta que contenga la respuesta correcta. En este caso, el alumno a cargo del dispositivo móvil tendrá también a su cargo la tarjeta asociada al cono que será cortado, y los demás tendrán tarjetas asociadas a planos con distinto ángulo de inclinación.

Los alumnos con las tarjetas que contienen los planos, expondrán las razones por las cuales creen que su tarjeta es la correcta, y el alumno con la tarjeta que contiene al cono decidirá, basado en su conocimiento y el de sus compañeros, con cuál plano hará colisionar el cono a su cargo.

De nuevo, si el profesor a cargo así lo decide, podrá hacer pasar al frente a dos integrantes de cada equipo, que serán los que tengan las tarjetas del cono y del

plano correcto, respectivamente. Esto para comparar las diferentes respuestas entre los equipos.

De igual forma, se recomienda rolar el dispositivo y las tarjetas entre todos los integrantes del equipo, para que todos puedan obtener una calificación en el ejercicio.

3.7.3 Estrategia para el juego de cálculo de volumen máximo

Para el juego de cálculo de volumen máximo se propone seguir la estrategia de los juegos anteriores, es decir, hacer equipos y repartir las tarjetas y el dispositivo móvil entre cada uno de los integrantes, con la variante de que aquí los integrantes del equipo podrán hacer cálculos matemáticos, tomando como datos las medidas de los cilindros asociados a sus respectivas tarjetas así como los datos correspondientes a las medidas del cono del problema.

Igual que en los juegos anteriores, el alumno a cargo del dispositivo elegirá la tarjeta que crea contiene la respuesta correcta, basado en las operaciones propias y las de sus compañeros. En este caso, como existen operaciones Matemáticas para encontrar la correcta solución, se propone que el profesor haga pasar a dos integrantes de cada equipo, los cuales repetirán los cálculos hechos para encontrar la solución, pero ahora escritos en el pizarrón, de modo a que todos puedan ver las operaciones realizadas por cada equipo y tener una realimentación

inmediata por parte del mismo profesor y de los compañeros de los demás equipos.

Si el profesor lo desea, podrá tomar en cuenta las operaciones realizadas por todos los integrantes de cada equipo para ser evaluadas como puntos extra al ejercicio, o simplemente como asistencia en clase.

Todo lo anterior son estrategias propuestas para trabajar dentro del aula con los juegos desarrollados, claro está que se pueden crear distintos tipos de actividades para el uso de estos juegos, teniendo siempre en mente la idea de interacción entre los alumnos, una inmediata realimentación y que se logre una adquisición y/o reforzamiento de conocimientos y habilidades en el alumno.

Por otro lado, fuera del aula, el alumno podrá descargar la aplicación a su dispositivo móvil, y podrá jugar las veces que quiera, esto a manera de reforzar los conocimientos y habilidades obtenidas en clase, o bien, si el profesor quisiera dejar como tarea la realización de alguno de estos juegos, podría tomarla en cuenta para evaluaciones de mismo curso.

4. Resultados

Se ha desarrollado una aplicación para el sistema operativo Android que busca apoyar las propuestas de la Facultad de Ingeniería para disminuir los problemas relacionados al rezago de alumnos. El desarrollo de esta aplicación se logró gracias a la integración de diversas herramientas de software de tal forma que se generaron tres diferentes juegos, enfocados a temas relacionados con las Matemáticas. Y aunque estos juegos pueden ser utilizados, tal y como han sido desarrollados, para trabajar en grupos o individualmente, la idea es que se usen

también como plantillas para el desarrollo de nuevos juegos que ayuden a una mejor comprensión de diversos temas matemáticos.

En todos los juegos se ha integrado la Realidad Aumentada como tecnología de desarrollo, gracias a que cuenta con gran aceptación entre las nuevas generaciones de estudiantes. Con cada juego se logra interactuar con el mundo real y el mundo digital en tiempo real, esto permite que el alumno tenga al alcance de su mano información que a veces es difícil imaginar o visualizar, es decir, con la Realidad Aumentada el alumno puede fácilmente visualizar la superficie generada por una ecuación, o las secciones de un cono al ser cortadas por un plano con cierta inclinación, y con ello facilitar la comprensión de ciertos temas.

Por otro lado, se logró la conexión con una base de datos que pretende servir de apoyo a los profesores que usen esta tecnología, de tal forma que el profesor pueda dar un seguimiento al avance de conocimientos que van teniendo sus alumnos. Con dicha base, el profesor tiene la oportunidad de tomar en cuenta los juegos desarrollados como ejercicios para evaluar, ya que se almacenan datos como: calificación, nombre del alumno, tipo de juego, etc., haciendo más amena la evaluación y de cierta manera, más organizada.

La base de datos creada para esta aplicación es una base sencilla pero con los elementos necesarios para una evaluación completa del progreso de conocimientos y habilidades que el alumno va adquiriendo durante un curso. Esta base sirve también como ejemplo para poder desarrollar una más compleja, y contar con más elementos que puedan ayudar al profesor a tener una mejor forma de evaluar a sus alumnos.

En general, se ha logrado desarrollar plenamente una aplicación de Realidad Aumentada, la cual sirve como apoyo a la comprensión de materias pertenecientes al campo de las Matemáticas. Aunque estos juegos son específicos para ciertos temas, cabe mencionar que son fácilmente modificables en cuanto a poder aumentar, o simplemente cambiar, modelos 3D, ecuaciones de superficies, medidas de objetos 3D, código para nuevas funciones, etc., en otras palabras, la aplicación es altamente escalable en muchos sentidos de tal forma que puede ser desarrollada también para diversos sistemas operativos.

5. Conclusiones

El avance tecnológico de los últimos años ha venido cambiando radicalmente el estilo de vida de la sociedad, y es que el surgimiento de las nuevas tecnologías ha logrado cosas verdaderamente sorprendentes, desde conectar al instante a personas en extremos opuestos de la Tierra, hasta enviar robots a otro planeta, y todo eso en un tiempo relativamente corto.

Es importante reconocer que la tecnología avanza a pasos agigantados, y hay que adaptarse lo mejor posible a los cambios que se generen a partir de la integración de ésta a la sociedad. Si se aprovecha de manera correcta, la tecnología puede ayudar en muchos aspectos, y la educación es algo que siempre ha requerido una constante actualización, tanto de herramientas de apoyo para la enseñanza como

de métodos de aprendizaje, y es ahí en donde puede integrarse la tecnología, como apoyo para el proceso enseñanza-aprendizaje.

El uso de estas nuevas tecnologías dentro del campo educativo, permite ampliar considerablemente el número de herramientas y métodos utilizados para la adquisición de conocimiento, y en la Facultad de Ingeniería se han logrado adaptar estas tecnologías a manera de apoyo para intentar solucionar los problemas referentes al rezago y a la deserción por parte de sus alumnos.

Si bien es cierto que la FI cuenta con diversas plataformas educativas que apoyan en gran medida el proceso de enseñanza-aprendizaje, también hay que tomar en cuenta que se deben generar recursos que trabajen a la par con dichas plataformas, y si esos recursos forman parte de las nuevas TIC, será más fácil que el alumno acuda y se apoye en ellos.

El generar este tipo de recursos educativos, como juegos lúdicos, busca ayudar a que la transferencia de información entre profesores y alumnos sea más eficiente y que el alumno se vuelva cada vez más autodidacta, o bien pueda trabajar colaborativamente en equipo dentro del aula, en general, se trata de que el alumno tenga al alcance de su mano toda la información que el profesor podría otorgarle sin necesidad de estar siempre presente y que dentro del aula pueda desarrollar otras habilidades que requiere el estudiante en su formación. Y qué

mejor recurso que un juego para potenciar la adquisición de conocimientos de una manera más atractiva para el estudiante.

Desarrollar juegos lúdicos como apoyo a métodos de aprendizaje no es una idea nueva, y gracias al surgimiento de las nuevas tecnologías, se pueden crear juegos cada vez más atractivos, de tal forma que puedan evitar que el alumno pierda interés por aprender cosas nuevas. Por otro lado, el profesor puede encontrar en estos juegos una herramienta que le ayuden a mejorar el proceso de enseñanza y al mismo tiempo apoya la integración de las nuevas tecnologías al campo educativo.

Las plantillas que se han desarrollado para este trabajo, tienen como objetivo principal ayudar a que el proceso enseñanza-aprendizaje sea más ameno y logre reforzar los conocimientos y habilidades que requiere el estudiante para su formación. Por otro lado, se busca que las plantillas sirvan como base para la generación de nuevos juegos que apoyen distintas materias.

Gracias a la escalabilidad con que cuentan las plantillas podrían mejorarse los juegos aquí planteados, desde el diseño hasta la funcionalidad, es decir, los juegos pueden mejorarse en cuanto a estilos, colores, distribuciones, o mejores modelos en 3D, y por la parte de la funcionalidad, se pueden reconocer patrones más específicos o crear juegos multijugador en línea para que los alumnos puedan competir entre sí.

En general, existen muchas mejoras que se pueden hacer a estos juegos, o bien se pueden crear nuevos juegos a partir de estas plantillas. Lo que debe tomarse en cuenta al momento de desarrollar nuevas ideas a partir de las que aquí se plantean, es que todo lo que se haga deberá estar enfocado a apoyar la transmisión de conocimientos, al refuerzo de los mismos y a mejorar la formación de los alumnos de la Facultad de Ingeniería.

6. Referencias Bibliográficas

Balderrama, R. I. (08 de 01 de 2003). *DCB-FI*. [En línea] Recuperado el 19 de 04 de 2016, de Dirección de Ciencias Básicas FI: <http://dcb.fi-c.unam.mx/Eventos/ForoMatematicas2/memorias/tres.pdf>

Cacuri, V. (2013). *Educación con TICs* (1 ed.). Buenos Aires, Argentina: Fox Andina.

CENTRO DE RECURSOS DE APRENDIZAJE PARA LAS CIENCIAS BASICAS. (04 de 2011). *CERAFIN*. [En línea] Recuperado el 23 de 06 de 2016, de CERAFIN: <http://dcb.fi-c.unam.mx/cerafin/index.php>

COPADI- FI. (2016). *COPADI*. [En línea] Recuperado el 20 de 05 de 2016, de COPADI-FI: <http://copadi.fi-c.unam.mx/bannerPrincipal/pddSAD2016.pdf>

Cubillo, M. (17 de 12 de 2012). *Recursos TIC en Educacion*. [En línea] Recuperado el 22 de 07 de 2016, de <http://recursostic.educacion.es/observatorio/web/en/software/software-educativo>

- Cubillo, M. (17 de 12 de 2012). *RecursosTIC Educación España*. [En línea] Recuperado el 29 de 06 de 2016, de <http://recursostic.educacion.es/observatorio/web/en/software/software-educativo>
- Dirección General de Tecnologías de la Información y la Comunicación. (16 de 04 de 2015). *TIC UNAM*. [En línea] Recuperado el 29 de 06 de 2016, de <http://www.tic.unam.mx/noticias/2015/04/2015-04-16-mathematica.html>
- División de Ciencias Básicas. (04 de 2011). *DCB*. [En línea] Recuperado el 23 de 06 de 2016, de DCB-FI: <http://dcb.fi-c.unam.mx/cerafin/index.php>
- División de Ciencias Básicas. (2015). *DCB-FI*. [En línea] Recuperado el 22 de 04 de 2016, de DCB-FI: http://dcb.fi-c.unam.mx/Documentos/PDD_DCB.pdf
- Edmodo. (2016). *Edmodo*. [En línea] Recuperado el 22 de 07 de 2016, de <https://www.edmodo.com/>
- Educa con TIC. (13 de 05 de 2014). *Educa con TIC*. [En línea] Recuperado el 26 de 07 de 2016, de <http://www.educacontic.es/blog/realidad-aumentada-y-educacion>
- Escobar, M. B. (27 de 04 de 2012). El perfil real del alumno de la Facultad de Ingeniería UNAM, en el momento de ingreso. México, Distrito Federal, México.
- Facultad de Ingeniería. (2015). *Facultad de Ingeniería*. [En línea] Recuperado el 23 de 05 de 2016, de FI: <http://www.ingenieria.unam.mx/paginas/estadisticas/matricula.php>
- García, F. B. (07 de 04 de 2008). *DCB-FI*. [En línea] Recuperado el 20 de 04 de 2016, de División de Ciencias Básicas Facultad de Ingeniería: <http://dcb.fi-c.unam.mx/Eventos/ForoMatematicas2/memorias2/ponencias/55.pdf>
- García, J. L. (16 de 05 de 2005). *UDLAP*. [En línea] Recuperado el 29 de 07 de 2016, de http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/lopez_g_j/capitulo1.pdf
- Gaspar, A. (02 de 11 de 2012). *Recursos TIC en Educación*. [En línea] Recuperado el 22 de 07 de 2016, de

- <http://recursostic.educacion.es/observatorio/web/en/software/software-educativo/1070-las-mejores-aplicaciones-educativas-para-android>
- Gómez, R. M. (2016). APOYOS QUE OFRECE LA FACULTAD DE INGENIERÍA PARA REDUCIR LA REPROBACIÓN, LA DESERCIÓN Y EL REZAGO., (pág. 2). México.
- González, L. d. (2008). *UAG*. [En línea] Recuperado el 20 de 02 de 2016, de Universidad Autónoma de Guadalajara:
<http://genesis.uag.mx/escholarum/vol11/ludica.html>
- González, M. P. (24 de 06 de 2013). *La plataforma Edmodo versus la plataforma Moodle*. [En línea] Recuperado el 22 de 07 de 2016, de
<http://157.88.20.45/bitstream/10324/3415/1/TFM-G%20140.pdf>
- Hernández, E. M. (02 de enero de 2012). *Revista UNAM*. [En línea] Recuperado el 03 de octubre de 2014, de <http://www.revista.unam.mx/vol.13/num1/art16/>
- INÉS, L. C. (14 de 04 de 2016). *Evaluación y seguimiento de los alumnos con ayuda de las TIC*. [En línea] Recuperado el 27 de 06 de 2016, de
http://dcb.fi-c.unam.mx/Eventos/Foro7/memorias/extensos/FIDI_02_T2_P47_3079.pdf
- MyScript. (2016). *Google Play*. [En línea] Recuperado el 11 de 02 de 2016 , de
<https://play.google.com/store/apps/details?id=com.visionobjects.calculator>
- PhotoMath. (2016). *Google Play*. [En línea] Recuperado el 02 de 11 de 2016 , de
<https://play.google.com/store/apps/details?id=com.microblink.photomath&hl=es>
- Pombo, H. L. (27 de 05 de 2014). *Universidad Complutense Madrid*. [En línea] Recuperado el 03 de 11 de 2016, de
http://eprints.ucm.es/11425/1/memoria_final_03_09_10.pdf
- Rábanos, J. M. (2015). *Comunicaciones móviles*. Madrid, España: Editorial Universitaria Ramón Areces.
- Rangel, J. L. (01 de 01 de 2013). *Revista Digital Universitaria*. [En línea] Recuperado el 22 de 09 de 2016, de Revista Digital Universitaria:
<http://www.revista.unam.mx/vol.14/num2/art16/art16.pdf>

- Rangel, J. L. (01 de 01 de 2013). *Revista UNAM*. [En línea] Recuperado el 19 de 06 de 2016, de Revista UNAM:
<http://www.revista.unam.mx/vol.14/num2/art16/art16.pdf>
- Roberto, A. P. (2016). *División de Ciencias Básicas-FI*. [En línea] Recuperado el 25 de 06 de 2016, de http://dcb.fi-c.unam.mx/Eventos/Foro7/memorias/lista.php?id=8&id_par=85
- Rodríguez, M. A. (2015). Diseño de recursos web para el apoyo docente en la FI, UNAM. *Diseño de recursos web para el apoyo docente en la FI, UNAM*. Ciudad Universitaria, Distrito Federal, México.
- RUA. (2016). *Red Universitaria de Aprendizaje*. [En línea] Recuperado el 22 de 07 de 2016, de <http://www.rua.unam.mx/>
- Sánchez, G. P. (2010). IMPACTO DE LOS RECURSOS MÓVILES EN EL APRENDIZAJE. *9ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática* (pág. 10). Orlando Florida: Editorial Digital del Tecnológico de Monterrey.
- Sarría, F. A. (13 de 02 de 2006). *Universidad de Murcia, Esp.* [En línea] Recuperado el 16 de 09 de 2016, de Universidad de Murcia, Esp.:
http://www.um.es/geograf/sigmur/temariohtml/node63_mn.html
- Telefónica, F. (2011). *Realidad Aumentada: una nueva lente para ver el mundo*. Barcelona, España: Editorial Ariel, S.A.
- UNAM. (2016). *Red Universitaria de Aprendizaje*. [En línea] Recuperado el 22 de 07 de 2016, de <http://www.rua.unam.mx/>
- Unity. (Febrero de 2014). *Unity Game Engine*. [En línea] Recuperado el 02 de octubre de 2016, de
<https://docs.unity3d.com/es/current/Manual/PrimitiveObjects.html>
- Universidad Nacional Autónoma de México. (2015). *División de Ciencias Básicas Facultad de Ingeniería*. [En línea] Recuperado el 22 de 04 de 2016, de DCB-FI: http://dcb.fi-c.unam.mx/Documentos/PDD_DCB.pdf
- UPM. (11 de 2013). [En línea] Recuperado el 19 de 06 de 2016, de Universidad Politécnica de Madrid:

http://serviciosgate.upm.es/docs/asesoramiento/guia_implementation_movil.pdf

Valdéz, J. L. (08 de Mayo de 2014). *Universidad Popular Autónoma del Estado de Puebla*. [En línea] Recuperado el 13 de Octubre de 2016, de <http://www.eumed.upap/tesis-doctorales/2014/jlcv/software.htm>