

Trabajo de Final de Grado
Grado en Ingeniería de Tecnologías Industriales

Estudio y desarrollo de una aplicación móvil de Realidad Aumentada

MEMORIA

Autor: Omar José Sedano Fernández
Director: Sergi Grau Carrión
Convocatoria: Junio 2014



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona







Resumen

El objetivo de este trabajo ha sido investigar sobre el campo de la Realidad Aumentada y desarrollar una aplicación basada en ese campo. La aplicación desarrollada ha tenido como objetivo ayudar en la docencia de profesores y mejorar el aprendizaje de conceptos a través del uso de la Realidad Aumentada en dispositivos móviles.

En la primera parte de la memoria se introducen los conceptos de Realidad Virtual y Mundo Real y se explican los distintos tipos de realidades que existen entre lo totalmente virtual y lo totalmente real, entre los cuáles se encuentra la Realidad Aumentada. Se explica con detalle en qué consiste la Realidad Aumentada, las etapas que la componen y las técnicas que se utilizan. También se presenta algún ejemplo de aplicación de Realidad Aumentada.

En la segunda parte se detallan el diseño y desarrollo de la aplicación. Se presentan las plataformas de desarrollo existentes (Android, iOS, etc.), los entornos de desarrollo (Vuforia, Metaio, etc.) y los lenguajes de programación de la aplicación. También se explican los distintos programas que se pueden utilizar para crear contenidos multimedia, ya sea imágenes, vídeos o animaciones. En esta parte es donde se detalla el diseño previo de la aplicación así como la ejecución y resultado final dividido en experiencias o animaciones.

Por último en la tercera y última parte, se habla sobre el impacto ambiental y social del proyecto, los costes que genera así como la planificación de las etapas del proyecto. Incluye además, un breve plan económico de beneficios.

En casi cada página de la memoria el lector puede encontrar referencias con notas a pie de página donde hay enlaces a vídeos relacionados con el tema que se trata, así como a documentación o páginas web. Las fuentes de cada figura o información están detalladas en forma de referencias a pie de página. Finalmente se detalla una bibliografía completa al final de la memoria.

Por último, en el soporte informático se encuentra todo el código, imágenes y animaciones de la aplicación así como una copia en formato digital de la memoria.

El resultado obtenido del proyecto es un prototipo de una aplicación de Realidad Aumentada destinada a la educación, en especial en el campo de la Física. Un prototipo capaz de mostrar de forma eficiente las posibilidades de la Realidad Aumentada y que pretende servir de base para desarrollar aplicaciones futuras en el campo.

Índice de contenido

Resumen.....	5
Glosario de abreviaturas y términos.....	9
1. Introducción.....	10
1.1. Objetivos.....	11
1.2. Alcance	11
1.3. Requisitos.....	12
2. Estudio del Arte.....	14
2.1. Realidad Virtual vs Realidad Aumentada vs Realidad Mixta.....	14
2.2. La Realidad Aumentada.....	17
2.2.1. Etapas de la Realidad Aumentada.....	19
2.2.2. Reconocimiento de objetos y/o imágenes.....	20
2.2.3. Tracking.....	22
2.2.4. Iluminación y renderizado.....	24
2.2.5. Aplicaciones de Realidad Aumentada	27
3. Diseño y desarrollo.....	31
3.1. Introducción y funcionalidades.....	31
3.1.1. Plataformas de desarrollo.....	31
3.1.2. Entornos de desarrollo	35
3.1.3. Lenguajes de programación	40
3.2. Creación de contenidos digitales para la aplicación	41
3.2.1. Imagen	42
3.2.2. Gráficos 3D/Animación.....	44
3.2.3. Video.....	46
3.3. Diseño de la aplicación	48
3.3.1. Diseño Experiencia 1: Ciclotrón.....	48
3.3.2. Diseño Experiencia 2: Rampa con rozamiento.....	51
3.3.3. Diseño Experiencia 3: Balance de energías.....	53
3.3.4. Diseño Experiencia 4: Sistema Solar	56
3.4. Desarrollo de la aplicación.....	58
3.4.1. Experiencia 1: Ciclotrón	58
3.4.2. Experiencia 2: Rampa con rozamiento	63



3.4.3. Experiencia 3: Balance de energías	66
3.4.4. Experiencia 4: Sistema Solar	71
4. Impacto ambiental y social	75
4.1. Impacto ambiental del desarrollo de la aplicación	75
4.2. Impacto ambiental de la aplicación.....	76
4.3. Impacto social de la aplicación	76
5. Planificación y valoración económica	77
5.1. Planificación	77
5.2. Valoración económica	79
5.2.1. Costes de recursos humanos.....	79
5.2.2. Costes de material y software.....	80
5.2.3. Otros costes.....	80
5.2.4. Coste total	81
6. Plan económico del proyecto	82
7. Trabajo futuro	83
8. Conclusiones	84
9. Agradecimientos	85
10. Bibliografía.....	86
10.1. Bibliografía referenciada.....	86
10.2. Bibliografía completa.....	87



Glosario de abreviaturas y términos

RA	Realidad Aumentada.
App	Aplicación móvil.
Smartphone	Teléfono inteligente.
Sistema operativo	Conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los otros tipos de programas.
Android	Sistema operativo de dispositivos móviles.
iOS	Sistema operativo de dispositivos móviles.
Windows Phone	Sistema operativo de dispositivos móviles.
Blackberry	Sistema operativo de dispositivos móviles.
Virtual	Propiedad de aquello que tiene existencia aparente y no real.
AR	Augmented Reality (Realidad Aumentada)
VR	Virtual Reality (Realidad Virtual)
Segmentación	Proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos.
Registro	Proceso de transformación de diferentes conjuntos de datos a un sistema de coordenadas.
Renderizado	Proceso de generar una imagen o vídeo mediante el cálculo de iluminación partiendo de un modelo en 3D.
API	Application Programming Interface (Interfaz de Programación de Aplicaciones).
Python	Lenguaje de programación.
Java	Leguaje de programación.
C	Leguaje de programación.
C++	Leguaje de programación.

1. Introducción

Este trabajo está enfocado en la investigación y desarrollo de una aplicación de Realidad Aumentada (RA). La RA es el concepto de mezclar imágenes reales con imágenes virtuales de forma que introduce elementos nuevos e inexistentes en la realidad del usuario.

Quizás el ejemplo más ilustrativo de la utilización de la RA se encuentra en el mundo del cine: en películas con elementos fantásticos como las de superhéroes o películas de acción muchas veces lo que se hace es grabar a los actores y actrices delante de una pared forrada de una tela verde o azul. Cuando la escena es editada más tarde se introducen por ordenador el lugar en el que se desarrolla la acción, monstruos, personajes fantásticos, dragones así como efectos especiales que parecen reales. Una de las técnicas utilizadas para componer estas escenas se denomina rotoscopia y consiste en dibujar sobre cada fotograma real los elementos que se quieren introducir. Otra técnica muy conocida es el croma, que consiste en extraer un color de la imagen grabada y reemplazar el área que ocupaba ese color por otra imagen, con ayuda de ordenadores (Figura 1).

Lo que se hace pues en estos casos es coger una escena con elementos reales, en este caso los actores y rodearlos de elementos virtuales o no reales de forma que se “aumenta” la realidad que vemos. Esta técnica se puede considerar Realidad Aumentada.

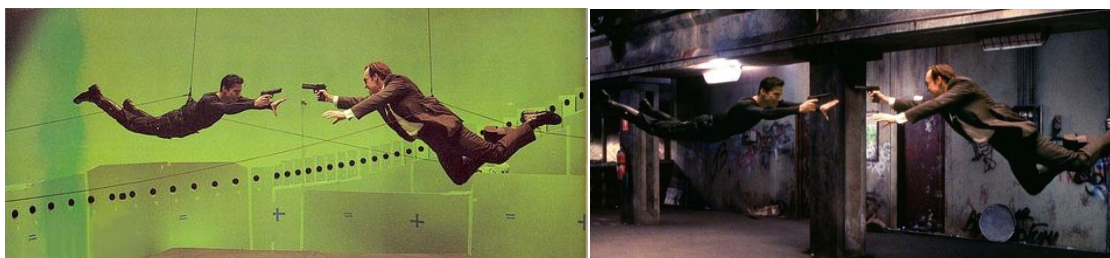


Figura 1: Escena de la película The Matrix. Escena original (real) rodada en el estudio vs Escena final, en la que se han introducido elementos virtuales, como el fondo.¹

Lo cierto es que parece una tecnología muy potente y lo es, el ejemplo que se acaba de presentar no es más que la punta del iceberg. Se puede ir un paso más allá, de hecho se puede hacer AR en tiempo real sin tener que esperar a introducir esos elementos virtuales a través de un ordenador y después mostrarlos en una sala de cine. Y eso es sobre en lo que se centra este proyecto, la AR en tiempo real a través de dispositivos móviles.

Sólo hace falta ver un par de vídeos de demostración de AR para darse cuenta de la variedad de aplicaciones y funcionalidades que se pueden desarrollar: una aplicación que mostrase a un cirujano visión en 3D del cuerpo del paciente², pasar de ver planos de edificios sobre un papel ver esos edificios surgir delante nuestro cómo si realmente existieran³, ver cómo queda un mueble en tu casa⁴, probarte la ropa sin tener que ir a la tienda⁵, etc.

¹ Fuente: <http://dungeondomeme.blogspot.com.es/2012/06/como-foram-feitas-10-cenas.html>

² Link: MITK pille, <https://www.youtube.com/watch?v=T4mboj-GbEA>

³ Link: Augmented Architectural Demonstration, https://www.youtube.com/watch?v=w0Uk-u_hZ3M

⁴ Link: IKEA Catalogue, <https://www.youtube.com/watch?v=vDNzTasuYEw>



Entre las muchas direcciones temáticas en las cuáles podría haber ido el proyecto escogí la educación. Ver un modelo en 3D de un motor de combustión delante nuestro, observar cómo un mecanismo se mueve, ver complicadas figuras matemáticas pueden complementar horas de explicaciones del profesor que en muchas ocasiones no acaban de quedar claras para el alumno. Además, el medio visual es por el que se recuerda la mayor cantidad de información que recibe un ser humano, por esa razón mostrar un concepto a través de una animación o un objeto puede resultar muy provechoso.

La motivación de este proyecto se centra en las aplicaciones que faciliten la labor docente y al mismo tiempo aumenten la motivación y el interés del alumno en la materia, algo que personalmente considero la piedra angular de cualquier lección. Despertar el interés debe ser la primera y más importante función de un docente, a mi parecer, y creo que aplicaciones desarrolladas en tecnologías tan punteras e innovadoras como la Realidad Aumentada puede ayudar a captar la atención del espectador.

1.1. Objetivos

El objetivo general del proyecto es desarrollar una aplicación para móvil o tablets, enfocada a mejorar el aprendizaje de conceptos teóricos y prácticos en el marco de la educación secundaria, en asignaturas como la Física o las Matemáticas. También se presenta como objetivo inherente al proyecto investigar en el campo de de la realidad aumentada.

A continuación se presentan los objetivos específicos que se pretenden alcanzar:

- Definir y entender la tecnología de Realidad Aumentada.
- Investigar sobre las diferentes aplicaciones de la Realidad Aumentada.
- Conocer las diferentes plataformas para desarrollar software de Realidad Aumentada.
- Desarrollar una aplicación de Realidad Aumentada para dispositivos portátiles, dirigida al mundo educativo.
- Observar y estudiar otras aplicaciones didácticas de Realidad Aumentada.

1.2. Alcance

La motivación del proyecto es conseguir facilitar el aprendizaje a las personas o alumnos que utilicen la aplicación a desarrollar. Esto corresponde a los problemas que por experiencia he tenido a lo largo de de mi carrera universitaria así como en la escuela secundaria y primaria. Problemas relacionados con la comprensión de conceptos en asignaturas como la Física o la Mecánica donde el profesor dibuja un esquema estático en 2 dimensiones y un problema que en su mayoría de veces es dinámico y tridimensional. Gran parte de la nota de una asignatura se debe al hecho de que no se explican y/o entienden bien los conceptos. Mi intención es

⁵ Link: Augmented Reality & Motion Capture Shopping App, <https://www.youtube.com/watch?v=NxQZuo6pFUw>

poder ayudar a que esos conceptos se enseñen de forma clara y amena mediante una experiencia interactiva y replicable, de tal forma que se pueda revivir la lección en cualquier lugar y en cualquier instante. Me ha pasado muchas veces que después de comprender un ejercicio en clase de Mecánica sobre el funcionamiento de una máquina, al repasarlo días más tarde no sabía cómo se comportaba el mecanismo del ejercicio.

Otro concepto que me gustaría destacar es la interacción sin la cual este proyecto no tiene sentido. En mi opinión todo se resume a esta frase atribuida al filósofo chino Confucio: *"Me lo contaron y lo olvidé; lo vi y lo entendí; lo hice y lo aprendí"*. Se trata de que el alumno pueda "hacer" y para ello debe ser capaz de interactuar con el objeto que muestra la aplicación. Volviendo al ejemplo del motor de combustión, si alguien que nos explica las partes del motor sin ninguna foto de ayuda será difícil que se recuerden dos o tres piezas como máximo. Si a continuación realiza la lección con ayuda de un video o imagen donde se muestre el motor y sus partes (Figura 2), la mayoría de nosotros olvidaremos el nombre de las piezas pero no la estructura y la forma de las piezas del motor. Si queremos ir un paso más allá cada uno debería desmontar y montar un motor para conocer todas sus partes, *"lo hice y lo aprendí"*, y es ahí donde con una aplicación interactiva se puede introducir virtualmente ese motor para que la gente pueda interactuar con él: ser capaz de montarlo como en la realidad por ejemplo.

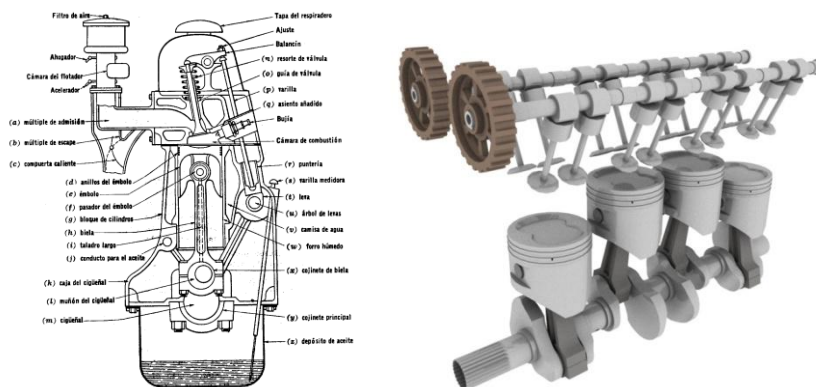


Figura 2: Motor de combustión. El modelo en 3D es más ilustrativo que el esquema en 2D. ⁶

1.3. Requisitos

A la hora de explicar los requisitos de este proyecto se hará una división de dos tipos de requisitos:

- Requisitos técnicos: se refieren a la vertiente tecnológica de la aplicación. Se pueden dividir en dos clases:
 - De desarrollo: se refieren a qué características software y/o hardware se necesitan para el desarrollo de la aplicación.

⁶ Fuente: http://upload.wikimedia.org/wikipedia/commons/6/69/Engine_movingparts.jpg, <http://html.rincondelvago.com/000251549.png>



- De ejecución: se especifican las características de los dispositivos móviles y el software necesarios para que el usuario ejecute la aplicación.
- Requisitos de la aplicación: qué características generales debe tener la aplicación para garantizar un buen producto final.

Requisitos técnicos

- a) De desarrollo: Ordenador que permita desarrollar aplicaciones para la plataforma escogida. Es necesario estudiar previamente las diferentes plataformas de desarrollo de Realidad Aumentada. Otro elemento importante es la disponibilidad de un dispositivo móvil como puede ser un smart-phone o una tablet (Figura 3), que dispongan de cámara de video para que se pueda testear la aplicación. Cabe destacar que también se puede probar perfectamente la aplicación con un ordenador con cámara integrada o externa.



Figura 3: Dispositivos móviles: tablets o teléfonos son la plataforma de uso de la aplicación.⁷

- b) De ejecución: se necesita un móvil o tablet con cámara de video que son los dispositivos a los que va dirigida la aplicación. Otro aspecto importante es el sistema operativo del dispositivo (Android, iOS, Windows Phone, etc.) que debe ser compatible con la aplicación, la cual debe aspirar a integrarse en todos los sistemas operativos.



Figura 4: Sistemas operativos de dispositivos móviles. De izquierda a derecha, Android, Windows Phone, iOS, BlackBerry.⁸

Requisitos de la aplicación

- a) Intuitiva. Esencial para el éxito del producto.
- b) Atractiva. El público al que va dirigido es un público joven (estudiantes de secundaria y/o universidad) por lo que debe atraer su atención con un buen diseño.
- c) Precisa. Los contenidos mostrados deben presentar gran rigor en los contenidos mostrados.
- d) Rápida. La aplicación debe proporcionar una interacción fluida sin grandes tiempos de espera.
- e) Interactiva. El usuario debe poder interactuar con la aplicación y no limitarse a observar el contenido. Debe crearse contenido que sólo se pueda mostrar y entender si el usuario participa.

⁷ Fuente: <http://etecnologia.com/movil/asus-padfone>

⁸ Fuente: <http://www.poderpda.com/investigacion-y-desarrollo/windows-phone-crece-a-expensas-de-symbian-y-blackberry/attachment/ios-android-windows-phone-blackberry-2/>

- f) Contenido multimedia. Modelos en 3D, animaciones, videos de corta duración y otros contenidos audiovisuales que hagan dinámica la experiencia.

2. Estudio del Arte

2.1. Realidad Virtual vs Realidad Aumentada vs Realidad Mixta

Para comprender perfectamente el concepto de la Realidad Aumentada se debe antes conocer qué tipo de “realidades” existen. Todas estas realidades tienen en común la presencia de objetos reales y/o elementos virtuales, y es la forma en la que se relacionan estos dos tipos de elementos y la predominancia de uno sobre el otro lo que diferencia las distintas realidades. Una de las clasificaciones hechas es la realizada por Paul Milgram y Fumio Kishino, que en el año 1994 definieron el concepto de *Continuo de la Virtualidad*⁹. El Continuo de la Virtualidad define que hay una escala continua de realidades (Figura 5) en cuyos extremos se sitúa por una parte la denominada Realidad Virtual, caracterizada por la presencia de un mundo completamente virtual, y por otra parte el llamado Mundo Real donde todo es completamente real.



Figura 5: Esquema del Continuo de la virtualidad, basado en Milgram and Kishino (1994).¹⁰

A medida que se va hacia la derecha la cantidad de elementos generados por ordenador va aumentando. A la izquierda de todo se sitúa el Mundo Real que es la realidad que el ser humano percibe en su vida cotidiana, en cambio a la derecha del todo se encuentra la Realidad Virtual donde todos los elementos son generados virtualmente y físicamente no existen. El espacio comprendido entre los dos extremos corresponde a la denominada Realidad Mixta, que mezcla tanto elementos reales como virtuales. Esta clasificación de tres es más amplia, de hecho se pueden distinguir diferentes tipos dentro de la Realidad Mixta y la Realidad Virtual.

⁹ Fuente: Milgram, Paul; H. Takemura, A. Utsumi, F. Kishino (1994). "Augmented Reality: A class of displays on the reality-virtuality continuum" (pdf). *Proceedings of Telematic and Telepresence Technologies*. pp. 2351–34. Retrieved 2007-03-15.

¹⁰ Fuente: http://en.wikipedia.org/wiki/File:Adapted_milgrams_VR-AR_continuum.png



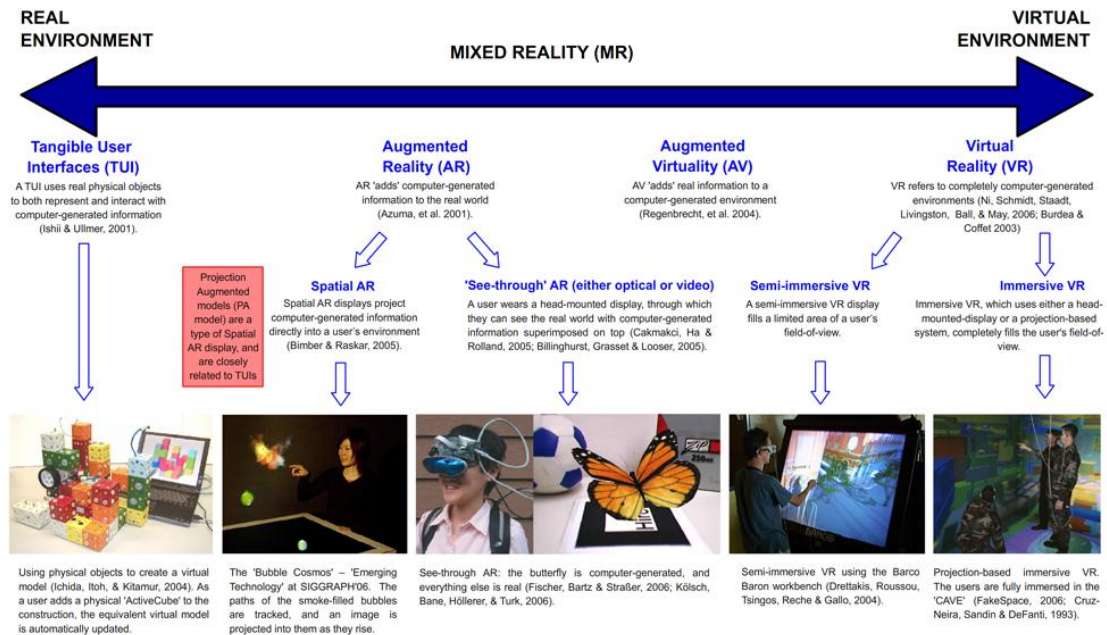


Figura 6: Continuo de la virtualidad, basado en Milgram and Kishino (1994).¹¹

Empezando en el mundo real y caminando hacia el mundo virtual lo primero que se puede destacar es el TUI:

TUI

Las *Tangible User Interfaces* (TUI) o Interfaces Tangibles de Usuario utilizan objetos reales que permiten interactuar con información generada por ordenador. Un ejemplo es la tabla de música electrónica Reactable¹², que genera diferentes sonidos y ritmos según cómo se coloquen o se muevan por su superficie unos cubos que el usuario maneja (Figura 7).



Figura 7: Tabla de música Reactable. Múltiples usuarios simultáneos comparten el control total del instrumento moviendo y rotando objetos físicos sobre la superficie de una mesa circular luminosa.¹³

Un paso más virtual que las TUI se encuentra la Realidad Aumentada.

¹¹ Fuente: http://en.wikipedia.org/wiki/File:Adapted_milgrams_VR-AR_continuum.png

¹² Link: Reactable, <https://www.youtube.com/watch?v=Mgy1S8qymx0>

¹³ Fuente: http://es.wikipedia.org/wiki/Reactable#mediaviewer/Archivo:Reactable_Multitouch.jpg

Realidad Aumentada

La Realidad Aumentada se sitúa en un tipo de Realidad Mixta más cercana al Mundo Real que a la Realidad Virtual y consiste en añadir elementos virtuales generados por ordenador al mundo real de forma que se enriquece esa realidad. Dentro de la propia Realidad Aumentada se pueden encontrar dos tipos:

- Spatial AR: se basa en introducir elementos virtuales a través de proyecciones visuales sobre el entorno del usuario.¹⁴
- See-through AR: se basa en introducir elementos virtuales mediante un dispositivo como unas gafas, una tablet o un móvil a través de los cuáles mira el usuario (Figura 8).¹⁵



Figura 8: See-through AR. La mariposa es virtual mientras que todo lo demás es real. A la izquierda se muestra la imagen que el usuario ve a través del casco.¹⁶

Cercana a la Realidad Aumentada se encuentra la Virtualidad Aumentada.

Virtualidad Aumentada

Si la Realidad Aumentada consiste en introducir algo virtual en un mundo real, la Realidad Virtual¹⁷ consiste en introducir algo real en un mundo virtual. Por lo tanto está más cerca de la Realidad Virtual que del Mundo Real (Figura 9).



Figura 9: Virtualidad Aumentada. El coche, la calle y en general el entorno es en su mayoría virtual y como elemento real se encuentra la imagen de una persona.¹⁸

¹⁴ Link: Spatial AR, <https://www.youtube.com/watch?v=iNT3eXwTfRY>

¹⁵ Link: See-through AR, <https://www.youtube.com/watch?v=zXWSpTnKTW>

¹⁶ Fuente: http://upload.wikimedia.org/wikipedia/en/d/d2/Adapted_milgrams_VR-AR_continuum.png

¹⁷ Link: Kinect Augmented Virtuality, https://www.youtube.com/watch?v=mpTKgs_cMCU



Realidad Virtual

Se refiere a un entorno generado por completo por ordenador, es decir, todo es virtual. Según el grado de inmersión visual del usuario se distinguen:

- *Semi-immersive VR*: el mundo virtual sólo llena una parte del campo de visión del usuario (Figura 10).



Figura 10: Ejemplo de Semi-immersiveVR¹⁹

- *Immersive VR*: el mundo virtual ocupa todo el campo de visión del usuario, ya sea empleando un casco de visión o proyecciones (Figura 11).²⁰



Figura 11: Ejemplo de ImmersiveVR²¹

2.2. La Realidad Aumentada

En este apartado se va repasar y profundizar un poco más en el concepto de Realidad Aumentada.

La realidad aumentada se puede definir como el tipo de realidad que una persona observa cuando se combina un entorno físico real con elementos virtuales. Esa realidad o visión tiene lugar a través de un dispositivo tecnológico que mezcla la realidad física del usuario con

¹⁸ Fuente: https://www.youtube.com/watch?v=mpTKgs_cMCU

¹⁹ Fuente: http://en.wikipedia.org/wiki/File:Adapted_milgrams_VR-AR_continuum.png

²⁰ Link: Immersive Virtual Reality, <https://www.youtube.com/watch?v=UT5xtpHRdIE>

²¹ Fuente: http://en.wikipedia.org/wiki/File:Adapted_milgrams_VR-AR_continuum.png

elementos físicamente inexistentes (generados por ordenador) y muestra el resultado al usuario. Ejemplos de estos dispositivos serían móviles, tablets, gafas, y en general cualquier dispositivo que disponga de una pantalla o superficie para mostrar la nueva realidad creada al usuario y que disponga de un sensor de captación de imagen o cámara para poder recoger la realidad física existente.

Un ejemplo de lo que es la Realidad Aumentada se muestra en la Figura 12.



Figura 12: Concepto visión en RA a través de gafas o lentillas. Escena ficticia, parte de una recreación.²²

En ella se muestra la visión o realidad que vería un usuario a través de unas lentillas o gafas que implementasen la tecnología de realidad aumentada. Se pueden observar 2 tipos de elementos:

- Elementos reales: la chica, las mesas, el suelo, el local, son elementos físicamente existentes.
- Elementos virtuales o no reales: el corazón situado en la parte superior izquierda, el texto que aparece a la derecha de la imagen y detrás de la chica las dos imágenes, son todos elementos virtuales introducidos externamente.

Estos dos tipos de elementos (reales y virtuales) se unen para enriquecer la realidad que capta el usuario que es la que muestra la imagen de ejemplo.

Por lo tanto en realidad aumentada siempre debe haber estos dos tipos de elementos: elementos reales y elementos virtuales, con predominancia de los elementos reales sobre los virtuales.



Figura 13: Esquema de elementos.

Además de la combinación de elementos reales y virtuales la Realidad Aumentada puede caracterizarse por poseer otras propiedades que pueden variar según cada autor. Por ejemplo, Ronald Azuma²³, considerado uno de los pioneros e investigadores de mayor

²² Fuente: Sight by Eran May-raz and Daniel Lazo. <http://vimeo.com/46304267>

²³ Ronald Azuma, <http://www.ronaldazuma.com/>



reconocimiento en Realidad Aumentada reconoce tres características fundamentales que la definen:

- Combina elementos reales y virtuales. Característica esencial.
- Es interactiva en tiempo real. Se refiere a que se interactúa y se generan los contenidos en el mismo instante en que se visualiza el mundo real. Según esta definición la introducción de elementos por ordenador en el cine no sería considerada RA.
- Está registrada en 3D. Si bien es verdad que una de las ventajas de la Realidad Aumentada es poder visualizar objetos tridimensionales a partir de esquemas o dibujos planos, esta característica excluye a animaciones en 2D o la visualización de videos.



Figura 14: App de la marca MINI. Muestra el modelo de un coche en 3D y en tiempo real. ²⁴

2.2.1. Etapas de la Realidad Aumentada

La Realidad Aumentada se construye a partir de una serie de de 3 acciones o etapas esenciales que se suceden una tras otra en el tiempo:

- Reconocimiento de objetos.
- Tracking o seguimiento de objetos.
- Iluminación y Renderizado o representación de contenidos.

A su vez estas 3 fases se pueden clasificar dentro de dos grandes grupos o campos:

- Visión por Computador.
 - Reconocimiento de objetos.
 - Tracking.
- Gráficos por Computador.
 - Iluminación y Renderizado

²⁴ Fuente: <http://cdn.thefwa.com/library/Bildschirmfoto-2013-01-29-TE7J.png>

Por una parte la Visión por Computador engloba el reconocimiento de imágenes y/o objetos así como el tracking. Es conocida también como Visión Artificial o Visión técnica²⁵ y consiste en programar un ordenador para que reconozca y “entienda” las características de una escena o imagen. Algunos de sus principales objetivos son:

- La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes (por ejemplo, caras humanas).
- La evaluación de los resultados (por ejemplo, segmentación, registro).
- Mapeo de escenas reales.

Por otra parte el campo de Gráficos por Computador engloba la iluminación y el renderizado. Conocido también por el nombre Computación Gráfica²⁶, el campo de los Gráficos por Computador consiste en utilizar ordenadores tanto para generar imágenes o modelos visuales como para integrarlos en el mundo real de forma eficaz.

A continuación se describen cada una de las 3 fases de la Realidad Aumentada.

2.2.2. Reconocimiento de objetos y/o imágenes



Figura 15: App de Realidad Aumentada que crea contenido entorno a la fachada del edificio.²⁵

Para conseguir representar correctamente los elementos virtuales en la escena del mundo real es necesario reconocer el entorno que rodea al usuario y los elementos que componen ese mismo entorno. Por ejemplo, una aplicación que permita crear contenido virtual sobre la fachada de un edificio debe reconocer primero el perfil del edificio para saber dónde debe crear el contenido multimedia (Figura 15).

Otra situación en la que es necesario el reconocimiento de imagen es en la detección de los llamados targets (Figura 16). Los targets son imágenes e incluso figuras que permiten posicionar los contenidos virtuales en la escena, es decir, los elementos visuales se crean entorno al target.²⁶



Figura 16: Ejemplo de target.²⁷

²⁵ Fuente: <http://ideabuilderhomes.com/wp-content/uploads/2011/07/Augmented-reality-001.jpg>

²⁶ Link: Ejemplo de Target, <https://www.youtube.com/watch?v=42MvJ1QgPz8>

²⁷ Fuente: http://www.knowledgevault.com/vault451/images/Horse_Augmented_Reality_example.jpg



Así pues se define el reconocimiento de objetos^[27] como la tarea de encontrar e identificar objetos en imágenes o secuencias de video.

Tanto si hace falta reconocer un elemento físico como un edificio, como si hace falta reconocer una imagen que haga de target, esta tarea es aparentemente sencilla para un ser humano sin embargo aún no se ha resuelto de forma satisfactoria en el campo de la Visión Computarizada. No al menos en el caso más general y a la vez complejo que se puede presentar: objetos arbitrarios en una situación o entorno arbitrario. Aún así se han logrado grandes avances en este campo en las últimas décadas.

Dentro del reconocimiento de objetos u *object recognition* se pueden distinguir diversos métodos o técnicas, a continuación se presentan algunas de ellas.

Appearance-based methods o Métodos basados en la apariencia

Usan imágenes del objeto a reconocer (llamadas templates) para ejecutar el reconocimiento por comparación. Para ello hay que tener en cuenta que cada objeto se muestra diferente según las condiciones:

- Cambios en la luz y el color.
- Cambios en la perspectiva desde la cual se mira.
- Cambios en el tamaño y la forma.

De esta forma habría que disponer de infinitas imágenes para poder identificar siempre el objeto con el 100% de eficacia, cosa que resulta imposible. Algunos métodos basados en este concepto son:

- Edge matching: se basa en marcar los vértices en la imagen de ejemplo del objeto o template y marcar los vértices en la imagen a analizar. Se comparan entonces los vértices de la imagen a analizar con los vértices de la imagen de ejemplo y en función de la coincidencia en la posición de los vértices, los píxeles cercanos a los vértices o las distancias entre vértices se determina la coincidencia o no. Para encontrar los vértices se usan las llamadas técnicas de de detección de vértices (Figura 17) o *edge detection*.²⁸



Figura 17: Imagen que muestra la técnica de edge detection o detección de vértices ²⁹

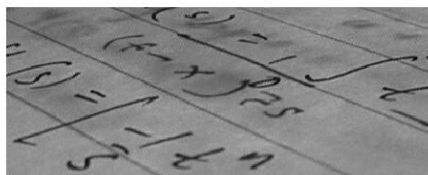
²⁸ Link: OpenCV edege detection, <https://www.youtube.com/watch?v=nxANbjz00Q>

²⁹ Fuente: <http://upload.wikimedia.org/wikipedia/en/8/8e/EdgeDetectionMathematica.png>

- *Divide-and-Conquer search*: es una técnica recursiva basada en dividir la imagen en pequeñas celdas u objetos de forma que el problema de reconocimiento se hace más sencillo de tratar. El procedimiento se basa en primero dividir la imagen en celdas que contengan información relevante. Luego, se determina la celda de menor diferencia respecto a la imagen de referencia. Si esta celda es demasiado grande se recorta, y si es aceptable se divide en celdas más pequeñas y se vuelve a realizar el proceso.
- *Greyscale matching*: coincidencia de color entre imágenes en la escala de grises. Esta técnica surge debido a que la detección por vértices, aunque es robusta a los cambios de iluminación, deshecha mucha información de la imagen. Calcula la distancia de píxeles como una función tanto de la posición de los píxeles como de la intensidad de los mismos.

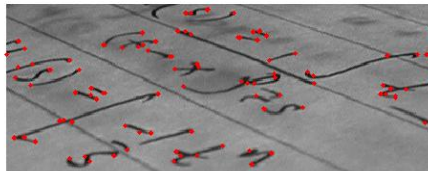
Feature-based methods o Métodos basados los “features”

Se basan en el análisis de los *features* que se definen como las partes “interesantes” de una imagen o que pueden servir para identificarla. Ejemplos de *features* son los vértices, las esquinas o los llamados blobs³⁰, que son regiones de una imagen donde se mantienen constantes ciertas propiedades o al menos varían muy poco (Figura 18).



30

Figura 18: Ejemplo de corner detection o detección por esquinas.



2.2.3. Tracking

El tracking³⁰ es el proceso de localizar un objeto que se está moviendo en el espacio y en el tiempo. Esta fase está íntimamente relacionada con el reconocimiento de objetos, ya que para posicionar el objeto en cualquier instante éste debe ser reconocido. Una parte importante del proceso de tracking además de las técnicas de reconocimiento previamente descritas, es la conversión de las coordenadas de la imagen a coordenadas globales pertenecientes al entorno real que el usuario ve. Esta relación entre coordenadas puede ser facilitada por el target o un marcador (entre otros) cuya orientación y posición es conocida (Figura 19).

³⁰ Fuente: http://en.wikipedia.org/wiki/Object_recognition#mediaviewer/File:Corner.png



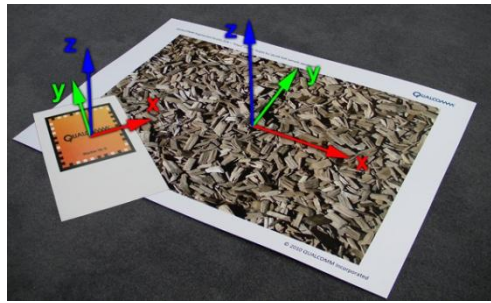


Figura 19: Targets/Marcador con un sistema de referencia de coordenadas ³¹

Para que el tracking sea posible un algoritmo analiza la secuencia de imágenes que la cámara del dispositivo (móvil, tablet, etc.) capta e interpreta y extrae el movimiento del objeto entre los frames. Algunos de los algoritmos utilizados para el tracking de objetos captados por video son:

- **Blob tracking:** segmenta el interior de los objetos.
- **Kernel-base tracking:** relacionado con el algoritmo “mean-shift”. Este algoritmo crea una función de densidad de probabilidad en la imagen que capta, asignando a cada pixel de la nueva imagen una cierta probabilidad. Probabilidad de que el color del píxel de la nueva imagen exista en la imagen del objeto de referencia. Entonces busca el pico de probabilidad/coincidencia cerca de la posición inicial del objeto.
- **Contour tracking:** detección del contorno o límites de los objetos.

Otro aspecto a destacar es que dentro del tracking se pueden distinguir diferentes tipos, según cuál sea el objeto o elemento del cual se esté estudiando el movimiento. Son ejemplos el face tracking o el hand tracking.

Face tracking

Consiste en determinar la localización en cada instante de caras humanas y detectar sus gestos (Figura 20). Mediante algoritmos se es capaz de seguir y reconocer los gestos de la cara del usuario dibujando una malla sobre ella. ³²

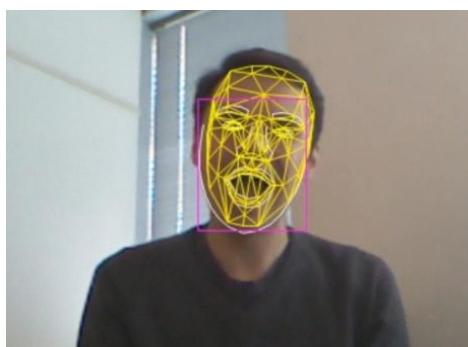


Figura 20: Face tracking: se malla la cara a fin de reconocer sus gestos ³³

³¹Fuente:

https://developer.vuforia.com/sites/default/files/dev_guide_images/create_targets_devguide/trackable_create_targets.jpg

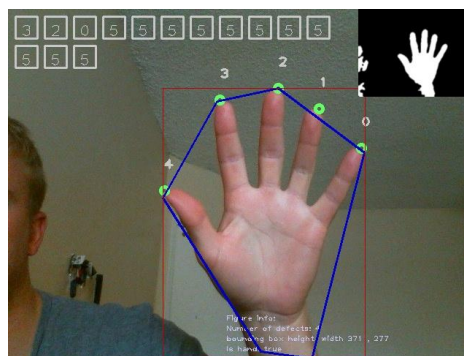
³² Link: Face Tracking, <https://www.youtube.com/watch?v=acKTCF0sAfc>

Head tracking

Se basa en determinar la posición y movimientos de la cabeza del usuario. El interés de esta técnica permite crear aplicaciones donde según se mueva la cabeza, la imagen mostrada en una pantalla cambia su perspectiva dando la sensación de profundidad, como si fuera tridimensional.³⁴ Un ejemplo de ello es el mostrado por Johnny Chung Lee, que con ayuda de los mandos de la videoconsola Wii y su sensor de movimiento consigue crear una experiencia de realidad aumentada con ayuda del head tracking.³⁵

Hand tracking

Determina la posición y movimientos de la mano así como de cada uno de sus dedos (Figura 21).³⁶ Un ejemplo de aplicación de esta tecnología es el desarrollo de un programa que permita introducir los números de un password o un pin mediante los dedos.³⁷



38

Figura 21: Ejemplo de Hand tracking

Optical Flow

Traducido al español como *flujo óptico* es definido como el patrón de movimiento aparente de un objeto o elemento desde la referencia del usuario, cuando hay movimiento relativo entre escena/entorno y observador/usuario. Algunas de sus aplicaciones consisten en la detección de movimiento o la codificación de movimientos.^{39 40}

2.2.4. Iluminación y renderizado

Una vez los objetos de la escena son identificados y analizados, es necesario introducir los elementos virtuales creados mediante programas de representación gráfica. Sin embargo no es suficiente con generar los gráficos sino que se deben intentar adaptar a las condiciones del

³³ Fuente: <http://www.codeproject.com/KB/library/394975/FaceMask.jpg>

³⁴ Link: Head Tracking, http://www.youtube.com/watch?v=h9kPI7_vhAU

³⁵ Link: Head Tracking con la Wii, <https://www.youtube.com/watch?v=Jd3-eiid-Uw>

³⁶ Link: Hand Tracking con Kinect3D, <https://www.youtube.com/watch?v=iv-azkZlpl#t=25>

³⁷ Link: Pin mediante Hand Tracking <https://www.youtube.com/watch?v=CYJjQgV0mZY>

³⁸ Fuente: http://simena86.github.io/images/handRecognition/final_result.png

³⁹ Link: Optical Flow, <https://www.youtube.com/watch?v=JlLkom6tWw>

⁴⁰ Link: Optical Flow : https://www.youtube.com/watch?v=5rR_9Ylcg_s



entorno, a fin de que la experiencia de Realidad Aumentada sea realista (Figura 22). Esto se consigue mediante dos vías:

- Coherencia geométrica o de posición.
- Coherencia en la iluminación.

Si se consigue cumplir las dos condiciones se obtiene una Realidad Aumentada satisfactoria.

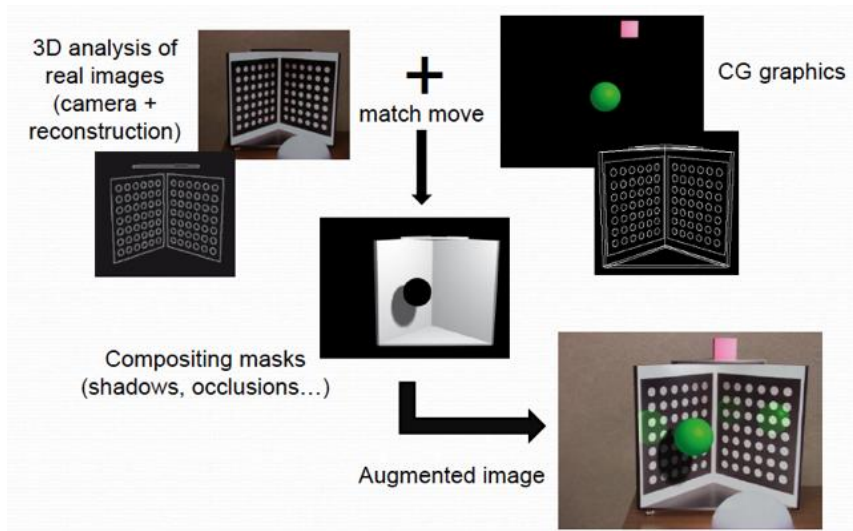


Figura 22: Esquema de construcción de RA realista.

Coherencia geométrica

Se refiere a que una vez se ha detectado el entorno real, el objeto virtual se ha de introducir en una posición determinada para que se mimetice con la escena de forma que parezca que realmente pertenece a ella. Por ejemplo, en las imágenes que se muestran en la Figura 23 se observa como un usuario se pone un chaleco lleno de marcadores dónde aparece posteriormente un traje de buzo. Sin embargo se observa que esto no es suficiente para dar la sensación de realidad ya que en la imagen superior derecha y la imagen inferior izquierda el traje aparece enfrente del usuario, fenómeno poco realista. En la imagen inferior derecha pero, sí parece que el usuario lleve puesto el traje ya que tiene en cuenta por ejemplo el brazo que debe tapar parte del traje y el resultado queda más realista.



Figura 23: Esquema de construcción de RA realista.

Coherencia en la iluminación

El objeto virtual a renderizar no sólo debe tener coherencia geométrica ya que como se observa en las imágenes del ejemplo del traje de buzo, aún parece no ser suficiente para que parezca real. Hace falta pues adaptar la iluminación del objeto virtual a la iluminación del entorno real. Esto es lo que se llama coherencia lumínica.



Figura 24: Coherencia lumínica. La imagen de la izquierda es una foto real mientras que en la de la derecha se ha introducido una estatua de forma virtual.⁴¹

En la Figura 24 la estatua se integra totalmente en la escena gracias a los efectos de luz así como la distribución de sombras. Aunque este ejemplo es de carácter fotográfico se aplica también en las aplicaciones móviles de Realidad Aumentada.⁴²

Concretamente en aplicaciones de Realidad Aumentada donde el usuario se mueve o bien el objeto o contenido cambia su orientación, es necesario capturar la luz del entorno de forma continuada para iluminar los objetos virtuales. Para ello existe los llamados *light probe* o sondas/sensores de luz que recogen las condiciones de la luz incidente en un punto particular del espacio.

⁴¹ Fuente: http://kevinkarsch.com/wp-content/uploads/2014/03/sa11-rsolp_3-1024x320.jpg

⁴² Link: Coherencia lumínica en RA, <https://www.youtube.com/watch?v=P3roYsl6pM>



2.2.5. Aplicaciones de Realidad Aumentada

Dado que el proyecto trata sobre el desarrollo de una aplicación educativa y dirigida a utilizarse en clase en este apartado se presentan algunos ejemplos de aplicaciones similares ya existentes.

4D ANATOMY



Desarrollador: DAQRI
Plataforma: Android y iOS.

Esta aplicación⁴³ como su propio nombre indica muestra la anatomía del cuerpo humano haciendo uso de la realidad aumentada. Para probarla hace falta descargar primero el target o imagen que se va a utilizar como marcador e imprimirlo (Figura 25).

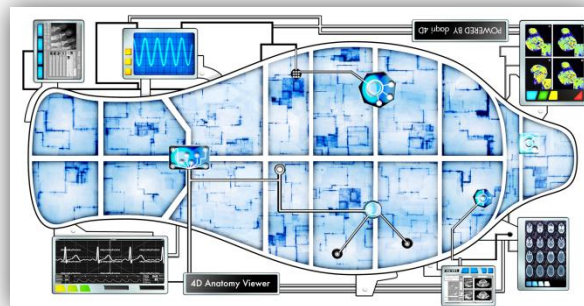


Figura 25: Target de Anatomy 4D ⁴⁴

Una vez impreso y con la aplicación descargada se debe enfocar con la cámara del móvil o tablet hacia la imagen. Lo que aparece entonces es un modelo en 3D de un cuerpo humano con el que se puede interactuar. Puedes ver los huesos o los músculos por separado, puedes escoger si quieres que aparezcan los órganos, etc. El usuario puede escoger activar los contenidos o no con las pestañas que se muestran en pantalla (Figura 26).

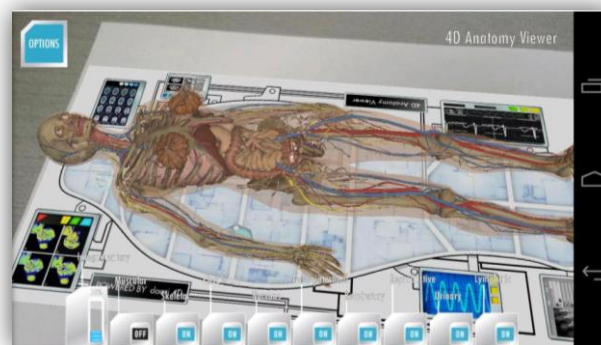


Figura 26: Anatomy 4D. Imagen de la aplicación. ⁴⁵

⁴³ Link: Web de la aplicación, <http://site.daqri.com/project/anatomy-4d/>

⁴⁴ Fuente: http://site.daqri.com/content/uploads/2013/05/target_4D_anatomy.png

⁴⁵ Fuente: <https://play.google.com/store/apps/details?id=com.daqri.d4DAnatomy>

ELEMENTS 4D

Desarrollador: DAQRI
Plataforma: Android y iOS.

Aplicación⁴⁶ dedicada al mundo de la química y sus elementos. Consiste en unos cubos de madera o de papel inscritos con los símbolos de 36 elementos de la tabla periódica, de forma que actúan como target o marcadores. Si se enfocan los cubos aparecen los elementos químicos con su símbolo y sus características (Figura 27). Además si se unen dos cubos con dos elementos aparece en cada cubo el compuesto químico resultante.



Figura 27: Cubos de Element 4D mostrando los elementos K y Cl⁴⁷

SKY MAP

Desarrollador: Google
Plataforma: Android y iOS

Aplicación⁴⁸ que permite ubicar en tiempo real estrellas, planetas y constelaciones, ideal para introducir o enseñar conceptos sobre la astronomía. Sólo hace falta enfocar la cámara del móvil o tablet hacia el cielo para que se muestren los astros tal y como se ve en la Figura 28. Está ideada para funcionar tanto de día como de noche.



Figura 28: Mapa con constelaciones, planetas y estrellas que muestra Sky Map⁴⁹

⁴⁶ Link: Web de la aplicación, <http://daqri.com/elements4D-mobile/>

⁴⁷ Fuente: <https://play.google.com/store/apps/details?id=com.daqri.elements4dbydaqri>

⁴⁸ Link: Web de la aplicación, <https://play.google.com/store/apps/details?id=com.google.android.stardroid&hl=en>



A medida que el usuario enfoca en diferentes direcciones va viendo los diferentes astros. También se pueden buscar planetas o estrellas y la aplicación muestra entonces indicaciones para encontrarlos (Figura 29).



Figura 29: Indicaciones dadas para encontrar un planeta⁵⁰

WORD LENS



Desarrollador: Quest Visual (Google)
Plataforma: Android y iOS

Aplicación⁵¹ que permite traducir el texto de un cartel, una señal y demás tipos de texto impreso, de una lengua a otra. Funciona a tiempo real, el usuario enfoca con su cámara hacia el texto e inmediatamente la aplicación lo traduce al idioma que desee y lo sustituye en la imagen. Actualmente permite las siguientes traducciones:

- inglés↔ruso
- inglés↔italiano
- inglés↔francés
- inglés↔español
- inglés↔alemán
- inglés↔portugués



Figura 30: Word Lens traduciendo al inglés un cartel en español⁵²

PLANETS 3D



Desarrollador: Popar Toys
Plataforma: Android y iOS

Aplicación⁵³ que permite ver contenido de Realidad Aumentada en un libro que se vende por separado, destinado a los niños. Aparecen animaciones del

⁴⁹ Fuente: <http://pequetablet.com/wp-content/uploads/2013/08/Skymap-cielo.jpg>

⁵⁰ Fuente: <http://pequetablet.com/wp-content/uploads/2013/08/Skymap-buscar-mover.jpg>

⁵¹ Link: Web de la aplicación, <https://play.google.com/store/apps/details?id=com.questvisual.wordlens.demo>

⁵² Fuente: <http://digitantes.com/wp-content/uploads/2014/05/wordlens.jpg>

Sistema Solar, de naves espaciales, satélites y demás material multimedia (Figura31).



Figura 31: Imagen de muestra de Planets 3D. El usuario enfoca la imagen y aparece un planeta. ⁵⁴

⁵³ Link: Web de la aplicación, <https://popartoy.com/shop/>

⁵⁴ Fuente: https://popartoy.com/shop/images/2790/mobile-examples_0006_space.png



3. Diseño y desarrollo

En este apartado se detalla todo el proceso de creación de la aplicación: el software utilizado en su desarrollo tanto para crear las funcionalidad de Realidad Aumentada como para construir los modelos o imágenes que se representarán, el diseño de las aplicaciones y finalmente su resultado final.

3.1. Introducción y funcionalidades

3.1.1. Plataformas de desarrollo

La aplicación se desarrolla para dispositivos móviles como smartphones y tablets, así pues es necesario conocer qué plataformas existen actualmente y qué presencia tienen en el mercado⁵⁵. Las principales plataformas en cuanto a volumen de mercado son actualmente: Android, iOS, Windows Phone y BlackBerry 10. Esto se puede comprobar en la Figura 32.

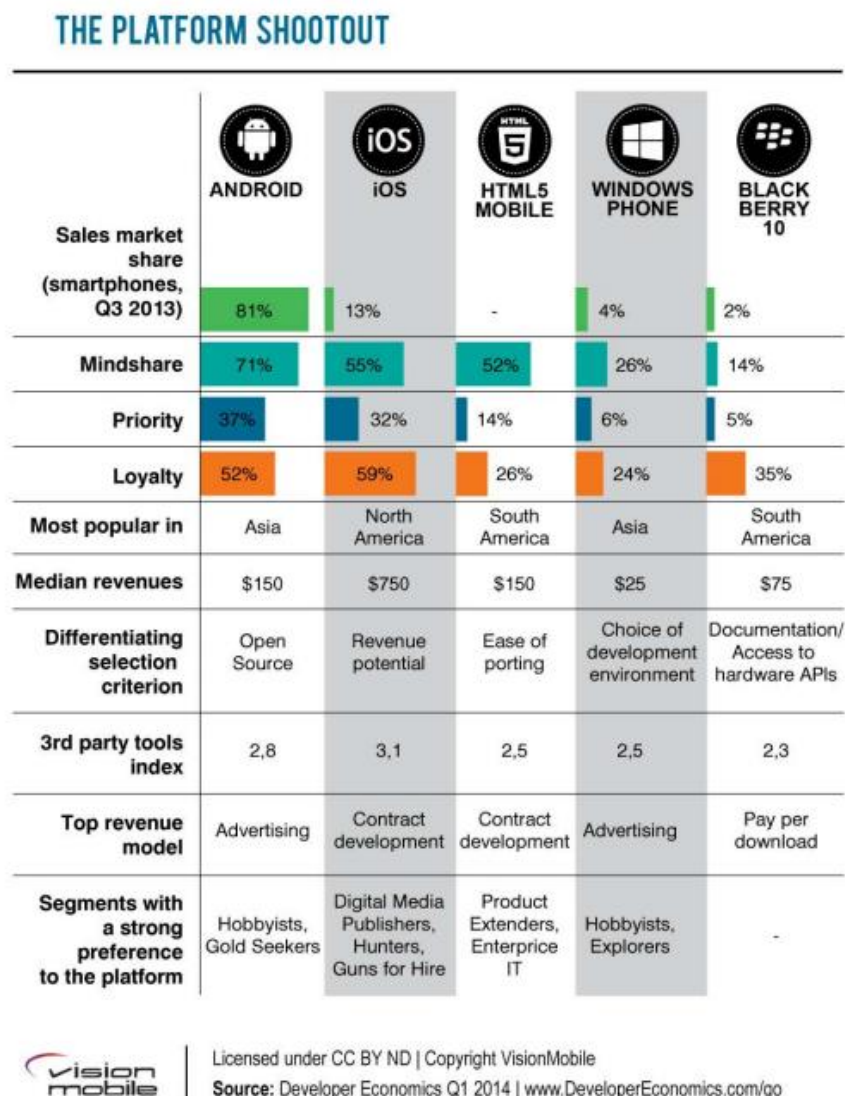


Figura 32: Comparativa de plataformas de desarrollo⁵⁵

⁵⁵ Fuente: Developer Economics Q1 2014, www.DeveloperEconomics.com/go

Android e Iphone abarcan el 94% de todo el mercado mundial de móviles, por lo tanto son las dos plataformas sobre las que podría interesar más desarrollar. De este 94%, el 71% desarrolla para Android mientras que el 55% desarrolla par iOS, por lo que se llega a la conclusión de que una gran cantidad de usuarios desarrolla en ambas plataformas.

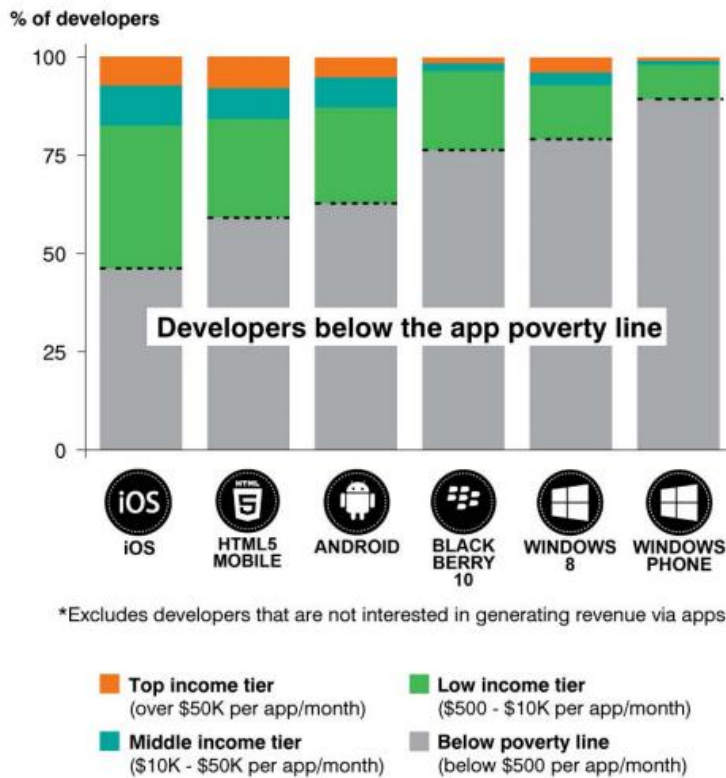
Android



Es el sistema operativo para móviles más extendido en el mundo con una presencia en un 81% de los terminales siendo ésta una de sus mayores virtudes. Aunque si por algo se caracteriza es por ser de código abierto. Con respecto a la obtención de ingresos cuando se desarrolla una aplicación en esta plataforma son los anuncios la mayor fuente de ingresos. Otro factor a destacar es el perfil de usuario que desarrolla en Android: gente que no se dedica profesionalmente a ello sino como hobby y profesionales.

IOS OFFERS A MORE EQUITABLE REVENUE DISTRIBUTION

% of developers in each revenue



vision mobile | Licensed under CC BY ND | Copyright VisionMobile
 Source: Developer Economics Q1 2014 | www.DeveloperEconomics.com/go

Figura 33: Distribución de desarrolladores según los ingresos⁵⁶

Otro dato a resaltar es que Android presenta un 38% de aplicaciones viables (Figura 33), es decir, que ingresan más de 500\$ por aplicación y mes. Sin embargo hay que tener en cuenta que mucha gente desarrolla por puro hobby o diversión y no está interesada en ingresar dinero, por lo que este dato hay que ponerlo en contexto.

⁵⁶ Fuente: Developer Economics Q1 2014, www.DeveloperEconomics.com/go



Es además la plataforma de desarrollo preferida por los usuarios, con un 37% de votos.

Google Play⁵⁷ es la tienda de aplicaciones para Android y para conseguir publicar aplicaciones en ella es necesario hacer un único pago para siempre de 25\$.⁵⁸

iOs


iOS Es el segundo sistema operativo más extendido entre móviles en el mundo con un 13% de la cuota de mercado. El 55% de los desarrolladores actuales utilizan iOs y los que se dedican en exclusiva actualmente a iOs son un 59% en frente del 52% de Android. En referencia a la obtención de ingresos la vía principal es mediante contratos de desarrollo, es decir, empresas que pagan al desarrollador para que realice una aplicación para ellos. Esto es destacable, ya que se calcula que el 56% de la economía asociada al mundo de las aplicaciones móviles proviene de esos mismos contratos. Quizás es por esto que el perfil más extendido en iOs es el de empresas y emprendedores de las tecnologías de la información, así como medios de comunicación digitales.

Respecto a la viabilidad económica, el 54% de las aplicaciones en iOs son viables. Cabe resaltar que el principal atractivo de esta plataforma son los ingresos y la rentabilidad que ofrece.

Otro hecho a remarcar es que de los desarrolladores que escogen las tablet como dispositivo de desarrollo principal, un 52% lo hace con iOs siendo el sistema operativo más utilizado en esta plataforma para desarrollar.

Apple Store⁵⁹ es la tienda de aplicaciones de iOs y requiere de un pago anual de 99\$ por ser desarrollador.⁶⁰

Windows Phone

 **Windows Phone** Es el tercer sistema operativo más presente en móviles con un 4% del mercado mundial. Como factor diferenciador más importante se encuentra la opción de seleccionar el entorno de desarrollo. Su vía de ingresos más importante son los anuncios. Su perfil de desarrollador es gente que lo hace por diversión y eso explica el 21 % (poco) de viabilidad económica de las aplicaciones de Windows Phone 8-

También cabe destacar que un 25% de los desarrolladores tiene pensado desarrollar en Windows Phone próximamente. Por último destacar que su cuota de mercado en tablets, como plataforma principal de desarrollo, es del 4%.

Para poder vender la aplicación por Windows Phone⁶¹ es necesaria una licencia anual de 19\$⁶².

⁵⁷ Link: Google Play, <https://play.google.com/store?hl=es>

⁵⁸ Link: <http://developer.android.com/distribute/googleplay/start.html>

⁵⁹ Link: Apple Store, <http://store.apple.com/es>

⁶⁰ Link: <https://developer.apple.com/programs/ios/>

⁶¹ Link: Windows Phone, <http://www.windowsphone.com/es-es/store>

⁶² Link: <http://betabeers.com/forum/cuanto-cuesta-desarrollar-moviles-156/>

BlackBerry



Es el cuarto sistema operativo en presencia en los móviles de todo el mundo con un 2% del total. Como factor diferenciador se le atribuye la documentación del software y el acceso a las APIs del hardware. Los ingresos en el caso de BlackBerry vienen fundamentalmente por los pagos por descarga de la aplicación. Actualmente está perdiendo usuarios, de hecho es la única de las 4 presentadas aquí que entre ingresos nuevos y abandonos acaba perdiendo usuarios (Figura 34).

MOBILE PLATFORMS: WINNERS AND LOSERS Q1 2014

% of developers planning to adopt or drop a platform (n=6,311)

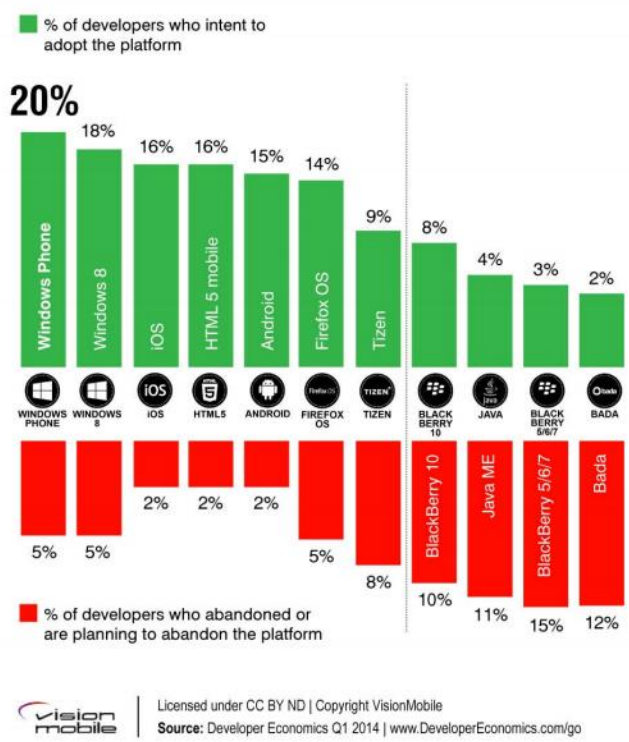


Figura 34: Reclutamientos y abandonos de desarrolladores en cada plataforma⁶³

Su viabilidad económica es del 26%. BlackBerry posee una licencia gratuita.⁶⁴

Plataforma de desarrollo de la aplicación

ELECCIÓN: Android

RAZONES: En primer lugar y principalmente se ha escogido Android porque es el sistema operativo del que dispone el autor en su teléfono móvil. Además su licencia no es excesivamente cara (unos 25\$) y sólo se paga una vez. También una



⁶³ Fuente: Developer Economics Q1 2014, www.DeveloperEconomics.com/go

⁶⁴ Link: <https://developer.blackberry.com/blackberryworld/>



de las razones de mayor peso es que es el sistema operativo más extendido del mundo, y dado que es una aplicación educativa se considera prioritario el número de personas a las que va dirigido antes que al beneficio económico.

3.1.2. Entornos de desarrollo

SDKS

El desarrollo de aplicaciones de Realidad Aumentada se puede realizar a través de varios SDK (*Software Development Kit*) disponibles.

Un SDK o kit de desarrollo de software, es un conjunto de herramientas y/o programas de que permiten al programador crear aplicaciones para un determinado software, sistema operativo o similar.

Otra definición de SDK podría ser el conjunto de librerías que contiene el código que se utiliza más comúnmente y de forma repetitiva en el desarrollo de aplicaciones. Según se quiera desarrollar para Android, Iphone, Windows u otras plataformas, se deberá utilizar un SDK determinado, aunque hay muchos que permiten publicar en múltiples plataformas al mismo tiempo.

Las herramientas que proporciona principalmente son recursos que simplifican la implementación de las siguientes funciones:

- Reconocimiento: es la función que proporciona la habilidad para comprender que lo que está viendo el dispositivo (Iphone, PC, etc.)
- Tracking (seguimiento o rastreo): son los ojos de la aplicación.
- Rendering de contenidos (generación de contenidos gráficos): en esta función es dónde se crea el contenido que el usuario final verá.

SDK's para desarrollar en RA

Existen diversos SDK con los que se puede desarrollar Realidad Aumentada. El proyecto se desarrolla con el SDK Qualcomm Vuforia y su elección corresponde al hecho de que su aprendizaje es rápido, intuitivo y es una buena elección para los iniciados en Realidad Aumentada, además de tener licencia gratuita. A continuación se presenta una lista de SDK existentes así como sus principales características (Tablas 1,2 y 3).

Mobile AR SDKs Short List

SDK	Purpose	Tracking	Platform	Graphics	Cloud	GPS	License
Qualcomm Vuforia	2D Images, Markers	NFT, Marker, Text	iOS, Android	Unity3D & Alt.	yes	no	Free
Metaio SDK	2D Images, GEO, 3D, Anywhere	NFT, GPS, 3D, SLAM	iOS, Android	Unity3D & Alt.	yes	yes	0-\$10,000
Totalmm. D'fusion	2D Images, faces	NFT, Face	iOS, Android Dev: MS only	Unity3D & Alt.	no	no	0-\$10,000
WikitudeSDK	2D Images, GEO	GPS, NFT	iOS, Android, Blackberry	HTML, Proprietary	yes	yes	0-\$2,300
Layar SDK	2D images	NFT, QR	iOS, Android	Proprietary	yes	no	\$3,250/app/yr + \$20/page
13 th Lab Point Cloud	2D Images, 3D, Anywhere	NFT, 3D, SLAM	iOS, Android	Unity3D & Alt.	no	no	0-\$5,000

Tabla 1: Lista de SDK para desarrollar Realidad Aumentada ⁶⁵

Mobile AR SDKs Short List – Cont'd

SDK	Purpose	Tracking	Platform	Graphics	Cloud	GPS	License
AR Toolkit	2D Markers 2D Images	NFT, Marker	iOS, Android	Unity3D & Alt.	no	no	\$0 (GPL) - \$2,495 + royalty
HP Aurasma SDK	VFX, Videos on 2D Images	NFT	iOS, Android	Proprietary	yes	yes	Per click
Obvious Engine	2D images, Cylinders	NFT, Cyl.	iOS, Android	Unity3D & Alt.	no	no	\$8000/app/platform
String	Framed 2D images	Framed Image	iOS	Unity3D & Alt.	no	no	\$499 - \$8000/app/year
Orbotix	Robotic AR	Sphero	iOS, Android	Unity3D	no	no	Free




Tabla 2: Lista de SDK para desarrollar Realidad Aumentada. *NFT: Natural Feature Tracking, permite que una imagen, un objeto, una cara o una mano por ejemplo, puedan servir como targets. ⁶⁶

⁶⁵ Fuente: <http://www.onlyaugmented.com/wp-content/uploads/2013/09/augmetned-reality-SDK-table-showing-license-page-1.jpg>

⁶⁶ Fuente: <http://www.onlyaugmented.com/wp-content/uploads/2013/09/augmetned-reality-SDK-table-showing-license-page-2.jpg>









APIS	iOS/ANDROID	Hybrid integration	Marker creation	Free	Content (3D, video...)
	✓	✓	✓	✓	✓
	✓	✓	✓	✓	✓
	✓	✗	✓	✓	✓
	✓	✓	✓	✗	✓
	✓	✓	✓	✓	✓
	✓	✓	✓	✗	✓

Tabla 3: Comparativa de APIs de desarrollo de realidad aumentada.⁶⁷

Tal y como se observa en la Tabla 3, Qualcomm Vuforia presenta la mayor parte de funcionalidades esenciales con la ventaja de tener una licencia entera gratuita. Metaio y Aurasma también disponen de potentes herramientas pero son de uso comercial.

Aquí se detallan algunos de los SDK mostrados en las Tablas 1, 2 y 3.

QUALCOMM VUFORIA⁶⁸

Es una plataforma de software para Android y iOS que permite a la aplicación ver imágenes de una base de datos de Vuforia localizada en el dispositivo del usuario (móvil, tablet) o bien que puede estar localizada en la nube o cloud.



Figura 35: Logotipo de Qualcomm Vuforia⁶⁹

Cuando una imagen o target (nombre utilizado más habitualmente) es reconocido por la cámara del dispositivo se genera el contenido de realidad aumentada.

Puede reconocer diferentes tipos de targets: textos, imágenes en 2D, objetos cilíndricos como botellas o latas, objetos simples en 3D y los llamados Frame Markers (Figura 36), que son unas imágenes con unas marcas en los bordes parecidas a los códigos QR.

⁶⁷ Fuente: Del autor.

⁶⁸ Link: Página web oficial de Vuforia : <https://www.vuforia.com/>

⁶⁹ Fuente: <http://www.qualcomm.com/sites/default/files/pods/promo/vuforia-cloud3.png>

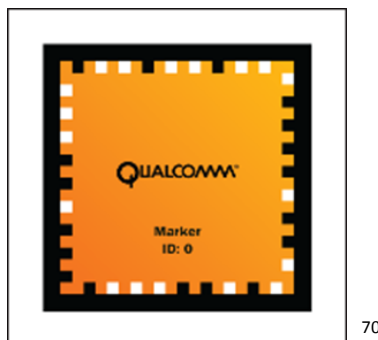


Figura 36: Frame Marker de Vuforia ⁷⁰

Una de las ventajas de Vuforia es que las aplicaciones se pueden publicar en varias plataformas: Android, iOS, Playstation, Windows Phone, etc. De esta forma se puede desarrollar la aplicación y luego decidir en qué plataformas publicarla evitando así tener que volver a rehacerla para cada plataforma.

Es un SDK que se puede implementar con el motor de videojuegos Unity 3D de forma que se pueden crear contenidos multimedia complejos. Para poder utilizarlo se debe tener conocimientos de Unity. También se puede utilizar la plataforma Eclipse en lugar de Unity.

Por último destacar que Vuforia es de licencia gratuita así como Unity, una gran ventaja delante de sus competidores algunos de los cuáles requieren de licencias de pago. Sin embargo hay funcionalidades de Vuforia que se sólo se pueden ampliar pagando, aunque está más enfocado a un desarrollo de aplicaciones profesional.

METAIO SDK ⁷¹



Figura 37: Logotipo de Metaio ⁷²

METAIO SDK es otra opción a considerar, puede desarrollarse en Android, iPhone y Windows. Al igual que Vuforia también se puede integrar con el motor Unity3D. Integra también un potente motor de renderizado en 3D.

Dispone de servicio en la nube como Vuforia y al igual que él también implementa los llamados Frame Markers.

Como plataformas para desarrollar se encuentra Unity, Eclipse y una plataforma propia llamada Metaio Creator. Ésta última dispone de una versión de prueba gratuita y está pensada para desarrolladores sin conocimientos de programación.

Es de licencia principalmente comercial: el SDK sin limitaciones cuesta 4.950 €. Si no se quiere pagar hay una versión del SDK pero que introduce una marca de agua en las aplicaciones.

⁷⁰ Fuente: https://developer.vuforia.com/sites/default/files/dev_guide_images/create_targets_devguide/frame_marker_example_create_targets.png

⁷¹ Link: Página web oficial de Metaio, <http://www.metaio.com/>

⁷² Fuente: http://www.avangate.com/docs/en/client_case/metaio.jpg?20131001131710



WIKITUDE SDK⁷³*Figura 38: Logotipo de Wikitude*⁷⁴

Wikitude SDK permite diseñar aplicaciones en iOS, Android y Blackberry. En referencia al contenido de la aplicación se puede crear utilizando JavaScript, CSS y HTML. La extensión del SDK se puede implementar con las plataformas PhoneGap (Adobe Cordova), Appcelerator Titanium y MonoTouch, MonoDroid.

Una característica reseñable es que también proporciona un SDK para desarrollar aplicaciones destinadas a gafas de realidad aumentada. Las gafas de AR para las cuáles es compatibles este SDK son: EPSON MOVERIO, GOOGLE GLASS y VUZIX.

Dispone de versión gratuita de prueba y de una versión comercial en versión pro desde 1.499€.

AURASMA SDK⁷⁵*Figura 39: Logotipo de Aurasma*⁷⁶

Pertenece actualmente al grupo tecnológico Hewlett Packard (HP) es una SDK que permite desarrollar en Android e Iphone y que puede mostrar contenidos en 3D, imágenes en 2D, videos, etc.

Es de licencia comercial y dispone de una plataforma propia de creación de aplicaciones llamada Aurasma Studio.

Entorno de desarrollo de la aplicación

ELECCIÓN: Unity + Vuforia

RAZONES: La razón para escoger a Unity como entorno es que es un motor de videojuegos que dispone de una versión totalmente gratuita y es multiplataforma (se puede publicar la aplicación en Android, iOS, Windows Phone, etc.), que además incorpora una gran cantidad de

⁷³ Link: Página web oficial de Wikitude, <http://www.wikitude.com/>

⁷⁴ Fuente: <http://www.wikitude.com/wp-content/uploads/2014/02/logo1.png>

⁷⁵ Link: Página web oficial de Aurasma, <http://www.aurasma.com/>

⁷⁶ Fuente: http://www.eiu.edu/cats/home/media_images/aurasma_logo.jpg

utilidades y características a la hora de crear y manejar contenidos gráficos. Con respecto al SDK de Vuforia la razón es que también dispone de una versión totalmente gratuita para desarrolladores y se puede integrar perfectamente con Unity.

Cabe mencionar que tanto Unity como Vuforia tienen versiones de pago para uso más profesional.



3.1.3. Lenguajes de programación

Una vez escogido Unity como entorno de desarrollo éste presenta tres posibles lenguajes de programación a utilizar que son:

- C# o C Sharp.
- JavaScript.
- Boo.

C# o C Sharp

Es un lenguaje de programación orientado a objetos desarrollado por Microsoft desde el año 2000. Su sintaxis básica deriva del C/C++ y utiliza un modelo de objetos similar al de Java. Algunos de sus objetivos principales son:

- Lenguaje simple, moderno y de carácter general.
- Fácil migración del programador familiarizado con C, C++ y Java.
- Adecuación para construir aplicaciones de cualquier tamaño.

Ejemplo: Función que permite escalar un objeto en Unity.

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour {
    void Example() {
        transform.localScale += new Vector3(0.1F, 0, 0);
    }
}
```

JavaScript

Es un lenguaje de programación interpretado (analizado y ejecutado en tiempo de ejecución), orientado a objetos y basado en prototipos (un estilo de programación orientada a objetos en que la creación de objetos no se realiza mediante instancias a una clase). Su sintaxis es similar



al C aunque adapta términos y convenciones de Java. Es utilizado principalmente en los navegadores web desarrollando interfaces de usuario y páginas web dinámicas. Se viene desarrollando desde el año 1995.

Ejemplo: Función que permite escalar un objeto en Unity.

```
// Widen the object by 0.1
transform.localScale += Vector3(0.1,0,0);
```

Boo

Es un lenguaje de programación orientado a objetos desarrollado desde el año 2004, con una sintaxis inspirada en Python y de código abierto. Además de Python también ha estado influido por C#.

Ejemplo: Función que permite escalar un objeto en Unity.

```
import UnityEngine
import System.Collections

public class ExampleClass (MonoBehaviour) :

    def Example() as void:
        transform.localScale += Vector3(0.1F, 0, 0)
```

Lenguaje de programación de la aplicación

ELECCIÓN: C#

RAZONES: Se ha escogido este lenguaje y no otro porque es el que más se parece a Java y C++, que son dos lenguajes con los que el autor se siente más cómodo. Tampoco hubiera sido problema desarrollar en Boo ya que tiene una gran similitud con Python.

3.2. Creación de contenidos digitales para la aplicación

En este apartado se detallan algunas herramientas de creación y diseño de contenidos gráficos que se pueden utilizar en el desarrollo de la aplicación. Se presentan clasificadas según el campo al que pertenecen en:

- Imagen.
- Gráficos 3D/Animación.
- Video.

3.2.1. Imagen

Reúne los programas de dedicados a la edición digital de imágenes, que consiste en poder modificar sus propiedades (luz, contraste, color, orientación, etc.) así como en manipularlas, escalarlas, recortarlas o cualquier otra operación necesaria (Figura 40).



Figura 40: Ejemplo de edición⁷⁷ de imagen: cambio de color

Adobe Photoshop

Adobe Photoshop⁷⁸ es un editor gráfico desarrollado por la empresa Adobe Systems utilizado principalmente para el retoque de imágenes y fotografías, y líder mundial en aplicaciones de edición de imagen. Es compatible con Windows y con Macintosh.

Si bien inicialmente poseía un espacio de trabajo de una única capa, a medida que ha ido evolucionando ha incorporado mejoras (Figura 41) como: espacio de trabajo multicapa, inclusión de elementos vectoriales, gestión avanzada de color, tratamiento extensivo de tipografías, control y retoque de color, efectos creativos, etc. Otra propiedad suya es que soporta muchos tipos de imágenes como BMP, JPC, PNG y GIF, entre otros.

Se trata de un software comercial por lo que hay que pagar una licencia comercial. Actualmente desde 12,29€/mes se puede disponer del software.⁷⁹



Figura 41: Interfaz de usuario de Adobe Photoshop CS6⁸⁰

⁷⁷ Fuente:

http://es.wikipedia.org/wiki/Edici%C3%B3n_de_im%C3%A1genes#mediaviewer/Archivo:Rgbtobandswexample11-28-200.jpg

⁷⁸ Link: Adobe Photoshop, http://es.wikipedia.org/wiki/Adobe_Photoshop

⁷⁹ Fuente: Precios Photoshop, <http://www.adobe.com/es/products/photoshop/buying-guide.html>

⁸⁰ Fuente: <http://3.bp.blogspot.com/-XerrvdRriis/UACWpBcTjNI/AAAAAAAAADU/mKKhuxHN9dM/s1600/adobe-photoshop-02-700x464.jpg>



Gimp

Gimp⁸¹ es un programa de edición de imágenes digitales de código abierto y gratuito (Figura 42). Es el editor disponible en más sistemas operativos: Linux, Windows, Mac, Solaris, etc. Como característica principal Gimp⁸² permite el tratado de imágenes en multicapa, permitiendo luego guardar la imagen final en formato .xcf que soporta capas o en formatos sin capas como pueden ser PNG, BMP, etc. En relación con los formatos Gimp soporta la mayoría incluyendo entre ellos JPG, GIF, PNG, PCX e incluso el formato de Photoshop. Su licencia actual es gratuita.



Figura 42.: Interfaz de usuario de Gimp 2.8.⁸²

Microsoft Paint

Paint⁸³ (originalmente Paintbrush) es un editor de fotografía e imágenes desarrollado por Microsoft (Figura 43). Su principal característica es su simplicidad, se trata de un programa que sólo permite realizar operaciones básicas de edición y dispone de las siguientes herramientas: selección libre de dibujo, selección, borrador/borrador de color, rellenar con color, tomar color, lupa, lápiz, pincel, aerógrafo, texto, línea, curva, rectángulo, polígono, elipse, y rectángulo redondeado.

Paint no tiene la capacidad de crear automáticamente gradientes de color. Por último mencionar que viene como programa de defecto en los sistemas operativos de Windows, de forma gratuita.

⁸¹ Link: Gimp, <http://www.gimp.org/es/>

⁸² Fuente: <http://img.genbeta.com/2012/05/gimp-2-8-en-windows-7.jpg>

⁸³ Link: Paint, <http://windows.microsoft.com/es-es/windows7/products/features/paint>



Figura 43: Interfaz de usuario de Paint⁸⁴

Programa de imagen utilizado

ELECCIÓN: Paint y AdobePhotoshop

RAZONES: Microsoft Paint es una herramienta sencilla pero eficaz, ideal si se tienen que realizar operaciones cortas como recortar una imagen. Photoshop es una plataforma necesaria para crear contenidos como los marcadores o adaptar resoluciones, saturación, etc. Se ha escogido éste y no Photoshop, porque el autor tiene experiencia con Photoshop.

3.2.2. Gráficos 3D/Animación

En este apartado se describen el conjunto de programas destinados a crear gráficos 3D por ordenador⁸⁴. Cabe mencionar que algunos de ellos también incorporan la posibilidad de realizar animaciones.

Blender

Blender⁸⁵ es un programa de creación de contenidos en 3D⁸⁴ que permite el modelado de objetos, animación y composición de escenas y renderización de escenas complejas en 3D (Figura 44). Es un software gratuito y en el cual se pueden desarrollar nuevas funcionalidades de forma particular mediante Python. Posee un robusto engine para programar juegos y un motor de render propio. Es compatible con Windows, Mac OS X, GNU/Linux, Solaris, FreeBSD e IRIX.

⁸⁴ Fuente: <http://www.scenebeta.com/archivos/www/Captura2Kpaint.jpg>

⁸⁵ Link: Blender, <http://www.blender.org/>





Figura 44: Interfaz de usuario de Blender⁸⁶

Autodesk 3ds Max

Autodesk 3ds Max⁸⁷, anteriormente conocido como 3D Studio Max, es un programa de creación de gráficos⁸⁷ y animación en 3D (Figura 45). Es uno de los programas de animación 3D más utilizados especialmente en la industria de los videojuegos, aunque también se utiliza en el desarrollo de proyectos de animación como películas o anuncios de televisión. Dispone de una sólida capacidad de edición y una arquitectura de plugins extensa. Su licencia es comercial su precio es de 195\$/mes.⁸⁸ La última versión es sólo compatible con Windows.

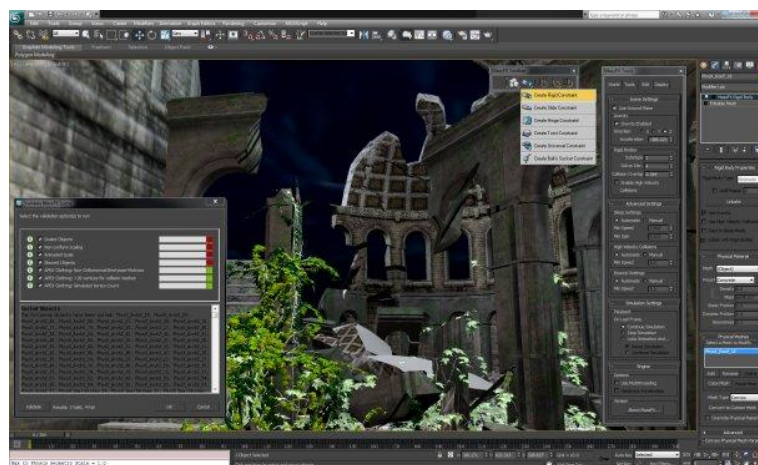


Figura 45: Interfaz de usuario de Autodesk 3ds Max⁸⁹

Autodesk Maya

Autodesk Maya⁹⁰, también conocido como Maya, es un programa de desarrollo de gráficos 3D por ordenador, efectos especiales y animación (Figura 46). Posee diversas herramientas para modelado, animación, renderización, simulación e ropa y cabello, dinámicas (simulación de fluidos), etc. Una de sus características principales⁹⁰ es lo abierto que es al software de terceros, siendo muy personalizable. La potencia y la flexibilidad son otros de sus rasgos. Es

⁸⁶ Fuente: <http://www.formaciononlinegratis.net/wp-content/uploads/2013/08/manual-de-blender-gratis.jpg>

⁸⁷ Link: Autodesk 3ds Max <http://www.autodesk.es/products/autodesk-3ds-max/overview>,

⁸⁸ Fuente: <http://www.autodesk.com/products/autodesk-3ds-max/buy>

⁸⁹ Fuente: http://www.cgchannel.com/wp-content/uploads/2011/03/110301_3dsMax2012.jpg

⁹⁰ Link: Autodesk Maya, <http://www.autodesk.es/products/autodesk-maya/overview>

compatible con Windows, Linux, y Mac OS X, y se trata de un software comercial cuyo precio es de 195\$/mes.⁹¹



Figura 46: Interfaz de usuario de Autodesk Maya⁹²

Programa de gráficos/3D utilizado

ELECCIÓN: Blender

RAZONES: Blender es un software gratuito y además permite tanto la creación de modelos en 3D como su posterior animación.

3.2.3. Video

La edición de vídeo es un proceso en el cual se elabora un trabajo audiovisual o visual a partir de archivos de video, fotografías, gráficos o animaciones. En este apartado se describen algunos editores de video.

Windows Movie Maker

Es un editor de vídeo⁹³ de Microsoft compatible con Windows (Figura 47). Su principal virtud es su sencillez y su rápido aprendizaje, destinada a un público general sin grandes conocimientos técnicos en edición, también es capaz de editar audio junto con el vídeo. Dispone de transiciones y efectos de vídeo, además de un creador de créditos. Como contras hay que resaltar que soporta pocos formatos de vídeo y sus efectos son limitados. Es gratuito.

⁹¹ Fuente: <http://www.autodesk.com/products/autodesk-maya/buy>

⁹² Fuente: <http://imagenes.es.sftcdn.net/es/scrn/64000/64306/autodesk-maya-1.jpg>

⁹³ Link: Windows Movie Maker, <http://windows.microsoft.com/es-es/windows-live/movie-maker#1=overview>



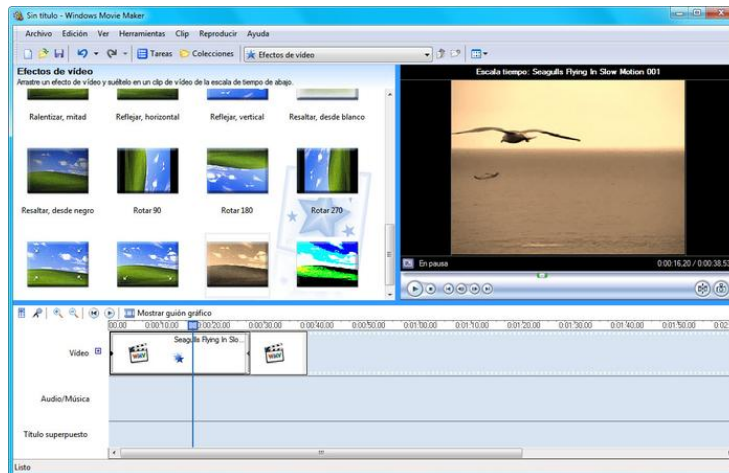


Figura 47: Interfaz de usuario de Windows Movie Maker⁹⁴

Adobe Premiere Pro

Es un editor de vídeo⁹⁵ profesional (Figura 48), y eso implica un nivel de efectos, propiedades y características complejo, con lo que puede resultar complicado para principiantes. Es compatible con las demás herramientas del grupo Adobe como Photoshop. Trabaja de forma nativa con una amplia variedad de formatos de vídeo y permite publicar en formatos como Blu-Ray. También incorpora una función que permite sincronizar proyectos desde varios ordenadores. Se trata de un software compatible con Windows y Mac, de carácter comercial. Actualmente la licencia tiene un coste de 24,59 €/mes.⁹⁶



Figura 48: Interfaz de usuario de Adobe Premiere Pro⁹⁷

Pinnacle Studio

Es un editor de vídeo⁹⁸ originalmente desarrollado para proporcionar una versión para el usuario medio, de Liquid Edition, un software de nivel profesional. Este software (Figura 49) permite publicar entre otros en DVD-Video y Blu-Ray. Incorpora las características típicas de un editor de vídeo como transiciones, efectos, etc. Destaca por ser fácil de usar, por el número de transiciones de que dispone y por disponer de una amplia biblioteca de ejemplos. Es

⁹⁴ Fuente: <http://imagenes.es.sftcdn.net/es/scrn/23000/23974/windows-movie-maker-06-700x452.jpg>

⁹⁵ Link: Adobe Premiere Pro, <http://www.adobe.com/es/products/premiere.html>

⁹⁶ Fuente: <http://www.adobe.com/es/products/premiere.html>

⁹⁷ Fuente: http://blogs.adobe.com/mallika/files/2013/06/premiere_screen.png

⁹⁸ Link: Pinnacle Studio, <http://www.pinnaclesys.com/PublicSite/sp/Products/studio/>

compatible con la mayoría de formatos de video como AVCHD, DV, HDV, AVI, MPEG-1, MPEG-2... Es de licencia comercial con un coste desde 59,95 €⁹⁹. Es compatible con Windows.

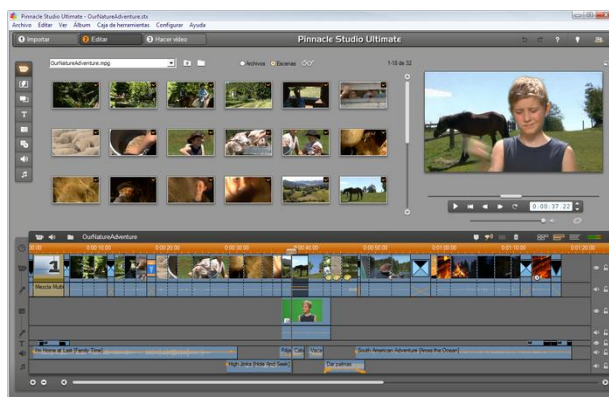


Figura 49: Interfaz de usuario de Pinnacle Studio ¹⁰⁰

Programa de video utilizado

ELECCIÓN: Windows Movie Maker

RAZONES: Es un programa sencillo y gratuito, ideal para hacer montajes simples de forma rápida.

3.3. Diseño de la aplicación

El objetivo de la aplicación es crear una serie de experiencias de realidad aumentada, todas ellas pertenecientes al campo de la Física. En concreto se trata de 4 experiencias o animaciones que tratarán los temas del electromagnetismo, las energías, el rozamiento y el movimiento de los planetas del Sistema Solar.

3.3.1. Diseño Experiencia 1: Ciclotrón

Contexto

Esta experiencia quiere explicar el funcionamiento de un ciclotrón (Figura 51). Un ciclotrón es un aparato formado por dos piezas de forma semicilíndrica (llamadas D) ligeramente separadas cuyo objetivo es acelerar partículas cargadas (p.ej. protones) hasta velocidades elevadas.

Para conseguir acelerar la partícula son necesarios un campo eléctrico (en color rojo en la figura 50) y un campo magnético (en color verde en la figura 50).

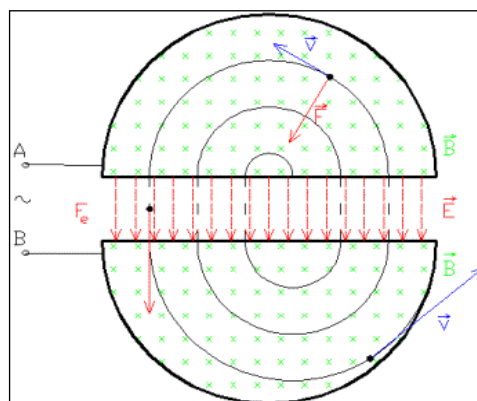


Figura 50: Esquema de un ciclotrón

⁹⁹ Fuente: <http://www.pinnaclesys.com/PublicSite/sp/Products/studio/standard/>

¹⁰⁰ Fuente: <http://imagenes.es.sftcdn.net/es/scrn/54000/54518/pinnacle-studio-20.png>



Inicialmente la partícula se coloca entre las dos D, justo en el centro del ciclotrón. Como se observa hay un campo eléctrico en el espacio entre las D, que sólo actúa allí. La partícula cargada entonces se acelera debido al campo eléctrico que experimenta aumentando así su velocidad y se dirige a la vez hacia una de las dos D. Cuando llega a una de las D, allí no existe campo eléctrico por lo que su velocidad deja de aumentar, pero en cambio es aquí donde sí actúa un campo magnético (color verde) perpendicular a la trayectoria de la partícula. Este campo hace que aparezca una fuerza magnética que a su vez provoca que la partícula realice un movimiento circular y vuelva entrar al espacio entre las dos D. Entonces antes de que entre, se cambia el sentido del campo eléctrico para que la partícula al entrar se vuelva acelerar y aumente aún más su velocidad.

La idea general es que cada vez que pasa por el centro se acelera y en las D lo que se hace es cambiar la trayectoria de la partícula para que vuelva a pasar por el medio.

Dado que se trata de un sistema cuyo funcionamiento puede llegar a ser confuso y complicado de explicar con palabras se pensó en realizar una animación.

Diseño

Se pensó en construir un modelo en 3D de un ciclotrón de forma muy simplificada, de hecho se siguió un modelo similar al de la figura 50. La idea era añadir dos tipos de flechas para los dos tipos de campo, campo eléctrico y campo magnético. A la vez se representaría la partícula con la forma de una esfera y también se dibujaría la trayectoria que seguiría encima del ciclotrón (Figura 51).

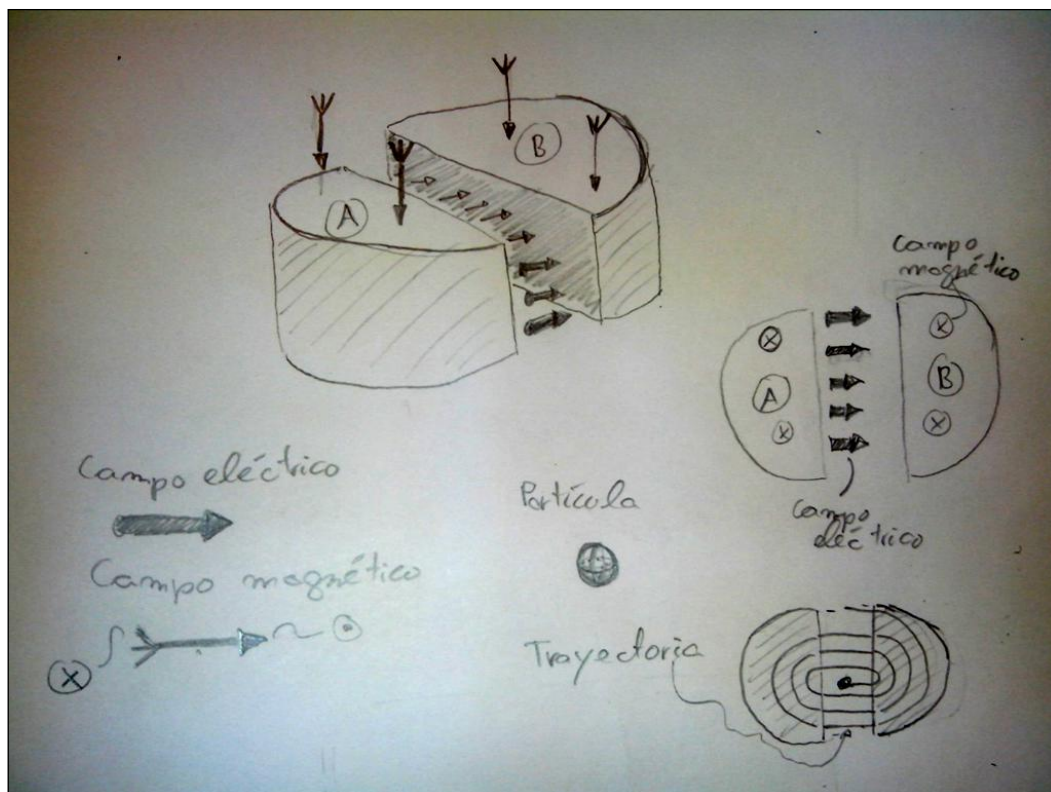


Figura 51: Esbozo del ciclotrón y sus partes

Una vez creadas, las flechas de campo eléctrico y magnético se moverían de forma constante formando un flujo. Además el usuario tendría la opción de poder seleccionar sí quiere ver el campo magnético y/o el campo eléctrico, pulsando en dos botones correspondientes a cada tipo de campo que aparecerían en la pantalla del dispositivo móvil. Mientras, la esfera que representa a la partícula se iría moviendo siguiendo la trayectoria dibujada.

Otro elemento presente en la pantalla del dispositivo móvil sería un botón de Play que aparecería una vez se acabase la animación y que al pulsarlo permitiría volver a reproducirla.

Reconocimiento y Tracking

El elemento sobre el que se creará la animación, es decir, el target, será una imagen del ciclotrón (Figura 52). Es lo que en Vuforia se conoce como ImageTarget.

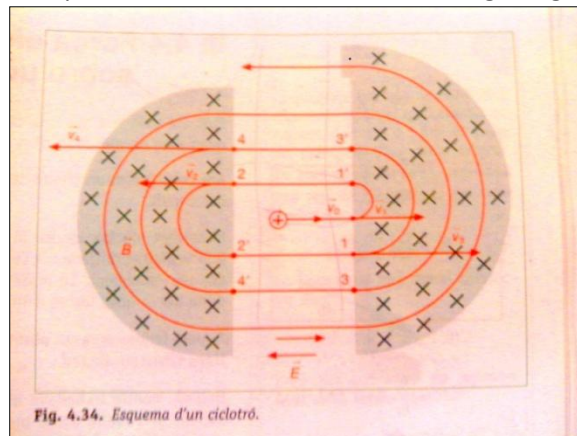


Figura 52: ImageTarget del ciclotrón.

Respecto al flujo de ejecución de la aplicación se distinguen tres posibles estados:

- Target no encontrado: cuando el usuario activa la aplicación por primera vez y no se enfoca o reconoce el target, o bien porque se ha cerrado previamente la animación. No aparecerá ningún elemento tan sólo la imagen real que capta la cámara.
- Target encontrado: cuando la cámara del dispositivo reconoce el target. Entonces aparece el ciclotrón y se anima inmediatamente con todos sus elementos y botones (Figura 53).

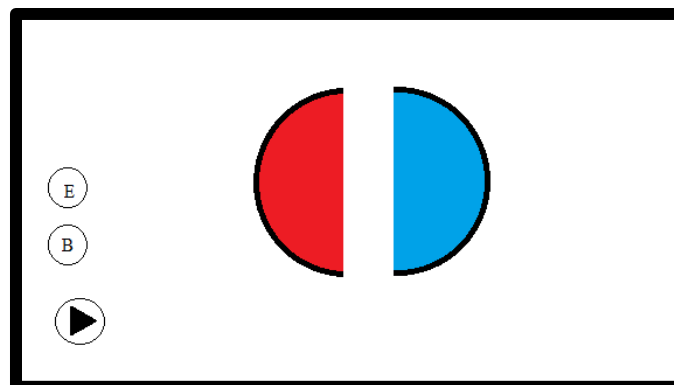


Figura 53: Reproducción de la vista por pantalla en Target encontrado



- **Target perdido:** si estando en el estado *de Target encontrado*, la cámara deja de captar el target, bien porque se ha desenfocado la cámara o porque el usuario ha dejado de apuntar hacia el target. Aquí se seguirá representando la animación con los mismos elementos y botones que en el estado *Target encontrado*. La única diferencia será la aparición de un botón en la pantalla que permitirá cerrar la animación (Figura 54) y pasar por lo tanto al estado de *Target no encontrado*.

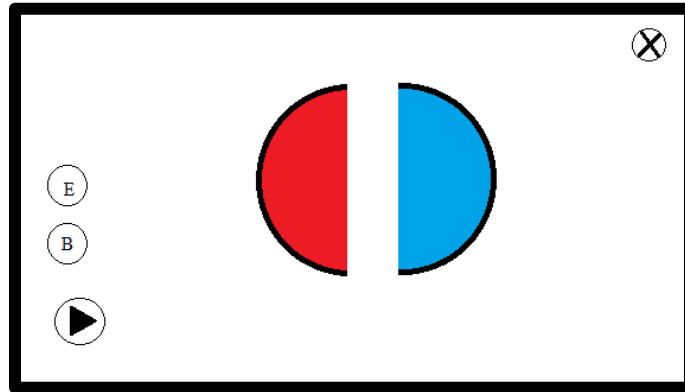


Figura 54: Reproducción de la vista por pantalla en Target perdido

3.3.2. Diseño Experiencia 2: Rampa con rozamiento

En esta experiencia se pretende que el usuario aprenda de la importancia del coeficiente de rozamiento en el movimiento de los objetos y de cómo su valor es independiente de la masa que tenga el objeto. Para ello se lanzará un objeto desde lo alto de una rampa que forma un ángulo α con la horizontal. El objeto entonces tendrá un movimiento uniformemente acelerado cuya aceleración viene dada por la expresión:

$$a = g \sin \alpha - \mu_d g \cos \alpha$$

Siendo g el valor de la gravedad en la superficie de la Tierra y μ_d el coeficiente de rozamiento dinámico. Como se puede observar, la aceleración no depende de la masa que tenga el objeto, y por lo tanto tampoco el tiempo de caída depende de la masa.

Otro aspecto a destacar es que la superficie y la rampa poseen un coeficiente estático de rozamiento μ_e , que es el rozamiento que existe cuando el objeto no se mueve. A su vez es el coeficiente de rozamiento mínimo que hace que el objeto no caiga por la rampa sino que se quede quieto en lo alto, y su expresión viene dada:

$$\mu_e = \tan \alpha$$

Siendo α el ángulo que la rampa forma con la horizontal (Figura 55). Se observa que el coeficiente de rozamiento límite sólo depende del ángulo α de la rampa.

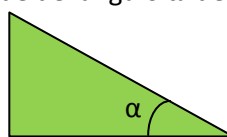


Figura 55: Esquema de la rampa

Diseño

El diseño de la aplicación consiste en dos elementos: una rampa y un cubo que bajará por ella. El usuario será capaz de modificar la masa del cuerpo así como el rozamiento entre el cubo y la rampa. Una vez se decida iniciar la animación el cubo deslizará por la rampa hacia abajo mientras su movimiento se ve frenado en mayor o menor medida según el coeficiente de rozamiento fijado. El usuario dispondrá de un cronómetro en pantalla que le permitirá medir el tiempo que tarda en bajar el cubo la rampa, para luego sacar conclusiones sobre qué factores afectan al descenso del objeto.

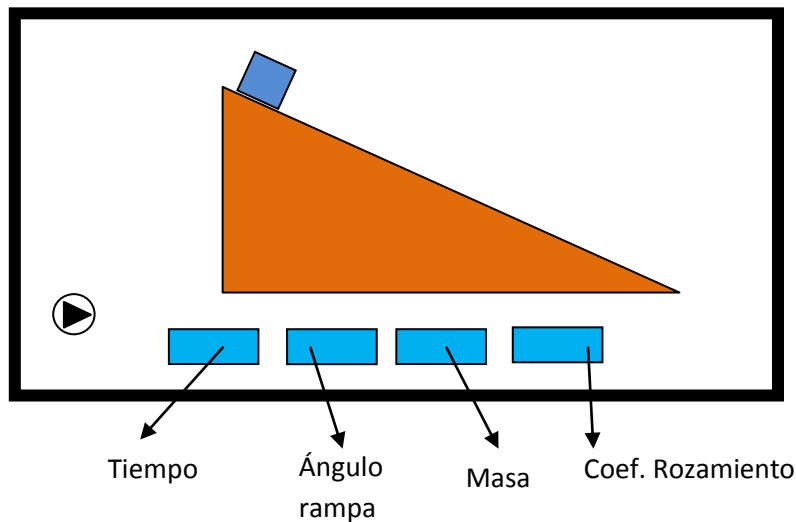


Figura 56: Diseño de la vista por pantalla.

Además del tiempo también se mostrará por pantalla (Figura 56) el ángulo del plano (que tiene una relación directa con el ángulo de rozamiento), la masa del objeto y el coeficiente de rozamiento. También incorporará el botón de reproducir de la animación así como el de cerrarla, comentados en el apartado [3.3.1. Diseño Experiencia 1: Ciclotrón.](#)

Reconocimiento y Tracking

El flujo de estados es el mismo que el detallado en el apartado [3.3.1. Diseño Experiencia 1: Ciclotrón](#) y que está formado por los tres estados: target encontrado, target perdido y target no encontrado.

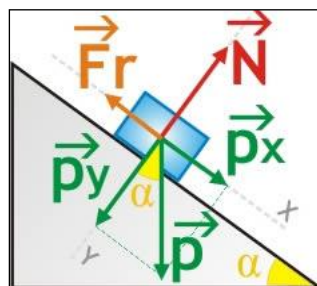


Figura 57: ImageTarget de Rampa con rozamiento.



3.3.3. Diseño Experiencia 3: Balance de energías

Esta experiencia tiene como objetivo practicar y entender mejor los problemas de balances de energías en Física. Para ello se estudiará un caso concreto: el balance de energías de un objeto que se deja caer desde una altura determinada sobre el suelo. Para entender mejor la experiencia se explica cómo se aplica el balance de energías:

- El objeto que se va a dejar caer tiene una energía total o energía mecánica. E_m [J]
- Esa energía mecánica es la suma de diferentes tipos de energías. En este caso el cuerpo posee dos tipos de energía:
 1. Energía cinética: cuando un cuerpo está en movimiento posee este tipo de energía. Su expresión es:

$$E_c = \frac{1}{2}mv^2$$

Donde m [kg] es la masa del cuerpo y v [m/s] es la velocidad del cuerpo.

2. Energía potencial gravitatoria: energía que posee un cuerpo debido a la interacción gravitatoria con otro objeto, en este caso la Tierra. Su expresión es:

$$E_p = mgh$$

Donde m [kg] es la masa del cuerpo, g [m/s²] es la gravedad en la superficie de la Tierra (9,81 m/s²) y h [m] es la altura a la que se encuentra el objeto respecto al suelo.

- Energía mecánica : $E_m = E_c + E_p$
- En este problema se van a hacer simplificaciones:
 - No aparecen fuerzas no conservativas, es decir, fuerzas como el rozamiento aerodinámico o cualquier otro tipo de fuerza que disipe energía.
- Si no aparecen fuerzas de carácter no conservativo entonces se puede aplicar el siguiente principio, llamado principio de conservación de la energía mecánica:

$$\Delta E_m = 0$$

La variación de la energía mecánica es 0, o dicho de otra forma, el cuerpo siempre mantiene la misma energía mecánica. Por lo tanto si A y B son dos puntos del espacio donde puede estar el cuerpo:

$$E_{m,A} = E_{m,B}$$

Ahora se detallarán los balances de energía en la situación de un objeto que se deja caer desde una altura h (como en la experiencia a diseñar). Se observan tres tipos de estados:

Estado 1

El objeto (Figura 58) se encuentra a una altura h_1 del suelo (sí tiene energía potencial) y en reposo (no tiene energía cinética). Por lo tanto su balance de energías es:

- $E_{c1} = 0$
- $E_{p1} = mgh_1$
- $E_{m1} = E_{c1} + E_{p1} = 0 + mgh_1 = mgh_1$

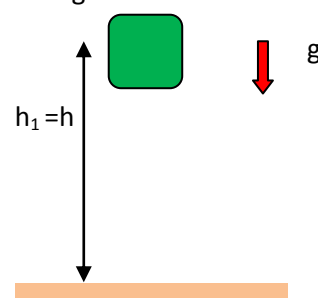


Figura 58: Esquema del Estado 1

Estado 2

El objeto (Figura 59) se encuentra a una altura h_2 del suelo (sí tiene energía potencial) y posee una velocidad v_2 (sí tiene energía cinética). Por lo tanto su balance de energías es:

- $E_{c2} = \frac{1}{2}mv_2^2$
- $E_{p2} = mgh_2$
- $E_{m2} = E_{c2} + E_{p2} = \frac{1}{2}mv_2^2 + mgh_2$

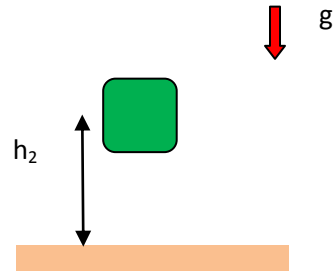


Figura 59: Esquema del Estado 2

Estado 3

El objeto (Figura 60) se encuentra en el suelo (no tiene energía potencial) y posee una velocidad v_3 (sí tiene energía cinética). Por lo tanto su balance de energías es:

- $E_{c3} = \frac{1}{2}mv_3^2$
- $E_{p3} = 0$
- $E_{m3} = E_{c3} + E_{p3} = \frac{1}{2}mv_3^2 + 0 = \frac{1}{2}mv_3^2$

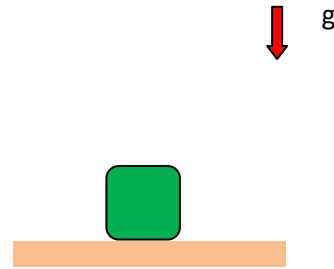


Figura 60: Esquema del Estado 3

Y aplicando el principio de conservación de la energía mecánica se obtiene que:

$$E_{m1} = E_{m2} = E_{m3}$$

Aplicando este balance entre dos posiciones cualesquiera, se puede llegar a conocer la altura h o la velocidad v a la que se encuentra el objeto en cualquier posición.

Diseño

El diseño de la aplicación se realizará acorde con los esquemas presentados en el apartado anterior, es decir, será una animación en 2D de un cuerpo en forma de cuadrado que cae y llega al suelo. En la figura 61 se muestra un esquema del diseño.

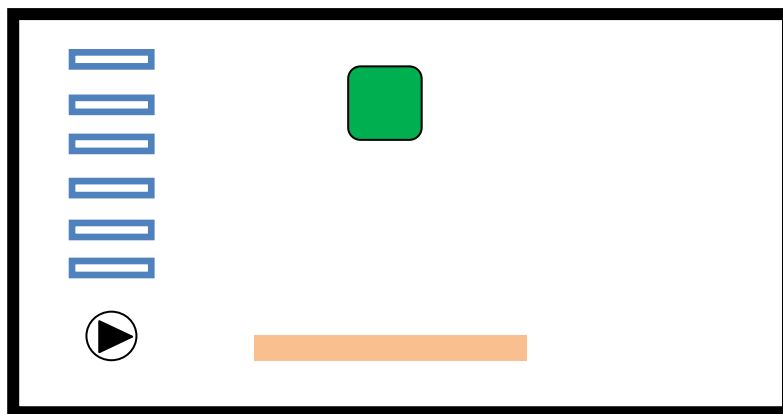


Figura 61: Diseño de la vista por pantalla de Balance de Energías.



El cuerpo está quieto arriba, en la parte inferior el suelo y a la izquierda el botón del Play que al pulsarlo iniciaría la caída del cuadrado. En la parte izquierda también hay una serie de rectángulos que señalan el lugar en el cual irá la siguiente información:

- Energía potencial: habrá un recuadro amarillo que indique la energía potencial en cada instante, tal como este:

E.potencial = 50 J

- Energía cinética: habrá un recuadro azul que indique la energía cinética en cada instante, tal como este:

E.cinética = 50 J

- Energía mecánica: habrá una barra que indicará la energía mecánica total. Esa barra será de dos colores (amarillo y azul) debido los dos tipos de energía (cinética y potencial). La proporción del color indicará la proporción del tipo de energía respecto al total.

Ejemplo:

En un instante determinado mientras el objeto desciende sus energías valen:

$$E_c = 50 J ; E_p = 50 J ; \text{Entonces } E_m = E_c + E_p = 100 J$$

La barra azul representa la Energía Cinética y la barra amarilla representa la Energía Potencial. En este caso las dos barras miden la misma longitud ya que las dos energías valen 50 J. El gráfico permite observar que la dos tienen el mismo peso en el total que es la E_m :

E.mecánica = 100 J

- Masa: la masa del objeto.
- Altura: la altura a la que se encuentra en cada instante.
- Gravedad: la gravedad de la Tierra.

Por último cabe comentar que habrá un botón que permitirá detener la animación en cualquier instante para poder observar en esa posición el valor exacto de las energías.

Reconocimiento y Tracking

El flujo de estados es el mismo compartido por todas las experiencias de la aplicación: target encontrado, target perdido y target no encontrado. Explicado en el apartado [3.3.1. Diseño Experiencia 1: Ciclotrón.](#)

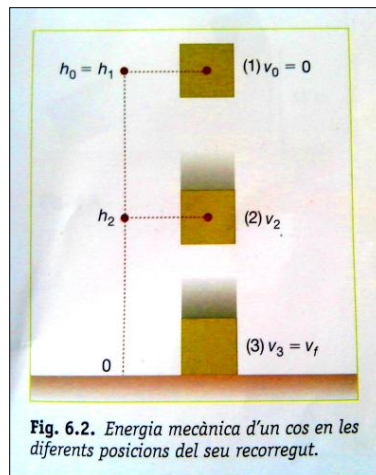


Figura 62: ImageTarget de Balance de Energías.

3.3.4. Diseño Experiencia 4: Sistema Solar

Esta experiencia tiene como objetivo general mejorar los conocimientos que el alumno tiene sobre el Sistema Solar o simplemente ayudar a introducirlo por primera vez. Sus objetivos particulares son:

- Tener una visión general del Sistema Solar y su configuración.
- Observar el orden de los planetas según la distancia al Sol.
- Conocer una escala del tamaño de los planetas.
- Comparar entre ellos los periodos de translación de los planetas (tiempo que tardan en dar una vuelta alrededor del Sol).

Uno de los conceptos teóricos que puede ayudar a la comprensión de esta experiencia es la denominada Tercera Ley De Kepler:

$$T^2 = k \cdot R^3$$

Donde T es el periodo de translación de un planeta, k es una constante y R es la distancia media del planeta al Sol. Lo que se observa es que cuanto más lejos se encuentra un planeta del Sol más tarda en dar una vuelta a su alrededor. De aquí se puede sacar como conclusión que Plutón (el planeta más lejano) tendrá un periodo de translación más grande que el de Mercurio (el planeta más cercano).

Diseño

Se tratará de un diseño animado en 3D del Sistema Solar similar al de la figura 63. Constará de los siguientes elementos:

- Planetas: se incluyen los planetas principales: Mercurio, Venus, Tierra, Marte, Júpiter, Saturno, Urano, Neptuno. También se incluye Plutón, que no es un planeta principal



sino un planeta enano pero que hasta el año 2006 fue considerado el noveno planeta principal.¹⁰¹

- Órbitas: Se representará la órbita de cada planeta por medio de una curva/alambre.
- Sol: al igual que los planetas el Sol dispone de su modelo 3D.
- Botón de Play: para empezar la animación.

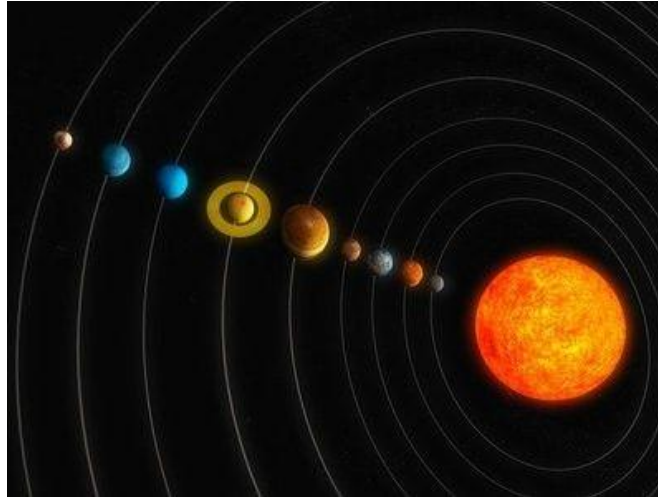


Figura 63: Recreación del Sistema Solar.¹⁰²

La animación incluye los siguientes fenómenos:

- Traslación de los planetas alrededor del Sol.
- Rotación de cada planeta sobre sí mismo.
- Rotación del Sol sobre sí mismo.

También se han hecho una serie de simplificaciones que cabe mencionar, alguna de ellas motivada por las limitaciones del software utilizado para hacer la animación otras porque se han priorizado otros aspectos:

- Algunas órbitas se han aproximado por círculos. En realidad son elípticas pero debido a su excentricidad tan pequeña se pueden aproximar por círculos, la mayoría. Sus valores por tanto no son exactos.
- Las órbitas no están a escala. Esto es debido a que la distancia entre ellas es tan grande que ni los planetas ni el Sol se verían.
- El tamaño del Sol no está a escala. Esto se debe a que ocuparía mucho espacio en relación a los planetas ya que es del orden de 109 veces más grande que la Tierra.
- Las rotaciones de los planetas y del Sol no son reales. Se debe a que el periodo de translación es tan grande en comparación con el de rotación que no se puede introducir el de todos los planetas.
- Se han menospreciado otros muchos aspectos de la dinámica celeste, que por extensión y desconocimiento se han omitido.

Así pues los datos 100 % verídicos son:

¹⁰¹ Fuente: http://es.wikipedia.org/wiki/Sistema_Solar#Planetas

¹⁰² Fuente: http://lh5.ggpht.com/_hbgloOXJrqc/TKTrVJ_Ubl/AAAAAAAAA0YI/NThY12foreg/s640/el-sistema-solar.jpg

- Escala de los planetas entre ellos.
- Escala de los períodos de translación de los planetas entre ellos.
- Ordenación de los planetas.

Reconocimiento y Tracking

El flujo de estados es el mismo compartido por todas las experiencias de la aplicación: target encontrado, target perdido y target no encontrado. Explicado en el apartado [3.3.1. Diseño Experiencia 1: Ciclotrón.](#)

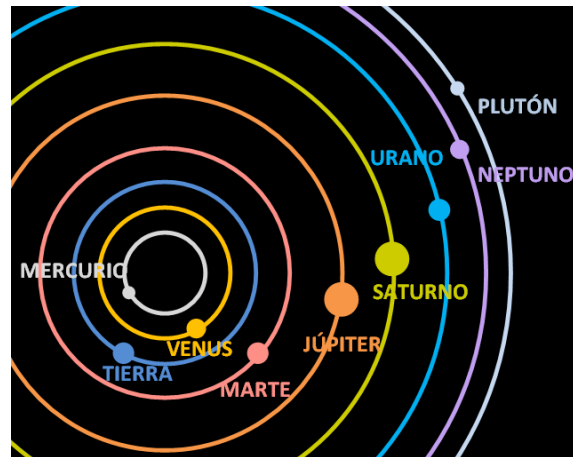


Figura 64: ImageTarget de Sistema Solar.

3.4. Desarrollo de la aplicación

3.4.1. Experiencia 1: Ciclotrón

Modelado

Inicialmente se procedió a modelar en 3D los elementos de la animación utilizando el programa de gráficos 3D Blender. Los elementos a modelar eran:

- Ciclotrón.

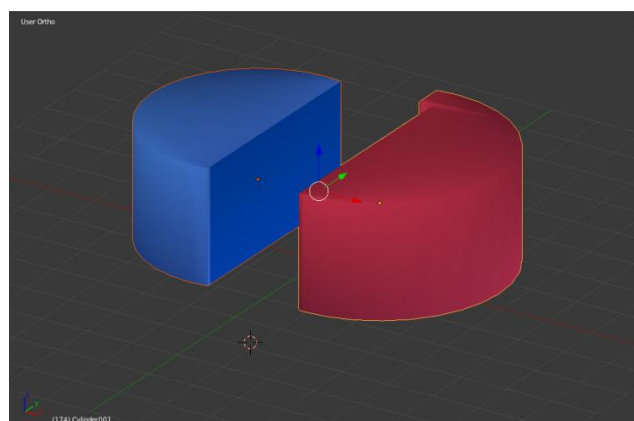


Figura 65: Modelo en 3D del ciclotrón



- Flechas del campo eléctrico.

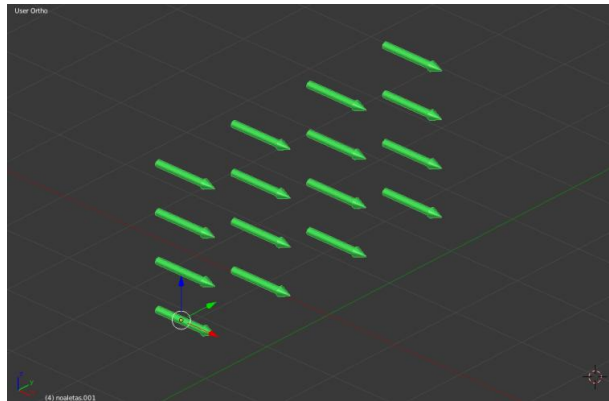


Figura 66: Modelo en 3D del campo eléctrico

- Flechas de campo magnético.

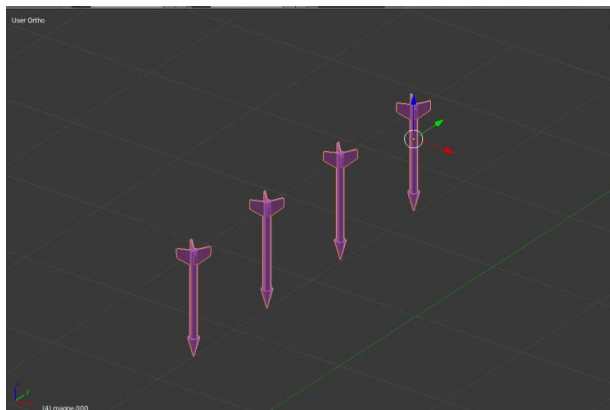


Figura 67: Modelo en 3D del campo magnético.

- Partícula cargada y trayectoria.
Se modeló como una esfera y un alambre.



Figura 68: Modelo en 3D de la partícula y su trayectoria.

Siendo el resultado final el siguiente:

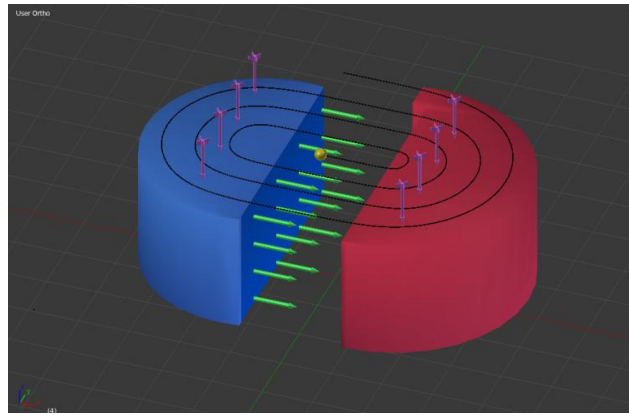


Figura 69: Modelo en 3D de todo el conjunto del ciclotrón.

Animación

Una vez realizado el modelado se procedió a realizar la animación de las flechas de campo eléctrico y las flechas de campo magnético, así como el movimiento de la esfera a lo largo del alambre. Todo esto con ayuda del Blender.

Implementación y Creación de la Aplicación

Una vez acabada la animación se procedió a importarla a Unity donde se creó posteriormente la aplicación. A la animación se le adjuntaron scripts o páginas de código que permitieron implementar las siguientes funcionalidades:

- **Flujo de ejecución o de estados**

Se implementó mediante código una clase que permitiera establecer los tres estados posibles explicados previamente en el apartado [3.3.1. Diseño Experiencia 1: Ciclotrón](#). Los tres estados son: target no encontrado, target encontrado y target perdido. Aquí se muestra el resultado final:

Target encontrado

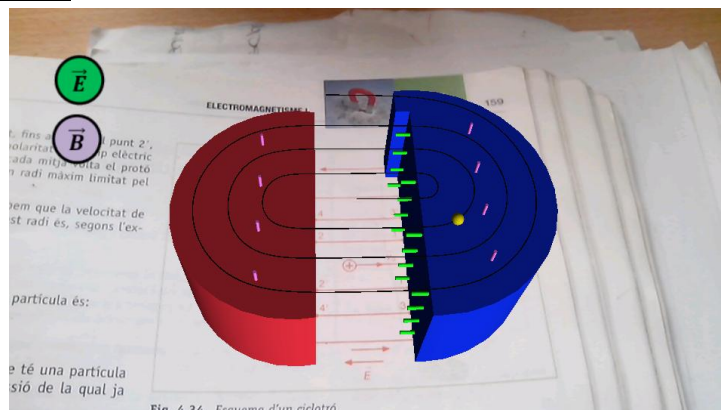


Figura 70: Ciclotrón. Target encontrado.

Estado activo cuando se detecta el target. El código manda a renderizar todo el contenido.



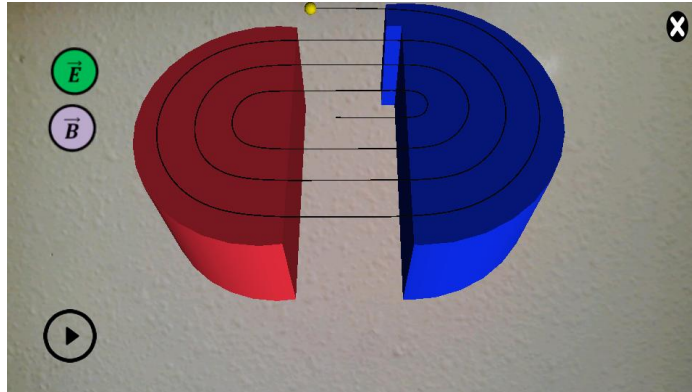
Target perdido

Figura 71:Ciclotrón.Target perdido.

Estado activo cuando se pasa de detectar a no detectar el target, por lo tanto, le precede el estado "Target encontrado". Lo que hace es coger el objeto y lo coloca en unas coordenadas y con una orientación determinada para que se vea entera la figura en la pantalla. Durante este estado se activa otra clase que permite escalar el objeto para que se ajuste a las dimensiones de la pantalla.

Target no encontrado

Estado activo en dos ocasiones: cuando se arranca la aplicación y aún no se ha detectado el target o bien cuando estando en el estado *Target perdido* el usuario pulsa el botón de cerrar. Su función es desactivar el renderizado de todos los objetos, es decir, los hace desaparecer.



Figura 72:Ciclotrón.Target no encontrado.

- **Auto escalado**

Cuando la animación pasa al estado de *Target perdido* el objeto se adapta de forma automática al tamaño de la pantalla. Para realizar esta función se creó un script que hacía básicamente lo siguiente:

1. Mira el estado de la aplicación y si es *Target perdido* ejecuta el código que hay a continuación.
2. Comprueba que no se haya escalado anteriormente. Si es así continua.
3. Una vez pasados los 2 primeros pasos, lo primero que hace es mirar las coordenadas de los contornos de cada componente o elemento de la animación. Se pasan esas coordenadas absolutas a coordenadas de la pantalla.
4. Compara esas coordenadas y da una estimación de los límites del conjunto.

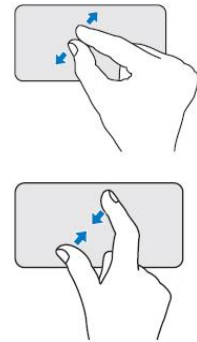
5. Se comparan las dimensiones de la pantalla con las dimensiones del conjunto de la animación.
6. Se determina el escalado a realizar.

Cabe mencionar que también incorpora un método que permite guardar la escala original de la animación y restaurarla cuando se pasa a *Target encontrado* o *Target no encontrado*.

- **Escalado**

La aplicación además del escalado automático, incorpora un escalado manual o zoom que queda activo en el estado *Target encontrado*. Lo único que debe hacer el usuario es tener dos dedos tocando la pantalla.

1. Zoom (+): deslizar los dedos sobre la pantalla separándolos entre sí.
2. Zoom (-): deslizar los dedos sobre la pantalla juntándolos entre sí.



En la figura 73 (Fuente¹⁰³) se detalla el proceso.

Figura 73: Zoom

- **Reproducción o Botón de Play**



En la parte inferior izquierda de la pantalla se encuentra el botón Play, que puede estar presente en los estados “Target perdido” y “Target encontrado”, sin embargo sólo es visible cuando la animación ha acabado.

Para realizar estas acciones hace lo siguiente:

1. Comprueba que la animación no se esté ejecutando y que el estado de la aplicación sea Target perdido” o “Target encontrado”.
2. Si se cumple 1. entonces muestra el botón.
3. Al pulsarse el botón se inicia la animación, por lo que el paso 1 no es superado ahora y por lo tanto no se llega al paso 2. El botón desaparece.

- **Botón de cerrado**



La aplicación incorpora un botón de cerrado de la animación en la esquina superior derecha (Figura 74). Sólo se activa en el estado “Target perdido”. Las órdenes de ejecución del código que lo implementa son:

1. Mira si el estado actual es “Target perdido”.
2. En caso afirmativo, muestra el botón de cerrar.
3. Si se pulsa, cambia el estado a “Target no encontrado”

¹⁰³Fuente:

http://www.techmonsters.com/DellTraining/bin/Foundation2010/Portables_Foundation_2010/Latitude/Z600/Z600/images/tp_mult_zoom.jpg



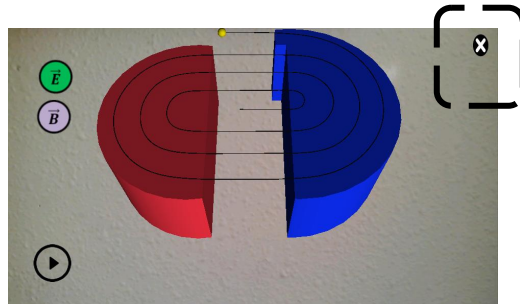
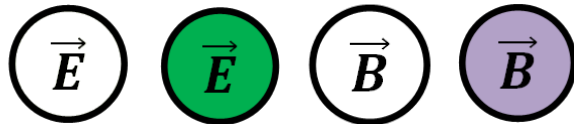


Figura 74: Botón de cerrado.

- **Activación/Desactivación de campos**



La aplicación incorpora dos botones en la parte izquierda de la pantalla (Figura 75). Esos botones fueron creados con ayuda de la herramienta GIMP. El botón verde con una E dentro simboliza el campo eléctrico que en la animación es del mismo color y el botón lila con una B dentro simboliza el campo magnético. Si alguno de ellos o los dos botones están coloreados y no en blanco, entonces se renderizan las flechas verdes y lilas. Si por el contrario se desmarca una, entonces ese campo se deja de mostrar. Esta funcionalidad está presente en dos estados: “Target perdido” y “Target encontrado”.

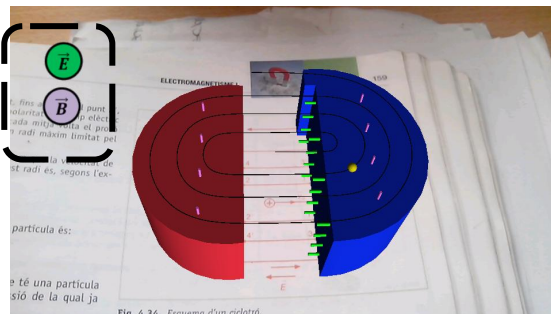


Figura 75: Botones de activación/desactivación del campo eléctrico y magnético.

3.4.2. Experiencia 2: Rampa con rozamiento

Modelado

Inicialmente se procedió a modelar en 3D los elementos de la animación, utilizando el programa de gráficos 3D Blender. Los elementos a modelar eran:

- Cubo y Rampa

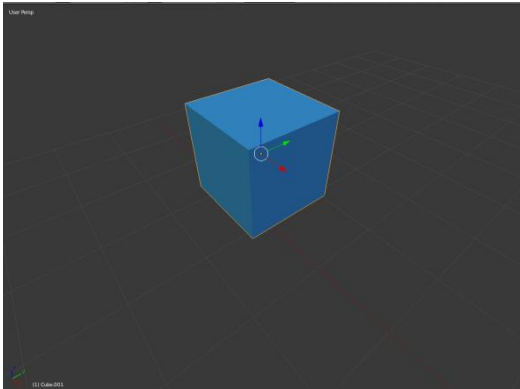


Figura 76: Cubo

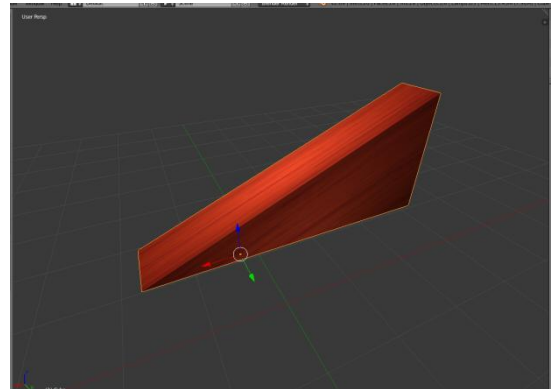


Figura 77: Rampa

Siendo el resultado final (visto en Unity):

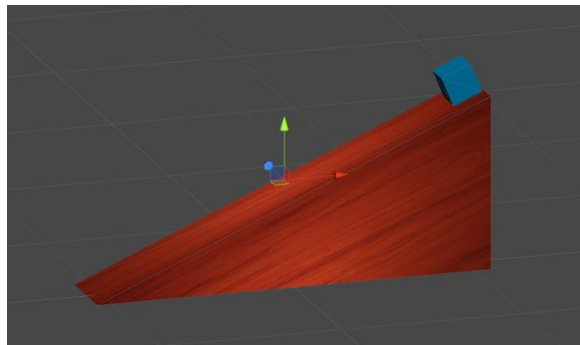


Figura 78: Cubo + rampa

Animación

Una vez realizado el modelado de los objetos se exportó a Unity donde se realizaría la animación del cubo, que consistía en que bajara deslizándose por la rampa. La animación la realizaba el método de una clase:

- Fijaba la posición del cubo en cada frame, escribiendo las ecuaciones del movimiento del cubo en las tres coordenadas del espacio tridimensional (X, Y, Z) las cuáles iban variando en función del tiempo de ejecución de la aplicación, que es proporcionado por un método en Unity.

Implementación y Creación de la Aplicación

A la animación se le adjuntaron más scripts o páginas de código que permitieron implementar las siguientes funcionalidades:

- **Flujo de ejecución o de estados**
Se implementó mediante código una clase que permitiera establecer los tres estados posibles explicados previamente en el apartado [3.3.1. Diseño Experiencia 1: Ciclotrón.](#) y [3.4.1. Experiencia 1: Ciclotrón.](#) Los tres estados son: target no



encontrado, target encontrado y target perdido. Aquí se muestra el resultado final de los tres estados:

Target encontrado

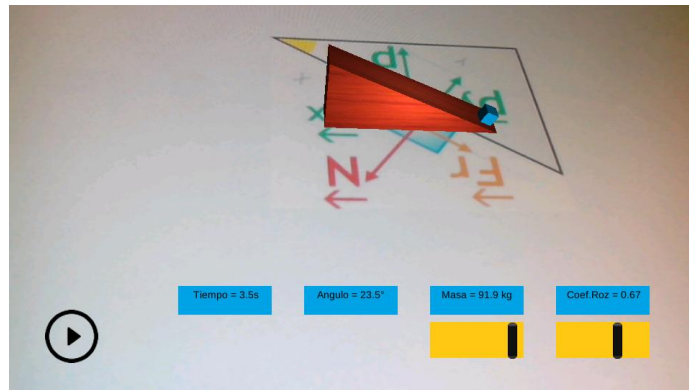


Figura 79: Rampa con rozamiento. Target encontrado.

Target perdido

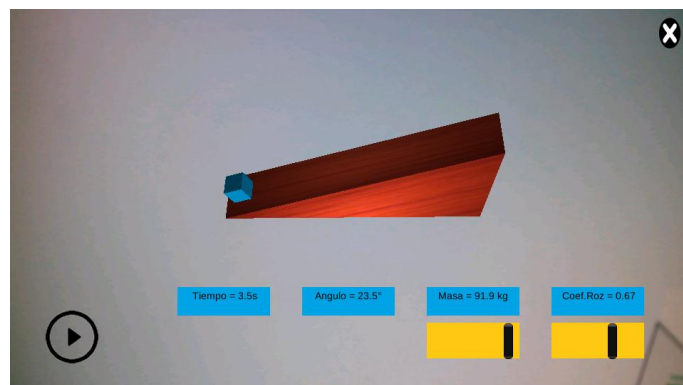


Figura 80: Rampa con rozamiento. Target perdido.

Target no encontrado: no se muestra nada.

- **Auto escalado**
Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)
- **Escalado**
Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)
- **Reproducción o Botón de Play**
Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)
- **Botón de cerrado**
Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)
- **Elementos visuales en pantalla**
Además de los botones de Play y de cerrado, se pueden ver otros elementos gráficos en la pantalla:

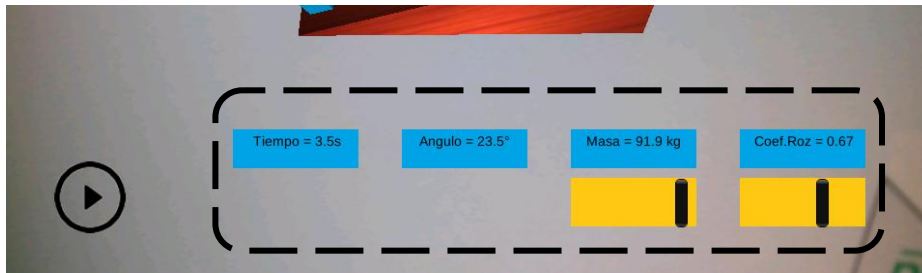
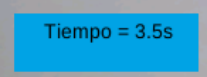
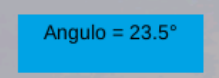




Figura 81: Rampa con rozamiento. Elementos visuales en pantalla.

1. **Tiempo:** marca el tiempo de caída del objeto. Para hacer esto se implementó una función que hace lo siguiente:
 1. Define dos variables: un tiempo inicial y un tiempo actual. Ambos tiempos los proporciona Unity, que da el Tiempo desde el inicio de la aplicación.
 2. Esto hace que el tiempo no sea el tiempo actual, ya que ese tiempo es el tiempo de duración de la aplicación.
 3. Por lo tanto se define: $\text{Tiempo} = \text{Tiempo actual} - \text{Tiempo inicial}$.
 4. Al iniciarla animación pulsando al Play, en ese instante:
 $\text{Tiempo inicial} = \text{Tiempo actual}$
 5. De esta forma el tiempo que se proporciona es correcto.
- 
2. **Ángulo:** proporciona el ángulo de la rampa que es constante.


 3. **Masa:** informa sobre la masa del cubo que el usuario ha fijado mediante la barra vertical negra que se desliza sobre el fondo amarillo. Esta función está implementada por Unity.


 4. **Coef.Roz:** proporciona el coeficiente de rozamiento que el usuario ha fijado mediante la barra vertical negra que se desliza sobre fondo amarillo. Esta función está implementada por Unity.



Además de variar el valor que se muestra por pantalla, el coeficiente de rozamiento es enviado a la función que se encarga de fijar la trayectoria del cubo ya que el rozamiento es un valor necesario en ese cálculo.

3.4.3. Experiencia 3: Balance de energías

Modelado

Inicialmente se procedió a modelar en 2D los elementos de la animación, utilizando el programa de edición de imagen Gimp. Los elementos a modelar eran:



- Cuadrado(cuerpo) y Suelo



Figura 82: Elementos. Balance de energías.

Siendo el resultado final (visto en Unity):

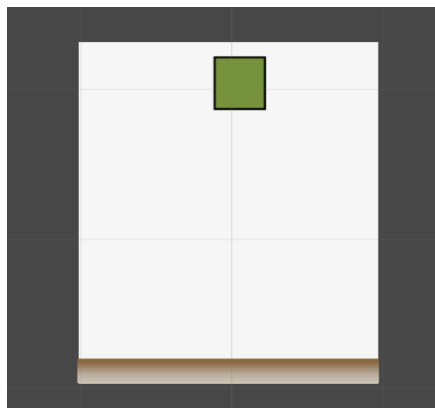


Figura 83: Resultado gráfico del conjunto: cuadrado + suelo. Balance de energías.

Animación

Una vez realizado el modelado de los objetos se exportaron a Unity donde se realizaría la animación del cuadrado que consistía en que cayera llegando al suelo. La animación consistía en un método que:

- Fijaba la posición del cubo en cada frame, escribiendo las ecuaciones del movimiento del cubo en las tres coordenadas del espacio tridimensional (X, Y, Z) las cuáles iban variando en función del tiempo de ejecución de la aplicación, que es proporcionado por un método en Unity.
- El tiempo con el que se calculaba la posición iba determinado por una función que hace:
 1. Define dos variables: un tiempo inicial y un tiempo actual. Ambos tiempos los proporciona Unity, que da el Tiempo desde el inicio de la aplicación.
 2. Esto hace que el tiempo no sea el tiempo actual, ya que ese tiempo es el tiempo de duración de la aplicación.
 3. Por lo tanto se define: $\text{Tiempo} = \text{Tiempo actual} - \text{Tiempo inicial}$.
 4. Al iniciar la animación pulsando al Play o al reiniciarla después de pulsar Pause, en ese instante:

$$\text{Tiempo inicial} = \text{Tiempo actual}$$
 5. De esta forma el tiempo que se proporciona es correcto.

Implementación y Creación de la Aplicación

A la animación se le adjuntaron más scripts o páginas de código que permitieron implementar las siguientes funcionalidades:

- **Flujo de ejecución o de estados**

Se implementó mediante código una clase que permitiera establecer los tres estados posibles explicados previamente en el apartado [3.3.1. Diseño Experiencia 1: Ciclotrón](#) y [3.4.1. Experiencia 1: Ciclotrón](#). Los tres estados son: target no encontrado, target encontrado y target perdido. Aquí se muestra el resultado final:

Target encontrado

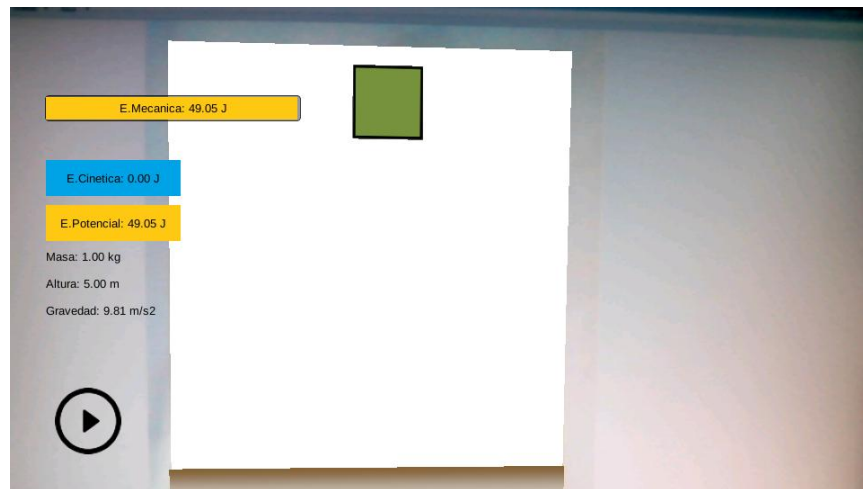


Figura 84: Balance de energías. Target encontrado.

Target perdido

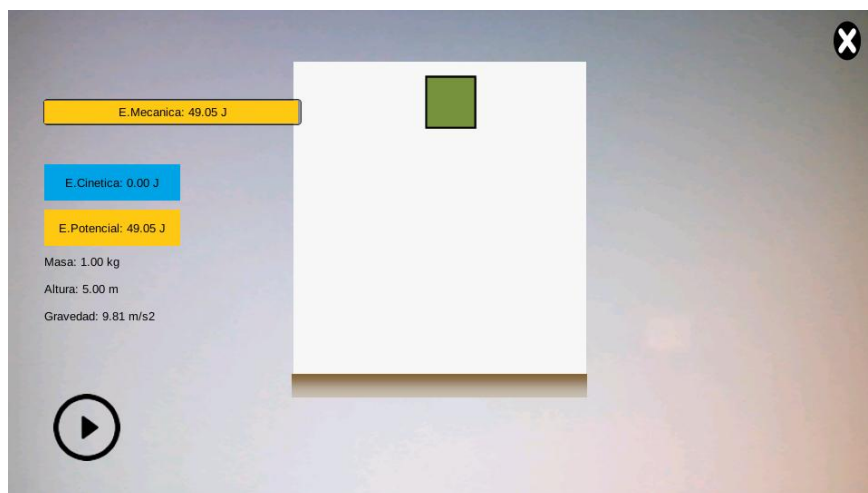


Figura 85: Balance de energías. Target perdido.

Target no encontrado: no se muestra nada.

- **Auto escalado**

Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón](#).

- **Escalado**



Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)

- **Reproducción o Botón de Play**

Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)

- **Botón de Pause**



En la parte inferior izquierda de la pantalla se encuentra el botón Pause, que puede estar presente en los estados “Target perdido” y “Target encontrado”, sin embargo sólo es visible cuando se ha iniciado la animación al pulsar Play. Para la animación y entonces se muestra de nuevo el botón Play. Hace lo siguiente:

1. Comprueba que la animación aplicación sea “Target perdido” o “Target encontrado”.
2. Si se cumple y alguien pulsa el botón Pause entonces muestra el botón.
3. Desaparece el botón de Pause y aparece el botón de Play en su lugar.
4. Fija valores de tiempo y velocidad, para que al reanudar la animación las ecuaciones del movimiento funcionen como se desea.

- **Botón de Replay**



En la parte inferior izquierda de la pantalla se encuentra el botón Replay, que puede estar presente en los estados “Target perdido” y “Target encontrado”, sin embargo sólo es visible cuando se ha terminado la animación, es decir, cuando el cuadrado llega al suelo. Realiza las siguientes acciones cuando se pulsa:

1. Desaparece el botón y hace aparecer el de Play.
2. Devuelve el objeto o cuadrado a su posición de origen.
3. Reinicia los valores de posición inicial y de velocidad. Al hacerlo se reinician automáticamente los valores de la Energía Cinética y la Energía Potencial.

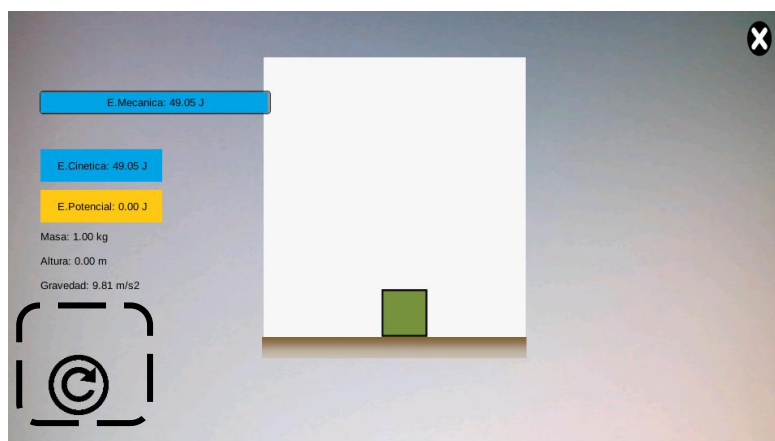


Figura 87: Botón de Replay. Balance de energías.

- **Botón de cerrado**

Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)

- **Elementos visuales en pantalla**

Además de los botones de Play, Pause, Replay y cerrado se pueden ver otros elementos gráficos en la pantalla:

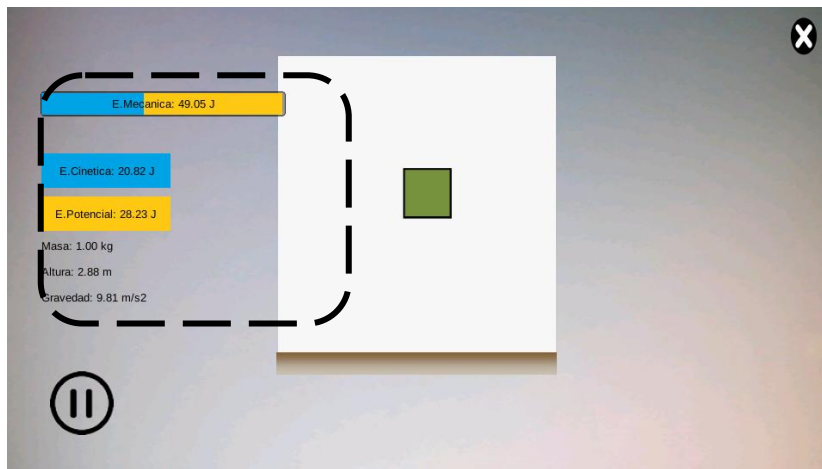


Figura 88: Elementos visuales en pantalla. Balance de energías.

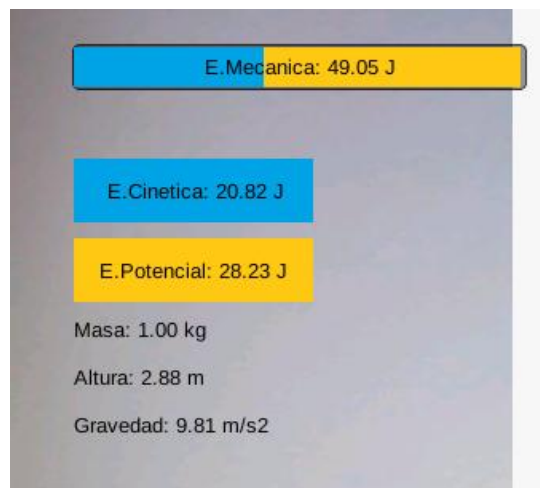
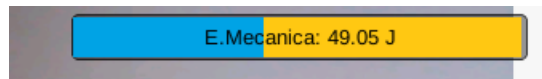


Figura 89: Elementos visuales en pantalla. Balance de energías.

1. Masa: es un valor que viene por defecto.
2. Altura: valor que viene por defecto.
3. Gravedad: valor que viene por defecto.
4. E.mecánica: valor que proporciona la energía mecánica en cada momento y a la vez muestra en colores el “peso” de cada una de las energías en su valor.



Para ello se realiza lo siguiente:

1. Se calcula la energía mecánica con el dato de la altura inicial.
2. Se dibuja la barra azul (E.cinética) y la barra amarilla (E.potencial) mirando el % de su valor respecto al total de la Energía mecánica.
3. Se representan las barras.
5. E.cinética: valor que proporciona la energía cinética en cada momento. Para ello se realiza lo siguiente:



1. Se calcula la posición del cuerpo en cada instante. Esa posición se encuentra en coordenadas del espacio de la aplicación, así que hay que establecer una relación la altura en metros
 2. Se pasa la posición/altura a metros.
 3. Se realiza el cálculo de la energía cinética, restando la energía potencial a la energía mecánica.
6. E.potencial: valor que proporciona la energía potencial en cada momento. Para ello se realiza lo siguiente:
1. Se calcula la posición del cuerpo en cada instante. Esa posición se encuentra en coordenadas del espacio de la aplicación, así que hay que establecer una relación la altura en metros
 2. Se pasa la posición/altura a metros.
 3. Se realiza el cálculo de la energía potencial con la altura en cada instante.

3.4.4. Experiencia 4: Sistema Solar

Modelado

Inicialmente se procedió a modelar en 3D los elementos de la animación, utilizando el programa de gráficos 3D Blender. Los elementos a modelar eran:

- 9 planetas

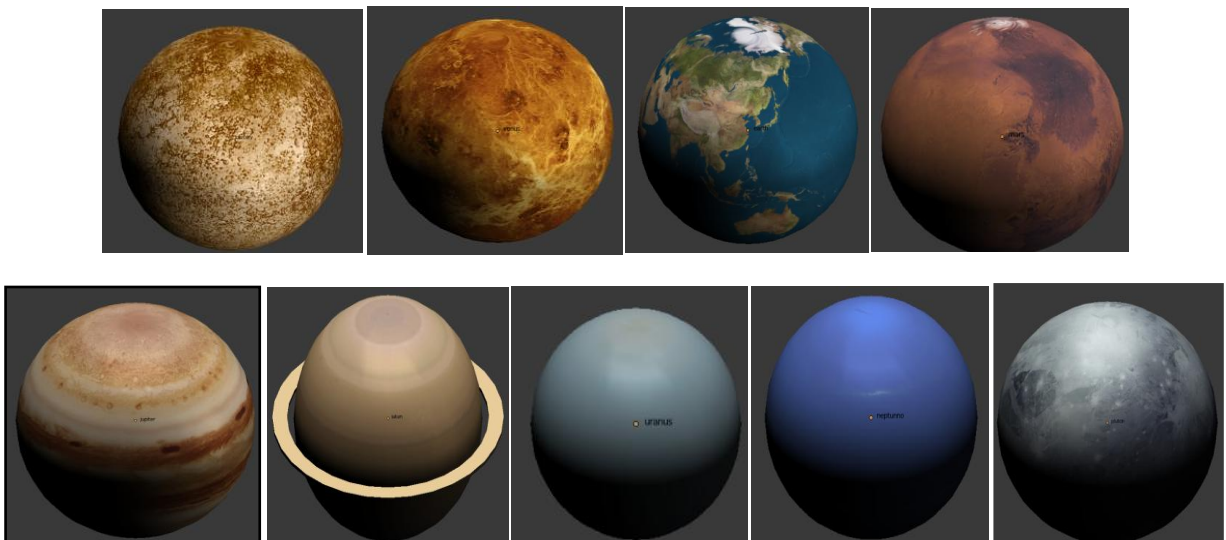


Figura 90: Modelos en 3D de los planetas. Sistema Solar.

- El Sol



Figura 91: Modelo en 3D del Sol. Sistema Solar.

- Órbitas de los planetas

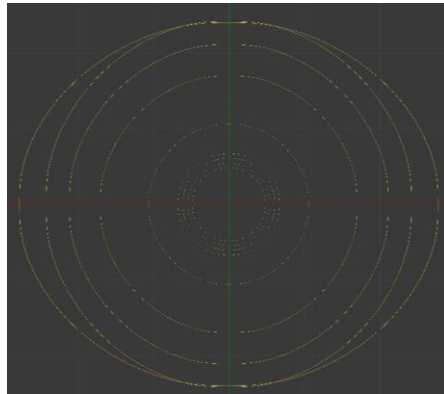


Figura 92: Modelos en 3D de las órbitas planetarias. Sistema Solar.

El resultado final es:

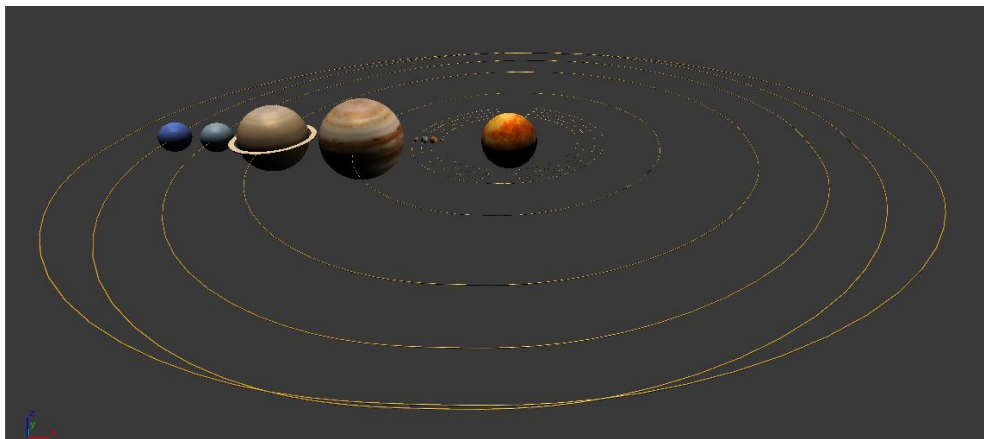


Figura 93: Modelos en 3D del Sistema Solar.

Animación

Una vez construidos el Sistema Solar se procedió a animarlo. En concreto se hicieron dos animaciones:

- **Traslación:** cada planeta se hizo que girara alrededor del Sol. El tiempo que tarda cada planeta se hizo de tal forma que el planeta que tarda menos tiempo (Mercurio) diese una vuelta entera en pocos segundos y todos los demás se hicieron por comparación con Mercurio.



- **Rotación:** dado que los periodos de traslación tardan pocos segundos, algunos periodos de rotación debían tardar centésimas de segundo o milésimas y dado que con Blender menos de 1 frame de animación no se puede hacer, se optó por darles a todos un mismo periodo de rotación que fuese apreciable.

Implementación y Creación de la Aplicación

A la animación se le adjuntaron más scripts o páginas de código que permitieron implementar las siguientes funcionalidades:

- **Flujo de ejecución o de estados**
Se implementó mediante código una clase que permitiera establecer los tres estados posibles explicados previamente en el apartado [3.3.1. Diseño Experiencia 1: Ciclotrón](#) y [3.4.1. Experiencia 1: Ciclotrón](#). Los tres estados son: target no encontrado, target encontrado y target perdido. Aquí se muestra el resultado final:

Target encontrado

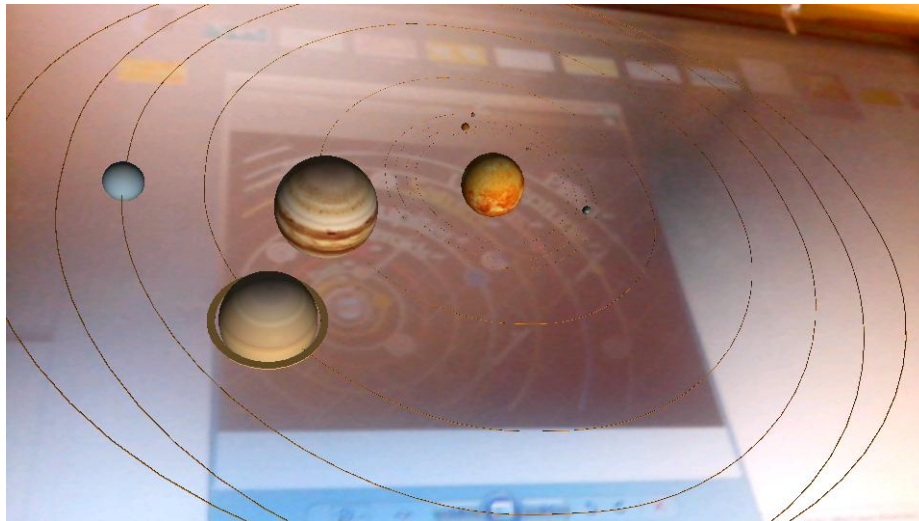


Figura 94: Sistema Solar. Target encontrado.

Target perdido

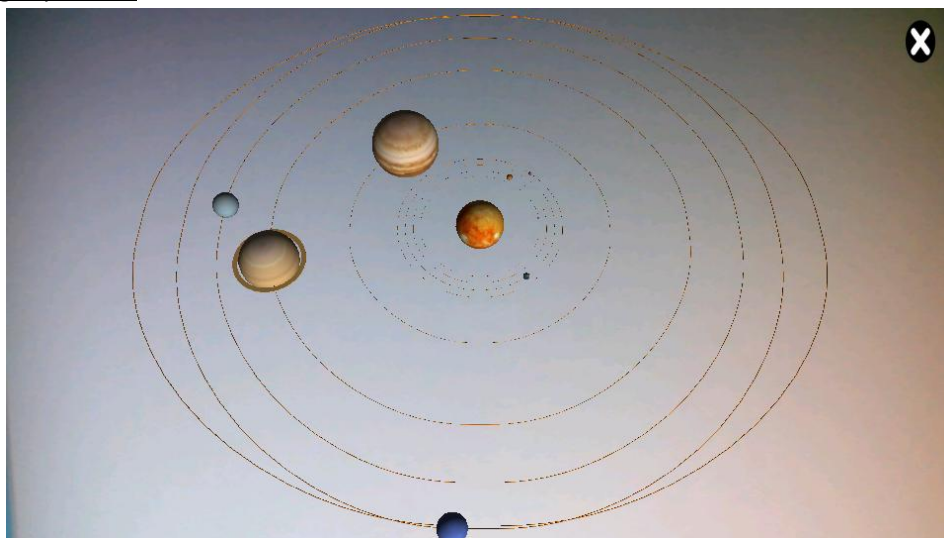


Figura 95: Sistema Solar. Target perdido.

Target no encontrado: no se muestra nada.

- **Auto escalado**
Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)
- **Escalado**
Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)
- **Reproducción o Botón de Play**
Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)
- **Botón de cerrado**
Explicado en el apartado [3.4.1. Experiencia 1: Ciclotrón.](#)



4. Impacto ambiental y social

4.1. Impacto ambiental del desarrollo de la aplicación

El proyecto presentado se ha basado enteramente en el desarrollo de software y diseño por ordenador, por lo tanto su mayor impacto ambiental ha estado asociado al uso de los recursos energéticos (electricidad) empleados para el funcionamiento de los equipos informáticos. No se tiene en cuenta ni el tiempo ni los recursos destinados a la redacción de la memoria, ya que no se los considera impacto ambiental.

El consumo de electricidad conlleva un impacto ambiental indirecto derivado de la producción de esa misma electricidad que se consume

Se supone un consumo medio del ordenador y la pantalla en modo encendido de 0,065 kW¹⁰⁴ y el tiempo de desarrollo del proyecto en ordenador es de 480 horas sin contar la redacción del trabajo (ver Tabla 4). A continuación se presentan en las Tablas 5 y 6 las emisiones atmosféricas y los residuos radiactivos asociados a la producción de la energía consumida utilizada durante el desarrollo del proyecto.

Tiempo de desarrollo	Potencia consumida	Consumo eléctrico
480 h	0,065 kW	31,2 kWh

Tabla 4: Datos sobre el consumo eléctrico durante el desarrollo.

Tipos de emisiones atmosféricas	Factor	Consumo eléctrico	Emisiones atmosféricas = Factor · Consumo
Dióxido de carbono (CO ₂)	0,179 kg/kWh	31,2 kWh	5,585 kg
Dióxido de azufre (SO ₂)	0,417 g/kWh	31,2 kWh	13,010 kg
Óxidos de nitrógeno (NO _x)	0,283 g/kWh	31,2 kWh	8,823 kg

Tabla 5: Emisiones contaminantes derivadas del consumo eléctrico realizado.¹⁰⁵

¹⁰⁴ Fuente: <http://www.leantricity.es/es/2012/07/11/cuanta-energia-gasta-un-ordenador-aproximaciones/>

¹⁰⁵ Fuente: WWF España. Observatorio de la Electricidad Mayo 2014[en línea]. Raquel García Monzón. Madrid,2014.[Consulta: 20 de enero 2014] Disponible en: http://awsassets.wwf.es/downloads/oe_mayo_02062014.pdf

Tipos de residuos radiactivos	Factor	Consumo	Residuos radiactivos = Factor · Consumo
Baja y media actividad	0,00234 cm ³ /kWh	31,2 kWh	0,07301 cm ³
Alta actividad	0,286 mg/kW	31,2 kWh	8,923 mg

Tabla 6: Residuos radiactivos derivadas del consumo eléctrico realizado.¹⁰⁶

4.2. Impacto ambiental de la aplicación

El impacto ambiental que puede ocasionar la aplicación es la utilización de recursos naturales como el papel: el software no necesita de recursos como el papel de los libros, sino de dispositivos de representación gráfica como pantallas que ya llevan incorporados los móviles (de uso muy extendido). Sin embargo a la hora de representar los targets normalmente se utiliza papel, aunque no resulta imprescindible.

4.3. Impacto social de la aplicación

El impacto social que puede ocasionar la aplicación es la mejora de las técnicas de aprendizaje: es el mayor impacto de la aplicación, el impacto que puede provocar en la sociedad y su cultura al mejorar los sistemas de enseñanza en la educación impartida en colegios, institutos y universidades. Las mejoras en el campo de la educación, conllevan mejoras en cualquier aspecto de una civilización o sociedad.

¹⁰⁶ Fuente: WWF España. Observatorio de la Electricidad Mayo 2014[en línea]. Raquel García Monzón. Madrid,2014.[Consulta: 20 de enero 2014] Disponible en:
http://awsassets.wwf.es/downloads/oe_mayo_02062014.pdf



5. Planificación y valoración económica

En este apartado se detalla la planificación y el coste del proyecto.

5.1. Planificación

La planificación del proyecto se dividió en diversas etapas que se organizaron según el diagrama de Gantt de la Figura 96 y que se detallan en la Tabla 7.

Etapa del proyecto	Descripción
Seminario sobre Realidad Aumentada	Asistencia a un seminario de introducción en el desarrollo de aplicaciones de RA con Unity 3D y Vuforia.
Formación en Unity	Cursos online, tutoriales, vídeos, manuales de referencia, foros, etc.
Formación en Blender	Curso presencial en la universidad, recursos online.
Formación en Vuforia	Seminario de RA, guía de desarrolladores de Vuforia.
Primeros ejemplos de desarrollo en Realidad Aumentada	Primeras aplicaciones para móvil utilizando Unity + Vuforia.
Decisión sobre el tema de la aplicación/Brainstorming	Decisión sobre qué aplicación de RA se iba a desarrollar.
Diseño de la experiencia 1	Planteamiento y diseño de la experiencia 1, el ciclón.
Desarrollo e implementación de la experiencia 1	Creación de elementos con Blender, implementación en Unity y desarrollo de funcionalidades.
Diseño de la experiencia 2	Planteamiento y diseño de la experiencia 1, la rampa con rozamiento.
Desarrollo e implementación de la experiencia 2	Creación de elementos con Blender, implementación en Unity y desarrollo de funcionalidades.
Diseño de la experiencia 3	Planteamiento y diseño de la experiencia 1, el balance de energías.
Desarrollo e implementación de la experiencia 3	Creación de elementos con Photoshop, implementación en Unity y desarrollo de funcionalidades.
Diseño de la experiencia 4	Planteamiento y diseño de la experiencia 1, el sistema solar.
Desarrollo e implementación de la experiencia 4	Creación de elementos con Blender, implementación en Unity y desarrollo de funcionalidades.
Implementación de la aplicación	Unión de todas las experiencias en una sola aplicación, optimización del código.
Redacción de la memoria	Escrito de toda la información recogida sobre la RA y sobre el desarrollo de la aplicación.

Tabla 7: Etapas del proyecto

Planificación del proyecto

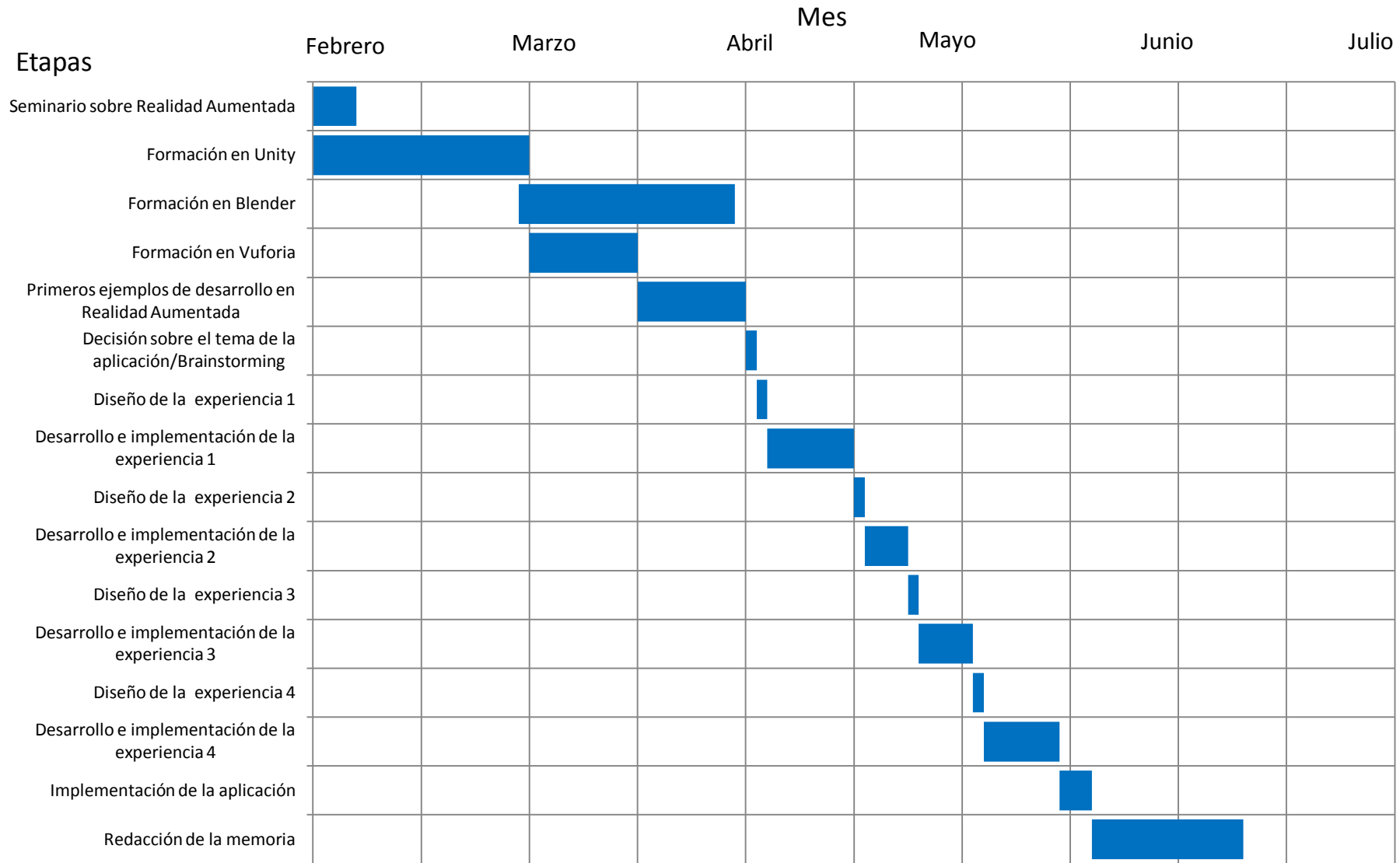


Figura 96: Diagrama de Gantt del proyecto.

5.2. Valoración económica

5.2.1. Costes de recursos humanos

En él se detallan los costes asociados a la mano de obra del ingeniero. En este caso se ha considerado que la realización del trabajo es llevada a cabo por un ingeniero industrial con un salario de 40€/hora. En la Tabla 8 se puede ver el coste de cada etapa y el coste total que asciende a unos 22.000 € sólo en recursos de capital humano

Etapa del proyecto	Tiempo dedicado (h)	Precio (€/hora)	Coste (€)
Seminario sobre Realidad Aumentada	20	40	800
Formación en Unity	100	40	4.000
Formación en Blender	100	40	4.000
Formación en Vuforia	50	40	2.000
Primeros ejemplos de desarrollo en Realidad Aumentada	50	40	2.000
Decisión sobre el tema de la aplicación/Brainstorming	5	40	200
Diseño de la experiencia 1	5	40	200
Desarrollo e implementación de la experiencia 1	40	40	1.600
Diseño de la experiencia 2	5	40	200
Desarrollo e implementación de la experiencia 2	20	40	800
Diseño de la experiencia 3	5	40	200
Desarrollo e implementación de la experiencia 3	25	40	1.000
Diseño de la experiencia 4	5	40	200
Desarrollo e implementación de la experiencia 4	35	40	1.400
Implementación de la aplicación	15	40	600
Redacción de la memoria	70	40	2800
Total	550 h	40 €/hora	22.000 €

Tabla 8: Etapas del proyecto, horas dedicadas a cada etapa y costes de cada etapa.

5.2.2. Costes de material y software

Los costes asociados al material necesario para el desarrollo de la aplicación han sido cero, ya que ya se disponía de un ordenador y teléfono móvil Android y las licencias de los programas utilizados eran gratuitos. Sin embargo se va a presentar el coste que conllevaría si no se dispusiera de ningún tipo de dispositivo (ver Tabla 9).

Material	Coste
Ordenador	600 €
Teléfono móvil Android	200 €
Licencia de Unity	0 €
Licencia de Vuforia	0 €
Licencia de Blender	0 €
Google Play	18,39 € = 25\$
Total	818,39 €

Tabla 9: Datos sobre los costes de material y software.

El coste orientativo del ordenador es de unos 600 € ya sea de sobremesa o portátil, el cual se podría amortizar en un periodo de unos 4 años por lo que el coste sería de 150€/año. El teléfono móvil el cual ha de disponer de cámara de video, está estimado en unos 200 € que se podrían amortizar en un periodo de 2 años a razón de 100€/año.

Con respecto al software utilizado se ha utilizado la versión gratuita de Unity así como la versión gratuita de Vuforia, aunque ambos disponen de versiones de pago con más funcionalidades. Blender, sin embargo, es un software con una única licencia gratuita.

Si se quiere publicar la aplicación en la tienda de aplicaciones de Android, el coste es de 25\$.

5.2.3. Otros costes

Finalmente destacar la existencia de otros costes asociados a la formación, al marketing o al acceso a Internet. No se ha invertido en formación ya que los recursos utilizados han sido tutoriales y cursos gratuitos, pero éstos podrían comportar un coste a tener en cuenta:

- Curso de iniciación al desarrollo de aplicaciones de Realidad Aumentada.
- Curso de Unity.
- Curso de Blender.
- Curso de programación en Boo, Java Script o C#.

Los costes de marketing y publicidad no se han desarrollado porque el objetivo del proyecto no ha sido vender un producto sino crear un prototipo de producto.

También hay que considerar la necesidad de poseer una conexión a Internet con el fin de crear los Targets en la página de Vuforia, así como para consultar cualquier documentación de los softwares utilizados.



Concepto	Coste por mes	Meses	Coste
Formación	-----	-----	0 €
Publicidad	-----	-----	0 €
Conexión a Internet	50 €/mes	5	250 €
Total	-----	-----	250 €

Tabla 10: Datos sobre los costes de publicidad, Internet y formación.

5.2.4. Coste total

Si se suman los costes de recursos humanos, de material y software y los otros costes considerados se obtiene un coste total de 23068,39 €. Se puede ver detallado en la Tabla 11.

Tipo de costes	Coste
Recursos humanos	22.000,00 €
Material y software	818,39 €
Otros	250,00 €
Total	23.068,39 €

Tabla 11: Datos sobre los costes totales del proyecto.

6. Plan económico del proyecto

En el caso de que se quisiera comercializar este producto sería necesario un plan con el objetivo de obtener beneficios. Dado que el carácter de este proyecto no es ofrecer un producto comercial, los cálculos hechos a continuación son orientativos y se realizarán todos desde el punto de vista del autor. Por lo tanto se omiten los costes de recursos humanos así como los costes de material (ya que se dispone de ellos) y ni siquiera se consideran los costes asociados a Internet. Se supone además que ya se dispone de una cuenta en Google Play (el gasto es pequeño).

Objetivo: obtener una beneficio mínimo de 500 €/mes.

Para conseguir esos beneficios se consideran tres vías:

- Aplicación de pago. Google Play.
- Aplicación gratuita con publicidad. Google Play.
- Aplicación personalizada de cara al cliente. Google Play.

En la Tabla se presentan diversos precios de venta y diversos volúmenes de venta necesarios para llegar al beneficio deseado. Hay que tener en cuenta que Google (propietaria de Google Play) se lleva un 30% del beneficio de las ventas de la aplicación.

Precio	Volumen	Beneficios netos
1,99 €	359 descargas/mes	500,09 €
0,99 €	722 descargas/mes	500,35 €
0,50 €	1429 descargas/mes	500,15 €

Tabla 12: Combinaciones precio-descargas para obtener un beneficio de 500€.

En el caso de ser gratuita con publicidad se baraja un beneficio de entre 0,3\$ y 3\$ el eCPM (cada mil impresiones) por parte de iAd, mientras que AdMob se mueve entre los 0 y 0,5 \$.¹⁰⁷

Por último se barajaría la opción de crear aplicaciones personalizadas para cada colegio o instituto. El precio de la aplicación variaría en función del número de horas dedicadas. Considerando una media de 40 horas/aplicación cobrando a un precio de 12€/hora, la aplicación saldría a unos 480 €.

¹⁰⁷ Fuente: <http://www.startcapps.com/blog/como-ganan-dinero-las-aplicaciones-y-juegos-para-moviles/>



7. Trabajo futuro

En este proyecto se han iniciado los primeros ejemplos de lo que podría ser una aplicación futura a desarrollar. Se han presentado las principales funcionalidades de una aplicación de Realidad Aumentada, aunque no todas, y por lo tanto aún es un campo muy extenso a explorar. Entre los campos a estudiar y a desarrollar en un futuro próximo estaría la mejora en los contenidos gráficos de la aplicación, ya que es un factor muy importante de cara al éxito de la aplicación. Se han presentado modelos gráficos correctos pero siempre se puede y se debe mejorar.

Trabajar con docentes sería idóneo, intercambiar opiniones con ellos, recibir su consejo y colaborar en el desarrollo de aplicaciones con rigor científico y técnico.

Otro de los aspectos sería trabajar las transiciones y la interfaz de la aplicación para hacerla más atractiva al público. Para ello se podría pedir la opinión de voluntarios para intentar mejorar en los aspectos necesarios.

Uno de los elementos interesantes sería también la exploración más a fondo de los recursos que ofrece Vuforia, el desarrollo de los demás tipos de Targets como pueden ser palabras, cajas de cereales, marcadores, etc. No limitarse a imágenes bidimensionales únicamente. Probar otras funcionalidades.

También resultaría interesante explorar en el concepto de identificación de objetos, los algoritmos que realizan ese reconocimiento, la inteligencia artificial que existe detrás, etc.

Y en un futuro más lejano, pero igualmente próximo, quizás en un par de años, explorar sobre las posibilidades que ofrecerán las gafas de realidad aumentada.

8. Conclusiones

Este proyecto ha sido una gran oportunidad para mí, ya que creo que he aprendido cosas en diferentes ámbitos.

Primero de todo me ha permitido introducirme en la Realidad Aumentada un campo que desconocía y que hasta hace unos meses me parecía de ciencia ficción. Aún recuerdo la primera vez que me enseñaron en qué consistía y la verdad es que quedé impresionado. De este trabajo salgo con una visión más amplia sobre el tema y con los conocimientos básicos para desarrollar o trabajar en conceptos de Realidad Aumentada. Quizás lo más importante que me llevo es conocer que lo que me falta por aprender, sé qué caminos debo recorrer de ahora en adelante.

Este trabajo también me ha permitido conocer de cerca el programa Unity, con el que se pueden hacer miles de cosas más de las que he hecho pero que he descubierto que es una gran herramienta. Otro programa utilizado como Blender me ha permitido poner en práctica conocimientos de animación que he aprendido a lo largo de estos meses y también me ha llevado los mayores quebraderos de cabeza, al tener problemas en exportar e importar archivos a Unity, sin embargo es una herramienta muy potente. Por supuesto, no quiero olvidar a Vuforia sin el cual este proyecto no hubiera sido posible. Me voy con la espina clavada de no haber profundizado más en las posibilidades que ofrece y sobre todo, conocer qué es lo que hay detrás de ello. Inteligencia Artificial, reconocimiento de objetos o tracking son conceptos sobre los que se podría continuar el proyecto.

Por último destacar de nuevo, la suerte que he tenido de realizar el trabajo un campo que aún no se ha explotado y que te da la sensación de estar descubriendo nuevos horizontes.

Terminado el proyecto, sé que pueden mejorarse muchas cosas y que aún falta todo el trabajo por hacer. Sin embargo también he descubierto cosas nuevas, sé que hay que llevar a cabo las ideas que uno tiene y eso me lo ha permitido este trabajo. De nada sirve tener una buena idea sino la llevas a cabo.

“La mejor manera de empezar algo es dejar de hablar de ello y empezar a hacerlo”

(Walt Disney)



9. Agradecimientos

Al tutor de este proyecto, Sergi Grau, por su generosidad, comprensión y paciencia. Gracias por aguantar al autor durante meses, por aconsejar, por escuchar y sobre todo por compartir su tiempo.

A mis padres, mi hermana y mis alumnos por haberme animado en la realización del trabajo y por haberme dado su opinión cuando la he necesitado.

10. Bibliografía

10.1. Bibliografía referenciada

A continuación se presenta la bibliografía referenciada a lo largo del texto:

- Augmented reality. En: Wikipedia [en línea] [Consulta: marzo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Augmented_reality
- Visión Artificial. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Visi%C3%B3n_artificial.
- Computación gráfica. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Computaci%C3%B3n_gr%C3%A1fica.
- Object recognition. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Object_recognition.
- Blob detection. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Blob_detection.
- Video tracking. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Video_tracking.
- Flujo óptico. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Flujo_%C3%B3ptico
- Javier Pastor. Desarrollo de aplicaciones móviles (I): así está el mercado. En: Xataka[en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.xatakamovil.com/mercado/desarrollo-de-aplicaciones-moviles-i-asi-esta-el-mercado>
- Photoshop. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: <http://es.wikipedia.org/wiki/Photoshop>
- Gimp. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: <http://es.wikipedia.org/wiki/Gimp>
- Software de gráficos 3D. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Software_de_gr%C3%A1ficos_3D
- Blender. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: <http://es.wikipedia.org/wiki/Blender>
- Gonzalo Lorenzo. ¿Qué es 3D Studio Max? .En: Blogs.ua.es [en línea] [Consulta: mayo de 2014]. Disponible en: <http://blogs.ua.es/gonzalo/2010/03/31/%C2%BFque-es-3d-studio-max/>
- Maya. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: [http://es.wikipedia.org/wiki/Maya_\(aplicaci%C3%B3n_gr%C3%A1fica\)](http://es.wikipedia.org/wiki/Maya_(aplicaci%C3%B3n_gr%C3%A1fica))



10.2. Bibliografía completa

A lo largo del texto hay bibliografía referenciada (vista en el apartado anterior) y también hay notas a pie de página donde se detallan fuentes y también bibliografía. En este apartado se pretende disponer de toda la bibliografía de forma completa, detallada para cada apartado. Cabe destacar que aquí no se recogerá la fuente de las imágenes utilizadas, que ya viene indicada a lo largo del documento.

1. Introducción

- Realidad aumentada. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Realidad_aumentada
- Rotoscopia. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: <http://es.wikipedia.org/wiki/Rotoscopia>
- Croma. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: <http://es.wikipedia.org/wiki/Croma>
- Croma. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: <http://es.wikipedia.org/wiki/Croma>
- MITK pille - mobile medical augmented reality App for the Apple iPad. En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=T4mboj-GbEA>
- Augmented Architectural Demonstration. En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: https://www.youtube.com/watch?v=wOUk-u_hz3M
- Augmented Reality & Motion Capture Shopping App, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=NxQZuo6pFUw>

2.1. Realidad Virtual vs Realidad Aumentada vs Realidad Mixta

- Milgram, Paul; H. Takemura, A. Utsumi, F. Kishino (1994). "[Augmented Reality: A class of displays on the reality-virtuality continuum](#)" (pdf). *Proceedings of Telem manipulator and Telepresence Technologies*. pp. 2351–34. Retrieved 2007-03-15.
- Reality–virtuality continuum, En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Virtuality_Continuum
- Reactj ReacTable Trance live performance @ submixpro studio Torino 02, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=Mgy1S8qymx0>
- Spatial AR, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=iNT3eXwTfRY>
- PlayStation Vita - Augmented Reality, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=zXWSpcTnKTW>
- Kinect Augmented Virtuality, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: https://www.youtube.com/watch?v=mpTKgs_cMCU
- Immersive Virtual Reality Technology - Experience More, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=UT5xtpHRdIE>

2.2. La Realidad Aumentada

- Augmented reality. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Augmented_reality
- Sight, En: Vimeo [en línea] [Consulta: mayo de 2014]. Disponible en: <http://vimeo.com/46304267>
- Ronald Azuma Home, [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.ronaldazuma.com/>

2.2.1. Etapas de la Realidad Aumentada

- Augmented reality. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Augmented_reality
- Visión Artificial. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Visi%C3%B3n_artificial.
- Computación gráfica. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Computaci%C3%B3n_gr%C3%A1fica.

2.2.2. Reconocimiento de objetos y/o imágenes

- Object recognition. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Object_recognition
- Qualcomm AR SDK Image Target Sample App, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=42MvJ1QgPz8>
- Edge detection. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Edge_detection
- OpenCV edge detection, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=nxANbjzO0Q>
- Blob detection. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Blob_detection.

2.2.3. Tracking

- Video tracking. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://en.wikipedia.org/wiki/Video_tracking
- Mean-shift. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: <http://en.wikipedia.org/wiki/Mean-shift>
- Real-Time Adaptive 3D Face Tracking and Eye Gaze Estimation, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=ackTCF0sAfc>
- 3D Head Tracking Webcam OpenCV, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: http://www.youtube.com/watch?v=h9kPI7_vhAU
- Head Tracking for Desktop VR Displays using the WiiRemote, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=Jd3-eiid-Uw>



- FORTH Kinect 3D Hand Tracking Exemplar Tracking Sequene 1, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=iv-azKZlpbl#t=25>
- Gesture Controlled Pin Code Interface, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=CYJjQgV0mZY>
- Flujo óptico. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Flujo_%C3%B3ptico
- Optical Flow, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=JlLkkom6tWw>
- OpenCV CUDA Dense Optical Flow, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: https://www.youtube.com/watch?v=5rR_9Ylcg_s

2.2.4. Iluminación y renderizado

- Realistic Shadows for Mobile Augmented Reality, En: Youtube [en línea] [Consulta: mayo de 2014]. Disponible en: <https://www.youtube.com/watch?v=P3roYysl6pM>

2.2.5. Aplicaciones de RA

- Anatomy 4D - DAQRI, En: DAQRI [en línea] [Consulta: mayo de 2014]. Disponible en: <http://site.dagri.com/project/anatomy-4d/>
- Elements 4D - DAQRI, En: DAQRI [en línea] [Consulta: mayo de 2014]. Disponible en: <http://dagri.com/elements4D-mobile/>
- SKY MAP, En: Google Play [en línea] [Consulta: mayo de 2014]. Disponible en: <https://play.google.com/store/apps/details?id=com.google.android.stardroid&hl=en>
- WORD LENS, En: Google Play [en línea] [Consulta: mayo de 2014]. Disponible en: <https://play.google.com/store/apps/details?id=com.questvisual.wordlens.demo>
- POPAR TOYS, En: Popar Toys [en línea] [Consulta: mayo de 2014]. Disponible en: <https://popartoy.com/shop/>

3.1.1. Plataformas de desarrollo

- Javier Pastor. Desarrollo de aplicaciones móviles (I): así está el mercado. En: Xataka [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.xatakamovil.com/mercado/desarrollo-de-aplicaciones-moviles-i-asi-esta-el-mercado>
- Google Play, En: Google Play [en línea] [Consulta: mayo de 2014]. Disponible en: <https://play.google.com/store?hl=es>
- Google Play Developer, En: Google Play [en línea] [Consulta: mayo de 2014]. Disponible en: <http://developer.android.com/distribute/googleplay/start.html>
- Apple Store, En: Apple Store [en línea] [Consulta: mayo de 2014]. Disponible en: <http://store.apple.com/es>
- Apple [en línea] [Consulta: mayo de 2014]. Disponible en: <https://developer.apple.com/programs/ios/>
- Windows Phone Store [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.windowsphone.com/es-es/store>

- ¿Cuánto cuesta desarrollar para móviles? [en línea] [Consulta: mayo de 2014]. Disponible en <http://betabeers.com/forum/cuanto-cuesta-desarrollar-moviles-156/>
- Blackberry [en línea] [Consulta: mayo de 2014]. Disponible en <https://developer.blackberry.com/blackberryworld/>

3.1.2. Entorno de desarrollo

- Vuforia [en línea] [Consulta: mayo de 2014]. Disponible en : <https://www.vuforia.com/>
- Metaio [en línea] [Consulta: mayo de 2014]. Disponible en : <http://www.metaio.com/>
- Wikitude [en línea] [Consulta: mayo de 2014]. Disponible en : <http://www.wikitude.com/>
- Aurasma [en línea] [Consulta: mayo de 2014]. Disponible en : <http://www.aurasma.com/>

3.1.3. Lenguajes de programación

- Flujo óptico. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Flujo_%C3%B3ptico
- CSharp. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/C_Sharp
- Javascript. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: <http://es.wikipedia.org/wiki/Javascript>
- Boo. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: [http://es.wikipedia.org/wiki/Boo_\(programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Boo_(programaci%C3%B3n))

3.2.1. Imagen

- Adobe Photoshop. En: Wikipedia [en línea] [Consulta: mayo de 2014]. Disponible en: http://es.wikipedia.org/wiki/Adobe_Photoshop
- Gimp. [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.gimp.org.es/>
- Paint. [en línea] [Consulta: mayo de 2014]. Disponible en: <http://windows.microsoft.com/es-es/windows7/products/features/paint>

3.2.2. Gráficos 3D/Animación

- Blender [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.blender.org/>
- Autodesk 3ds Max [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.autodesk.es/products/autodesk-3ds-max/overview>
- Autodesk Maya [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.autodesk.es/products/autodesk-maya/overview>

3.2.3. Video

- Movie Maker [en línea] [Consulta: mayo de 2014]. Disponible en: <http://windows.microsoft.com/es-es/windows-live/movie-maker#t1=overview>
- Adobe Premiere [en línea] [Consulta: mayo de 2014]. Disponible en:



<http://www.adobe.com/es/products/premiere.html>

- Pinnacle Studio [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.pinnaclesys.com/PublicSite/sp/Products/studio/>

4.1. Impacto ambiental del desarrollo de la aplicación

- Fuente: WWF España. Observatorio de la Electricidad Mayo 2014[en línea]. Raquel García Monzón. Madrid,2014.[Consulta: mayo de 2014] Disponible en:http://awsassets.wwf.es/downloads/oe_mayo_02062014.pdf
- Energía que gasta un ordenador [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.leantricity.es/es/2012/07/11/cuanta-energia-gasta-un-ordenador-aproximaciones/>

5. Plan económico del proyecto

- Como ganan dinero las aplicaciones y juegos para móviles [en línea] [Consulta: mayo de 2014]. Disponible en: <http://www.startcapps.com/blog/como-ganan-dinero-las-aplicaciones-y-juegos-para-moviles/>