

Prototipo de aplicación de realidad aumentada para dispositivos móviles Android, como apoyo en el aprendizaje de conceptos básicos de electrónica.

Jorge Iván Cuadros Acosta

Máster universitario de Desarrollo de aplicaciones para dispositivos móviles

Trabajo final de máster DADM

Eduard Martín Lineros

Carles Garrigues Olivella

5 de junio de 2020



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Prototipo de aplicación de realidad aumentada para dispositivos móviles Android, como apoyo en el aprendizaje de conceptos básicos de electrónica</i>
Nombre del autor:	<i>Jorge Iván Cuadros Acosta</i>
Nombre del consultor/a:	<i>Eduard Martín Lineros</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	06/2020
Titulación::	<i>Máster Universitario de Desarrollo de aplicaciones para dispositivos móviles.</i>
Área del Trabajo Final:	<i>Trabajo final de máster DADM</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Realidad aumentada, Android, Educación</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.</i></p>	
<p>El desarrollo y estudio de materiales educativos que integren sistemas de realidad aumentada, ha ido aumentando progresivamente en los últimos años. Esto es debido en gran parte a la capacidad que tiene la realidad aumentada de enriquecer las experiencias de los estudiantes en sus procesos de aprendizaje ya que permite por medio de una pantalla mezclar imágenes reales con objetos virtuales con los cuales puede interactuar y en algunos casos hasta experimentar de una forma segura y controlada.</p> <p>Teniendo en cuenta lo anterior, a lo largo de este documento, se presentará el proceso de desarrollo de un prototipo de una aplicación de realidad aumentada que sirva como apoyo para el proceso de aprendizaje de conceptos básicos de electrónica.</p> <p>Dicho desarrollo se realizó siguiendo una metodología de desarrollo en cascada con la cual por medio de pasos completamente secuenciales se desarrollaron las etapas de diseño desde la conceptualización de usuarios hasta desarrollo de un prototipo de alta fidelidad, prosiguiendo con el desarrollo de la aplicación utilizando el IDE y motor de videojuegos Unity 3D en conjunto con el SDK de realidad aumentada Vuforia, con lo cual se obtuvo el producto final que se muestra como resultado de trabajo final de máster.</p>	

Abstract (in English, 250 words or less):

The development and study of educational materials that integrate augmented reality systems has been increasing progressively in recent years. This is due in large part to the ability that reality has to enrich the experiences of students in their learning processes since it allows using a screen to mix real images with virtual objects with which they can interact and in some cases even experiment in a safe and controlled way.

Taking into account the above, throughout this document, the process of developing a prototype of an augmented reality application that serves as a support for the process of learning basic electronics concepts is presented.

This development was carried out following a cascade development methodology with quality through completely sequential steps. The design stages were developed from the conceptualization of users to the development of a high-fidelity prototype, following the development of the application using the IDE and Unity 3D video game engine in conjunction with the Vuforia augmented reality SDK, obtaining the final product that is shown as the result of final master's work.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo.....	4
1.4.1 Tareas a realizar en el desarrollo del proyecto.....	4
1.4.2. Recursos a utilizar en el desarrollo del proyecto.....	5
1.4.3. Cronograma del proyecto.....	7
1.5 Breve resumen de productos obtenidos.....	10
1.6 Breve descripción de los otros capítulos de la memoria.....	10
2. Marco Conceptual.....	12
2.1. Realidad aumentada.....	12
2.2. Realidad aumentada en la educación.....	14
2.3. Motores de videojuegos.....	15
2.4. SDK para la implementación de realidad aumentada.....	17
3. Marco Procedimental.....	17
3.1. Diseño de la aplicación móvil.....	17
3.1.1. Recopilación de información.....	18
3.1.2. Parámetros de diseño de la aplicación móvil.....	23
3.1.3. Fichas de usuario.....	24
3.1.4. Contextos de uso del sistema.....	26
3.1.5. Diagrama general de casos de uso.....	27
3.1.6. Diagrama de flujos de interacción del sistema.....	28
3.1.5. Prototipo de alta fidelidad.....	30
3.2. Implementación de la aplicación móvil.....	33
3.2.1. Desarrollo de modelos 3D.....	33
3.2.2. Implementación de SDK de realidad aumentada.....	36
3.2.3. Desarrollo de la aplicación Geek electrónica AR.....	41
3.2.4. Evaluación de la aplicación.....	58
3.2.5. Corrección de fallos.....	61
3. Conclusiones.....	62
4. Glosario.....	64
5. Bibliografía.....	65
6. Anexos.....	66
6.1. Ejemplos de imágenes válidas para prueba de la aplicación.....	66

Lista de gráficos

Gráfico 1. Pasos de desarrollo del presente proyecto bajo método de desarrollo en cascada	3
Gráfico 2. Cronograma de actividades del proyecto	9
Gráfico 3. Etapas y procesos de la realidad aumentada	12
Gráfico 4. Encuesta: Edad	18
Gráfico 5. Encuesta: Sexo	18
Gráfico 6. Encuesta: Ocupación	19
Gráfico 7. Encuesta: Sistema operativo de su dispositivo móvil.....	19
Gráfico 8. Encuesta: Tipo de interfaz preferida	19
Gráfico 9. Encuesta: Tipo de conectividad preferida	20
Gráfico 10. Encuesta: Lugar de uso de las aplicaciones	20
Gráfico 11. Encuesta: Ha usado aplicaciones de realidad aumentada	20
Gráfico 12. Encuesta: finalidad de uso de aplicaciones de realidad aumentada	21
Gráfico 13. Encuesta: ¿Cree que los objetos 3D pueden facilitar el proceso de aprendizaje?.....	21
Gráfico 14. Encuesta: Disposición de usar aplicación de realidad aumentada en proceso de enseñanza-aprendizaje	21
Gráfico 15. Diagrama UML de casos de uso de aplicación Geek Electrónica AR	28
Gráfico 16. Diagrama de flujo de interacción del sistema	29
Gráfico 17. Diagrama de clases escena AR.....	46
Gráfico 18. Diagrama de clases de la escena Resistor	49
Gráfico 19. Diagrama de clases de la escena Display7Seg.....	51
Gráfico 20. Diagrama de clases de la escena Motor	54
Gráfico 21. Diagrama de clases de la escena Diode.....	56
Gráfico 22. Diagrama de clases de la escena Capacitor.....	58

Lista de ilustraciones

Ilustración 3. Ficha de usuario 1	25
Ilustración 4. Ficha de usuario 2.....	26
Ilustración 5. Pantalla principal de la aplicación	30
Ilustración 6. Pantalla de realidad aumentada - Detectando marcador	31
Ilustración 7. Pantalla de realidad aumentada - Modelo 3D sobrepuesto al marcador.....	31
Ilustración 8. Pantalla funcionalidad del modelo resistencia	32
Ilustración 9. Pantalla de créditos de la aplicación	32
Ilustración 10. Entorno de software Blender. Desarrollo del modelo de una resistencia	34
Ilustración 11. Resultado del renderizado del modelo 3D de la resistencia	34
Ilustración 12. Resultado del renderizado del modelo 3D del diodo rectificador	35
Ilustración 13. Resultado del renderizado del modelo 3D del condensador.....	35
Ilustración 14. Resultado del renderizado del modelo 3D del display 7 Segmentos	35
Ilustración 15. Resultado del renderizado del modelo 3D del motor (Armado). 36	
Ilustración 16. Resultado del renderizado del modelo 3D del motor (Desarmado)	36
Ilustración 17. Imágenes diseñadas como marcadores de realidad aumentada	37
Ilustración 18. Imágenes adicionales generadas para el display siete segmentos.....	38
Ilustración 19. Base de datos creada con marcadores creada en vuforia	38
Ilustración 20. Marcador de resistencia con puntos detectados por el sistema de vuforia	39
Ilustración 21. Habilidad del soporte a realidad aumentada con vuforia en Unity 3D.....	40
Ilustración 22. Eliminar Vulkan Graphics de las configuraciones del proyecto . 40	
Ilustración 23. Herramientas del SDK de vuforia en Unity 3D	40
Ilustración 24. Importación de base de datos con los marcadores AR	41
Ilustración 25. Aspecto escena Inicio	42
Ilustración 26. Muestra del código realizado para la transición entre escenas. 42	
Ilustración 27. Aspecto escena Credits	43
Ilustración 28. Aspecto escena AR sin detección de marcador.....	43
Ilustración 29. Aspecto escena AR con marcador de resistencia detectado	44
Ilustración 30. Aspecto escena AR con marcador de motor detectado	44
Ilustración 31. Muestra del código realizado para la transición de la escena AR a la escena correspondiente para cada elemento electrónico	45
Ilustración 32. Aspecto de la escena Resistor.....	46
Ilustración 33. Aspecto de la resistencia luego de rotar el modelo.....	47
Ilustración 34. Muestra de modificación del atributo rotate del componente transform del modelo 3D	47
Ilustración 35. Aspecto de la resistencia luego de modificar sus valores	48
Ilustración 36. Arreglo que contiene todos los colores del código de colores de resistencias	48

Ilustración 37. Muestra de modificación del atributo color para la franja 1 de la resistencia	48
Ilustración 38. Aspecto escena Display7Seg	50
Ilustración 39. Aspecto de la escena Display7Seg al hacer modificaciones sobre el modelo.....	50
Ilustración 40. Muestra de fragmento de código utilizado para realizar los cambios en botones y modelo 3D	51
Ilustración 41. Aspecto escena Motor con modelo 3D armado	52
Ilustración 42. Línea de tiempo de animación de despiece del motor	52
Ilustración 43. Controlador de animaciones del motor	52
Ilustración 44. Aspecto escena Motor con despiece del motor y descripción de la pieza rotor.....	53
Ilustración 45. Muestra del diccionario creado para las descripciones de las piezas del motor	53
Ilustración 46. Aspecto escena Diode	55
Ilustración 47. Aspecto escena Diode con circuito rectificador de media onda seleccionado	55
Ilustración 48. Muestra de la clase AnimationGifs.....	56
Ilustración 49. Aspecto escena Capacitor	57
Ilustración 50. Aspecto teclado para el ingreso de datos	57
Ilustración 51. Aspecto escena Capacitor con muestra de resultados del calculo	58
Ilustración 52. Reconocimiento símbolo resistencia de una revista	59
Ilustración 53. Reconocimiento símbolo diodo del simulador electrónico multisim 12	59
Ilustración 54. Reconocimiento símbolo condensador del simulador electrónico multisim 12	60
Ilustración 55. Reconocimiento símbolo motor de los resultados de búsquedas de Google.....	60
Ilustración 56. Reconocimiento símbolo display 7 segmentos de los resultados de búsqueda de Google	60
Ilustración 57. Configuraciones de optimización utilizadas para corregir fallo..	61
Ilustración 58. Imagen Marcador resistencia	66
Ilustración 59. Imagen Marcador diodo rectificador.....	66
Ilustración 60. Imagen Marcador condensador	67
Ilustración 61. Imagen Marcador motor.....	67
Ilustración 62. Imagen Marcador display siete segmentos.....	68

Lista de tablas

Tabla 1. Dedicación horaria para el desarrollo del proyecto	7
Tabla 2. Dedicación horaria por cada PEC	7
Tabla 3. Dedicación horaria por cada tarea a realizar	8
Tabla 4. Ventajas y desventajas de la realidad aumentada aplicada a la educación	15

1. Introducción

1.1 Contexto y justificación del Trabajo

El avance tecnológico constante que se ha presentado en los últimos años, en especial en áreas del desarrollo digital y la computación, han hecho que las personas se sientan cada vez más cómodas con el uso de diferentes herramientas tecnológicas como apoyo o complemento para la ejecución de sus tareas cotidianas. Y en este sentido, la educación no ha sido apática a este desarrollo tecnológico, es así que ha adoptado diferentes técnicas de las Tecnologías de la Información y la Comunicación (TIC) para adaptarlas en el proceso de enseñanza – aprendizaje.

Sin embargo, estos progresos tecnológicos en el sector educativo han sido orientados en gran parte a la ampliación de la cobertura por medio de la virtualidad y a la globalización del conocimiento a través de herramientas o conceptos como la web 2.0. Pero, en el caso específico de la creación o desarrollo de materiales de apoyo al proceso de enseñanza-aprendizaje, los materiales que en su mayoría se siguen distribuyendo son los libros tradicionales, en donde el mayor aporte tecnológico que incluyen son materiales extra que pueden ser contenidos multimedia o applets que el estudiante puede observar o interactuar en cierto modo, pero que como común denominador siguen siendo textos teóricos o gráficos ilustrativos en dos dimensiones (2D), que para ciertas asignaturas como las ciencias naturales o las ingenierías pueden llegar a dificultar o ralentizar el proceso de aprendizaje, teniendo en cuenta que los objetos reales que se explican en estas áreas y con los que el alumno, tendrá que interactuar a futuro, así como todo su entorno, son modelos tridimensionales (3D).

En este sentido, la realidad aumentada, que aunque los inicios de su desarrollo se remontan a la década de los noventa, sigue siendo una tecnología nueva y poco explorada en diferentes campos, entre ellos la educación. Esta tecnología que teniendo en cuenta lo anterior, sigue siendo relativamente nueva, puede llegar a configurarse como una alternativa para el desarrollo de nuevos materiales educativos, con los que el estudiante pueda adquirir un mayor contexto de las temáticas que se abordan en las diferentes asignaturas de su formación académica, ofreciéndole modelos 3D de los objetos con los que posteriormente va a interactuar, aumentando el grado de inmersión y compenetración que el alumno logrará con el material educativo. Esto teniendo en cuenta, que autores como Azuma (1997), definen a la realidad aumentada como una tecnología que sobrepone objetos virtuales en un entorno real, mezclándolos y dando la impresión que se encuentran todos en el mismo entorno, lo cual puede mejorar la percepción sensorial del usuario.

Con el presente trabajo, se pretende desarrollar un material educativo que implemente la tecnología de realidad aumentada como apoyo en el aprendizaje de conceptos básicos de electrónica y que le permita al estudiante interactuar con diferentes modelos 3D de objetos electrónicos, a los cuales les puede modificar diferentes parámetros y de esta forma observar como estos cambios afectan el funcionamiento del dispositivo, agregando mayor interactividad que los libros estáticos tradicionales en el proceso de aprendizaje, además de una mejor percepción de realidad que los applets o simuladores tradicionales que generalmente son desarrollados como entornos 2D. Todo esto dentro de un ambiente controlado.

La implementación del material propuesto, se hará por medio de un prototipo de aplicación móvil construida con el entorno de desarrollo (IDE) y motor de videojuegos Unity 3D y el SDK Vuforia. Dicha aplicación interactuará con una cartilla que contiene las explicaciones teóricas de los conceptos básicos de electrónica que se pretenden explicar y además contiene los marcadores para ser leídos por la aplicación, necesarios para sobreponer los objetos virtuales 3D en el entorno real y completar de esta forma la experiencia de realidad aumentada.

1.2 Objetivos del Trabajo

Objetivo General:

- Desarrollar un prototipo de aplicación de realidad aumentada para dispositivos móviles Android, que sirva como apoyo al proceso de enseñanza de conceptos básicos de electrónica.

Objetivos específicos:

- Desarrollar mínimo cinco modelos 3D de elementos electrónicos que serán mostrados en realidad aumentada.
- Desarrollar el entorno gráfico de la aplicación propuesta.
- Desarrollar la programación que permita la interacción del usuario con los modelos realizados

1.3 Enfoque y método seguido

Para el desarrollo de la aplicación móvil, se tuvieron en cuenta diferentes métodos para el desarrollo de software, así como los tipos de aplicaciones móviles que existen y los diferentes lenguajes o entornos de desarrollo móvil existentes. Con esta información y la premisa de desarrollar una aplicación móvil nueva, empezando el trabajo de diseño

e implementación desde ceros, se tomó la decisión de utilizar lo siguiente:

Método de desarrollo de software:

En este sentido se evaluaron los métodos de desarrollo en cascada (waterfall), desarrollo rápido de aplicaciones y desarrollo ágil. Teniendo en cuenta las características de estos métodos, se optó por utilizar el modelo de cascada ya que su característica principal es tener un proceso estructurado, secuencial y una planificación del trabajo clara y estricta desde el inicio del proyecto, tal cual como sucede con la estructura planteada para el desarrollo del trabajo final de máster.

Elaboración propia



Gráfico 1. Pasos de desarrollo del presente proyecto bajo método de desarrollo en cascada

En cuanto a los métodos de desarrollo rápido de aplicaciones y desarrollo ágil, se descartaron ya que están basados en procesos iterativos, que requieren de demasiadas revisiones en diferentes instantes del desarrollo del proyecto, lo cual en alguna clase de proyectos puede ser beneficioso para que el cliente pueda hacer revisiones y reorientar el proyecto de forma oportuna, pero para este proyecto en específico, puede representar retrasos de tiempo, además

de innecesario ya que desde un principio se han definido todas las etapas y características que debe tener el producto final a entregar.

Tipo de aplicación y entorno de desarrollo:

Entre los diferentes tipos de aplicaciones que existen, se encuentran las nativas, las web y las híbridas, cada uno de estos tipos cuenta con ventajas y desventajas que tienen que ser evaluadas dependiendo de los requerimientos del proyecto.

La aplicación a desarrollar para este proyecto, requiere el uso de algunas partes de hardware o periféricos del dispositivo móvil como lo son la cámara y algunos sensores. Por esta razón, no solo es recomendable, sino que es necesario desarrollar una aplicación móvil nativa, ya que dentro de las opciones es la única que permite acceder a los recursos del dispositivo de una forma más libre, debemos tener en cuenta que por términos de seguridad del usuario, las aplicaciones web y las aplicaciones híbridas, tienen algunas restricciones sobre el acceso o control al hardware del dispositivo.

En cuanto al entorno de desarrollo a utilizar, se optó por emplear un entorno para el desarrollo multiplataforma como lo es Unity 3D, esta decisión está fundamentada en la necesidad de trabajar con objetos virtuales en 3D, para lo cual este IDE, presenta gran facilidad de trabajo ya que contiene un motor de videojuegos con una gran cantidad de herramientas que simplifica el trabajo con objetos tridimensionales, además de la fácil integración de diferentes SDK para llevar a cabo tareas necesarias para el desarrollo de la aplicación, como por ejemplo el SDK de vuforia para trabajar realidad aumentada.

1.4 Planificación del Trabajo

1.4.1 Tareas a realizar en el desarrollo del proyecto

Para el desarrollo de este proyecto, se tienen definidas tres etapas las cuales están demarcadas por los diferentes entregables descritos en cada una de las pruebas de evaluación continuada (PEC), de la asignatura de trabajo final de máster DADM. Las etapas a considerar y cada una de las tareas que implican se relacionan a continuación:

Etapas de planeación de trabajo (PEC1):

Esta es la fase inicial del proyecto, implica la recopilación de información, determinación del trabajo que se va a realizar y la planeación de recursos, fases del proyecto, tareas a realizar y un cronograma detallado a seguir para la consecución del objetivo general. Las tareas a llevar a cabo en esta etapa son las siguientes:

- Recopilación de información (Proceso constante durante todo el desarrollo del proyecto)
- Determinar el proyecto a realizar.
- Determinar objetivos del proyecto.
- Realizar la justificación del proyecto
- Determinar el enfoque y modelo de trabajo
- Determinar el plan de trabajo a seguir

Etapa de diseño de la aplicación (PEC2):

En esta fase se realiza el proceso de diseño y prototipado de la aplicación móvil que se va a desarrollar. Durante este proceso se deben llevar a cabo las siguientes tareas:

- Definición de usuarios y contextos de uso.
- Realizar fichas de usuario.
- Definir flujos de interacción con el sistema.
- Desarrollo de un prototipo de alto nivel.

Etapa de implementación de la aplicación (PEC3):

En esta fase se desarrollan e integran en un solo producto todas las tareas necesarias para conseguir la aplicación final que se presentara como trabajo de máster. Las tareas a realizar en esta etapa son las siguientes:

- Modelado 3D de los objetos virtuales.
- Creación de los marcadores de realidad aumentada e integración del SDK de realidad aumentada con Unity 3D.
- Desarrollo de la aplicación móvil.
- Evaluación de la aplicación.
- Corrección de fallos.

1.4.2. Recursos a utilizar en el desarrollo del proyecto

Para la realización del presente proyecto, será necesario el uso de diferentes herramientas informáticas en cada una de las etapas de desarrollo. A continuación, se mencionarán que herramientas se utilizarán de acuerdo al objetivo o etapa del proyecto que se requiera.

Diseño y prototipado de la aplicación móvil:

Para esta etapa específica, se utilizarán diferentes herramientas disponibles en la web que facilitan o agilizan el desarrollo de tareas

típicas de esta fase de los proyectos como lo son la realización de fichas de usuario y el prototipado de alta fidelidad.

- ***xtensio***: esta es una plataforma web, que entre sus funciones cuenta con la posibilidad de crear fichas de usuario a través de plantillas llamativas, personalizables e intuitivas para la edición de la información por parte del usuario. Cuenta con una opción de uso gratuito la cual tiene características limitadas, pero que son suficientes para el desarrollo del presente proyecto.
- ***Marvel App***: plataforma web, para el desarrollo de prototipos de alta fidelidad a los que se les puede agregar herramientas de interacción, con lo cual se logra que el usuario que utilice el prototipo tenga una mejor percepción del producto final que puede llegar a recibir. Cuenta con una versión limitada de uso gratuito que brinda herramientas suficientes para el desarrollo de este proyecto.

Modelado 3D:

Para la implementación efectiva de la realidad aumentada, es necesario el desarrollo de los modelos 3D que se le mostrarán al usuario de la aplicación y que serán sobrepuestos al entorno real. Para tal fin se utilizará el software de uso libre blender el cual cuenta con una amplia documentación y comunidad, lo que facilita la consecución de información y la solución de posibles problemas durante la etapa de desarrollo de los modelos. Además, los modelos 3D generados en blender, se pueden integrar fácilmente al entorno de desarrollo de aplicaciones móviles que se va a utilizar.

Entorno de desarrollo de la aplicación móvil:

Este proyecto requiere de un entorno de desarrollo potente y que además facilite la integración y trabajo con modelos 3D y con diferentes API o SDK. Teniendo esto en cuenta, se optó por el entorno de desarrollo y motor de videojuegos Unity 3D. A pesar de su potencia, la interfaz gráfica de usuario es sencilla de manejar y su programación se hace por medio del lenguaje de programación C#, el cual cuenta con gran cantidad de documentación y soporte.

Implementación de realidad aumentada:

Para la implementación de realidad aumentada se requiere un sistema que cuente con un algoritmo de visión artificial capaz de reconocer diferentes marcadores, para luego asociarlos a objetos 3D virtuales que serán sobrepuestos a un entorno real.

Para llevar a cabo este proceso existen diferentes herramientas tanto de código abierto, uso gratuito y uso de pago. Para este proyecto hasta la fecha se tiene planeado utilizar el SDK vuforia, gracias a la facilidad de integración junto con Unity 3D. Sin embargo, debido a los tipos de licencia que maneja, donde la única opción gratuita es la de desarrollador y que no permite hacer una distribución de las aplicaciones, se están evaluando otras opciones que no presenten limitaciones en cuanto a la distribución de la aplicación como por ejemplo Google ARCore y ARToolKit.

1.4.3. Cronograma del proyecto

Las tareas a realizar durante el desarrollo del presente proyecto se han definido en términos de horas, de forma tal que para cada día de la semana se establecieron una cantidad de horas de trabajo determinadas y así mismo, cada una de las PEC tiene asignadas una cantidad de horas definidas por el calendario académico y a cada una de las tareas, se les ha definido una cantidad de horas específicas para su ejecución, como se puede observar en las siguientes tablas.

Elaboración propia

Día	Tiempo de dedicación
Lunes	8 Horas
Martes	2 Horas
Miércoles	2 Horas
Viernes	6 Horas
Sábado	8 Horas
Domingo	4 Horas
Total Horas semana	30 Horas

Tabla 1. Dedicación horaria para el desarrollo del proyecto

Elaboración propia

PEC	Tiempo de dedicación según calendario académico y dedicación estudiante
PEC 1 (Plan de trabajo)	72 Horas
PEC 2 (Diseño)	90 Horas
PEC 3 (Implementación)	180 Horas
Total Horas (Plan trabajo – Implementación)	342 Horas

Tabla 2. Dedicación horaria por cada PEC

PEC a que corresponde la tarea	Tarea a realizar	Tiempo de dedicación
PEC 1 (72 Horas)	Recopilación de información	32 Horas
	Determinar proyecto a realizar	10 Horas
	Determinar objetivos del proyecto	2 Horas
	Realizar la justificación del proyecto	10 Horas
	Determinar plan de trabajo	14 Horas
	Determinar enfoque y modelo a seguir	4 Horas
PEC 2 (90 Horas)	Recopilación de información	32 Horas
	Definición de usuarios y contextos de uso	4 Horas
	Elaboración de fichas de usuario	6 Horas
	Definir flujos de interacción con el sistema	6 Horas
	Elaboración de prototipo de alto nivel	24 Horas
	Documentación del proceso	18 Horas
PEC 3 (180 Horas)	Recopilación de información	32 Horas
	Modelado 3D de objetos virtuales	20 Horas
	Creación de marcadores de realidad aumentada	6 Horas
	Integración del SDK de realidad aumentada y marcadores con el entorno de desarrollo.	10 Horas
	Desarrollo de la aplicación móvil	55 Horas
	Evaluación de la aplicación	12 Horas
	Corrección de fallos	27 Horas
	Documentación del proceso	18 Horas

Tabla 3. Dedicación horaria por cada tarea a realizar

En la figura que se muestra a continuación, se puede observar un diagrama de Gantt, donde se especifica el orden de ejecución de las diferentes tareas a realizar durante el desarrollo del proyecto.

Elaboración propia

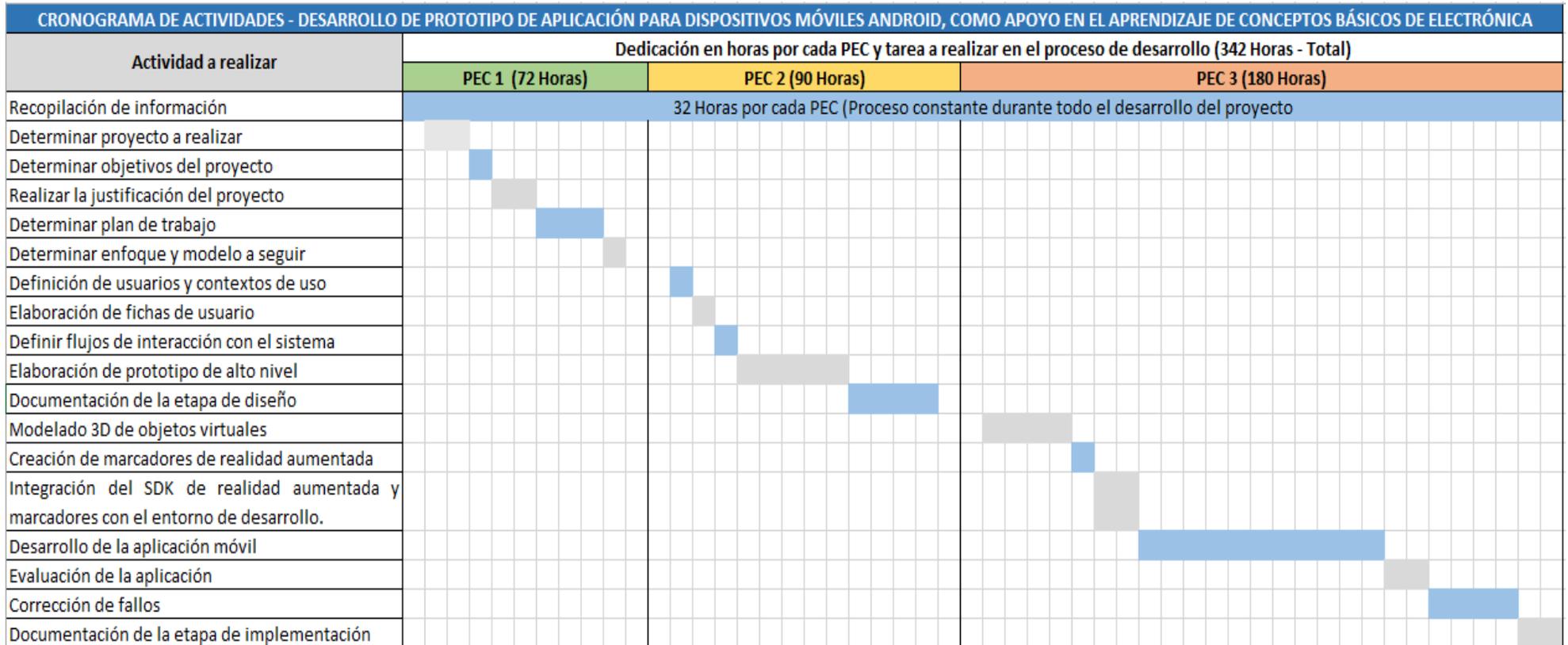


Gráfico 2. Cronograma de actividades del proyecto

1.5 Breve resumen de productos obtenidos

El producto resultante de este trabajo final de máster, es una aplicación móvil para dispositivos Android (prototipo), se habla de prototipo y no de producto final, ya que, aunque la aplicación es completamente funcional, cuenta con un número de modelos 3D bastante limitado (mínimo cinco modelos), esto es debido al poco tiempo con el que se cuenta para el desarrollo del trabajo final de máster, contrastado con el tiempo elevado que requiere hacer el diseño y modelado 3D de una extensa cantidad de dispositivos electrónicos y su posterior implementación dentro de la aplicación de realidad aumentada.

Aplicación móvil para dispositivos Android (prototipo):

Este producto será una aplicación de realidad aumentada, para ser ejecutada en dispositivos móviles Android. Su desarrollo se hará utilizando el IDE y motor de juegos Unity 3D, en conjunto con el SDK vuforia el cual proveerá las herramientas y facilitará el proceso de implementación de realidad aumentada en la aplicación móvil.

1.6 Breve descripción de los otros capítulos de la memoria

En la presente memoria de trabajo final de máster, para ofrecer una completa explicación y contextualización sobre el trabajo desarrollado, se tendrán en cuenta los siguientes capítulos:

- ***Marco conceptual:*** En este capítulo, se abordarán conceptos relacionados con los temas que se tienen que abordar, usar o implementar para un correcto desarrollo del proyecto.

Algunos de los temas a abordar serían:

- Realidad aumentada
 - Realidad aumentada en la educación
 - Motores de videojuegos
 - SDK para implementación de realidad aumentada
- ***Marco procedimental o desarrollo del proyecto:*** En este capítulo, se presentará toda la documentación del trabajo realizado durante el desarrollo del proyecto. Tiene como finalidad evidenciar y explicar el trabajo realizado en cada una de las fases de diseño e implementación del proyecto.

Algunos temas a tratar en este capítulo son:

- Diseño de la aplicación
 - Definición de usuario y contextos de uso
 - Fichas de usuario
 - Flujos de interacción del sistema
 - Prototipo de alta fidelidad de la aplicación

- Implementación de la aplicación
 - Modelado 3D de los objetos virtuales 3D
 - Creación de marcadores de realidad aumentada
 - Integración del SDK de realidad aumentada con el IDE Unity 3D.
 - Desarrollo de la aplicación
 - Evaluación de la aplicación
 - Corrección de fallos

2. Marco Conceptual

2.1. Realidad aumentada

La Fundación Telefónica en su libro publicado en 2011 titulado “Realidad aumentada: una nueva lente para ver el mundo”¹, menciona como el ser humano entiende su realidad física por medio de lo que puede llegar a experimentar con sus sentidos (vista, tacto, gusto, olfato y oído) y como con las TIC progresivamente se han ido desarrollando o encontrando distintas estrategias con las cuales enriquecer el entorno y la experiencia del ser humano a través de una estimulación extra a alguno de los sentidos anteriormente mencionados.

Teniendo esto en cuenta, la realidad aumentada (AR), es una tecnología enmarcada dentro del área de la informática con la cual se busca enriquecer la percepción que tiene el usuario de la realidad, al estimular el sentido de la vista integrando objetos o imágenes digitales a un entorno real, utilizando como agente integrador de estos dos tipos de elementos (real – virtual) un dispositivo como lo puede ser una pantalla.

Dicho de otra forma, lo que se pretende con la realidad aumentada es enriquecer el entorno real en el que se encuentra inmerso el usuario agregando por medios digitales y a través del uso de dispositivos electrónicos unos objetos virtuales con los cuales se puede brindar una mayor información o incluso proveer nuevas actividades o experiencias dentro de ese mismo entorno real.

Para lograr este proceso debe existir un agente intermedio entre el usuario y el mundo real, el cual se trata de un dispositivo electrónico capaz de capturar una imagen del mundo real, procesar la información capturada y colocar sobre esta un objeto virtual, para finalmente presentarle al usuario esta imagen final en donde se mezcla lo real con lo virtual.

Elaboración propia

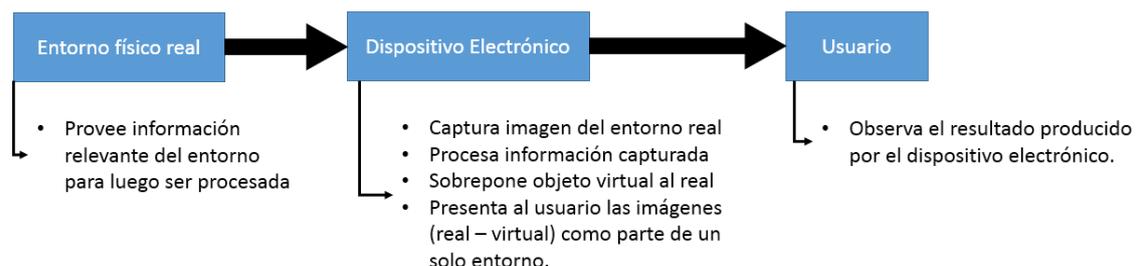


Gráfico 3. Etapas y procesos de la realidad aumentada

¹ Telefónica, F. (2011). *Realidad Aumentada: una nueva lente para ver el mundo*. Fundación Telefónica, 10-13.

Por lo tanto, teniendo en cuenta el grafico anterior, para obtener un producto de realidad aumentada, o como es el caso del producto objetivo del presente trabajo una aplicación móvil de realidad aumentada, es necesario contar como mínimo con los siguientes elementos:

Entorno físico real:

Este es el que le provee al sistema de realidad aumentada toda la información que debe procesar, para posteriormente sobre ella agregar los objetos virtuales que van a proporcionar la experiencia deseada. Esta información puede ser de distintos tipos como por ejemplo información de variables físicas (posición, velocidad, aceleración, etc), imágenes bidimensionales o incluso objetos tridimensionales que se comportan como marcadores y sobre los cuales se presenta el objeto virtual.

Para el caso de la información de las variables físicas antes mencionadas, se suelen desarrollar aplicaciones de realidad aumentada para dispositivos móviles ya que estos dentro de su hardware incluyen sensores o circuitos internos capaces de medir estas variables como por ejemplo el GPS, acelerómetro o giroscopio.

Dispositivo electrónico:

Es el encargado de capturar la información del entorno físico real, procesarla para encontrar los detalles característicos de cada espacio y sobre estos colocar un objeto virtual, dando la impresión que ambos objetos hacen parte de un mismo entorno. Para lograr este efecto, el dispositivo electrónico que va a ejecutar el proceso de realidad aumentada debe tener como mínimo los siguientes elementos a nivel de hardware:

- **Cámara:** se encarga de capturar las imágenes del entorno y tomar todos los detalles que puedan servir como información para determinar los posibles puntos en donde se va a incrustar el objeto virtual.
- **Procesador:** se encarga de procesar toda la información obtenida por la cámara, este procesamiento se realiza a través de distintos algoritmos que reconocen los distintos patrones de una imagen u objeto o en el caso de ser una aplicación de realidad aumentada basada en el posicionamiento u otras variables físicas, ejecuta los algoritmos para el procesamiento de estas señales transmitidas por los sensores.
- **Pantalla:** se encarga de presentar el resultado final del proceso al usuario, donde se mezcla lo real con lo virtual y el usuario puede interactuar con esta nueva experiencia.

- **Sensores:** en casos de aplicaciones de realidad aumentada basadas en el posicionamiento u otras variables, es necesario que el dispositivo electrónico cuente con sensores capaces de medir dichas variables.

2.2. Realidad aumentada en la educación

Las TIC, han incursionado en cada uno de los aspectos del diario vivir del ser humano actual y la educación no ha de ser la excepción a este fenómeno, siendo así que conforme pasa el tiempo, cada vez es más común encontrarse diferentes herramientas o productos tecnológicos siendo partícipes del aula u otros espacios de formación, ya sea en el apoyo en el proceso de enseñanza – aprendizaje o como medio motivacional para el cuerpo estudiantil.

Estas herramientas o productos pueden ir desde elementos sencillos como el desarrollo sitios web o wikis sobre un tema específico aprovechando la característica de interactividad y participación que ofrece la web 2.0, hasta el desarrollo de software especializado sobre áreas del conocimiento puntuales o niveles de educación definidos.

Dentro de este grupo de desarrollo de software especializado, se encuentra el desarrollo de aplicaciones que implementan la realidad aumentada. Dichas aplicaciones tienen diferentes objetivos tal como lo mencionan (Gómez, Rodríguez & Marín, 2019) como lo pueden ser la innovación, la motivación estudiantil, el entrenamiento de los estudiantes en habilidades o conocimientos específicos que normalmente requieren espacios controlados como por ejemplo en áreas como la medicina, las ciencias o la ingeniería, entre otras.

Diferentes autores han realizado estudios sobre el impacto, ventajas y desventajas de la realidad aumentada en la educación en diferentes áreas o contextos. Por ejemplo (Kellems, Cacciatore & Osborne, 2019) realizan un estudio en donde se implementa la realidad aumentada como una estrategia de enseñanza de las matemáticas en estudiantes secundaria con discapacidades, otro caso podría ser (Wang, 2017) que presenta un estudio en el cual se explora la efectividad del uso de técnicas de realidad aumentada en un curso de edición de software y de esta misma forma existen diferentes ejemplos de aplicación en contextos educativos distintos.

En una recopilación y estudio de distintos artículos, (Gómez, Rodríguez & Marín, 2019) logran determinar distintas ventajas y desventajas del uso de la realidad aumentada dentro del campo de la educación dentro de las cuales se encuentran:

REALIDAD AUMENTADA APLICADA A LA EDUCACIÓN	
Ventajas	Desventajas
Facilidad para apreciar sucesos o situaciones que sin la realidad aumentada sería complejo.	Dificultades por limitaciones técnicas.
Aumenta la motivación del estudiante	Requiere una adecuada guía para su uso.
Fomenta la creación y desarrollo de nuevos entornos de aprendizaje interactivos	Si el docente no tiene habilidades tecnológicas previas debe documentarse previamente.

Tabla 4. Ventajas y desventajas de la realidad aumentada aplicada a la educación

Con lo presentado anteriormente, se puede observar que a pesar de lo relativamente nueva que es la tecnología de la realidad aumentada (respecto a otras tecnologías informáticas), su uso y potencial de aplicabilidad en el ámbito educativo, no ha pasado desapercibido, presentando una cantidad considerable de estudios académicos sobre el tema, ya sea en el aspecto de consideraciones y favorabilidad de esta tecnología para la educación, así como proyectos de desarrollo de software de realidad aumentada para el uso en aula.

2.3. Motores de videojuegos

En el inicio de lo que hoy es la industria de los videojuegos, cada vez que se desarrollaba uno de estos juegos, el equipo de trabajo dispuesto para tal fin, no solamente debía responsabilizarse de aspectos como la idea del juego, la estética, la historia o la mecánica del mismo, sino que también existía un equipo de trabajo que tenía como función el desarrollo de algoritmos necesarios para llevar tareas como renderizado de los personajes u objetos del juego, simulación de aspectos físicos, detección de colisiones, manejo de eventos y audio del juego entre muchas otras funciones.

Al software que reunía todos estos algoritmos y funcionalidades necesarias para el correcto funcionamiento de un videojuego, se les denominó motor de videojuegos. En este inicio cada juego contaba con su propio motor de videojuego específico para las características y requerimientos de cada proyecto, esto sin dudas hacía que el desarrollo de estas piezas de entretenimiento fuera un proceso demorado y costoso.

Sin embargo con el avance del tiempo y la tecnología, la industria de los videojuegos se ha convertido en una de las más grandes y que más ganancias registran a nivel mundial, tal como lo mencionan (Gonzalez, Montalvo & Palma, 2019) en el año 2016 dicha industria a nivel global creció un 8.5% respecto al año anterior, generando ganancias de alrededor de los 99600 millones de dólares y en publicaciones más

recientes de revistas o sitios web especializados en temas económicos como por ejemplo Forbes México² o en temas de tecnología como FayerWayer³, las ganancias de esta industria en el año 2019 superaron los 120000 millones de dólares.

Teniendo en cuenta las cifras anteriores, para la industria de este sector, se ha hecho completamente necesario el agilizar el proceso de desarrollo de videojuegos y de esta forma poder llegar producir una mayor cantidad de distintos títulos de juegos en menor tiempo. Además, este auge, también implica un mayor interés en este sector industrial por parte de desarrolladores independientes que quieren incursionar en este campo y que no cuentan con los grandes capitales de las empresas ya posicionadas en el mercado.

Para suplir las necesidades de estos sectores, es que surgen distintos motores de juego genéricos que de forma predeterminada ya incluyen todos estos algoritmos y funcionalidades propias de un motor de juego como lo son el renderizado, físicas, detección de colisiones, manejo de audio y eventos del sistema, inteligencia artificial, entre otros. Con esto se logra que cualquier empresa o desarrollador que acceda a uno de estos motores genéricos, pueda simplificar y agilizar el proceso de desarrollo de un videojuego evitando el tener que hacer su propio motor.

En la actualidad existe una gran cantidad de estos motores de juegos genéricos, los cuales pueden tener diferentes características, contextos de uso o público objetivo. Algunos de ellos son los siguientes:

- *Unity 3d*
- *Unreal*
- *GameMaker Studio*
- *Construct*
- *Godot*

Aunque la finalidad principal de estos motores de juegos es el desarrollo de videojuegos, su gran potencia y variedad de herramientas ha logrado que su uso se diversifique y también sean utilizados en el desarrollo de software con propósitos diferentes a los videojuegos. Como es el caso del proyecto al que se refiere este documento, donde se utilizó el motor de juegos Unity junto con el entorno de desarrollo Unity 3D, para facilitar el proceso de implementación de realidad aumentada y el manejo de objetos virtuales 3D entre otras funcionalidades que se implementaron en el sistema.

² Peres Irving. (2019). El valor de la industria de los videojuegos en 2019. Forbes México. Recuperado de: <https://www.forbes.com.mx/el-valor-de-la-industria-de-los-videojuegos-en-2019/>

³ Rodríguez Deivis. (2020). Industrial de los videojuegos arrasó en 2019 con ganancias sobre los USD \$120.000 millones. FayerWayer. Recuperado de: <https://www.fayerwayer.com/2020/01/videojuegos-2019-ganancias/>

2.4. SDK para la implementación de realidad aumentada

En la sección 2.1 del presente documento, se explicó de manera general en que consiste la realidad aumentada y los requisitos a nivel de hardware para su implementación. Sin embargo, además de estos requisitos ya mencionados, también es necesario contar con un software con características específicas para poder llevar a cabo el trabajo de implementación de realidad aumentada en una aplicación.

La característica principal que debe tener un software para la implementación de realidad aumentada es el contar con algoritmos de visión artificial que le permitan al sistema llevar a cabo un proceso de reconocimiento de distintos patrones en imágenes del mundo real para posteriormente, luego de procesar la información, indicar en que puntos específicos de la imagen capturada, se debe mostrar y sobre poner el objeto virtual.

Si bien es cierto, que para cada proyecto de realidad aumentada, el equipo desarrollador puede realizar su propio algoritmo de AR en donde se incluya el sistema de visión artificial anteriormente mencionado. Al igual que como sucede con los motores de videojuegos, lo más común en la actualidad es la utilización de kits de desarrollo de software (SDK) desarrollados por terceros para cumplir la tarea específica de implementación de realidad aumentada y de esta forma facilitar y agilizar la tarea del desarrollador final de la aplicación.

En el mercado existe una gran variedad de estos SDK enfocados a la implementación de proyectos de realidad aumentada. Un ejemplo de esto se encuentra en el estudio de (Cabero & Llorente, 2019), en donde se realiza una evaluación de 16 SDK, plataformas o aplicaciones para la implementación de realidad aumentada, en donde entre otros resaltaron algunos SDK como por ejemplo ARtoolkit, ARmedia y Vuforia.

3. Marco Procedimental

3.1. Diseño de la aplicación móvil

En esta sección del documento, se presentará todo el proceso de diseño de la aplicación móvil que compone este proyecto. Desde la recopilación de información por medio de una encuesta para así determinar algunos parámetros de diseño importantes a tener en cuenta en la aplicación, hasta el desarrollo de un prototipo de alta finalidad del producto final del presente trabajo.

3.1.1. Recopilación de información

Teniendo en cuenta que el objetivo de la aplicación es el apoyo en el proceso de enseñanza-aprendizaje de conceptos básicos de electrónica, se realizó una encuesta virtual⁴ a personas interesadas en el área de la electrónica, con el fin de determinar distintos contextos de uso, parámetros de diseño y la posible aceptación de la aplicación por parte de esta población.

A continuación, se presentan los resultados a las preguntas hechas en la encuesta y posteriormente se hace un breve análisis de los resultados y los parámetros de diseño que se decidieron para la aplicación móvil.

Pregunta 1:

Edad
116 respuestas

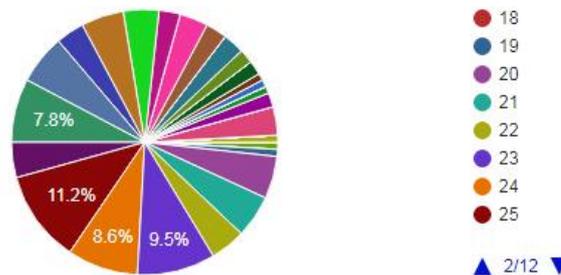


Gráfico 4. Encuesta: Edad

Pregunta 2:

Sexo
116 respuestas

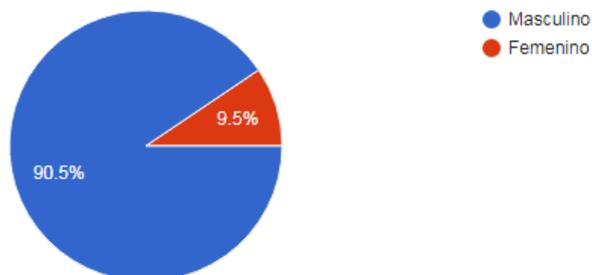


Gráfico 5. Encuesta: Sexo

⁴ Link de encuesta realizada utilizando Google Forms: <https://forms.gle/8UbQccPCBrVvz8pd9>

Pregunta 3:

Ocupación
116 respuestas

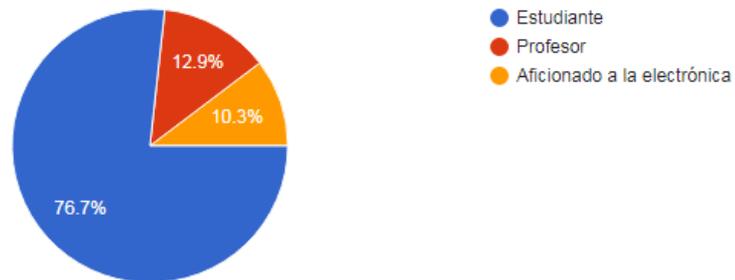


Gráfico 6. Encuesta: Ocupación

Pregunta 4:

Sistema operativo de su dispositivo móvil
116 respuestas

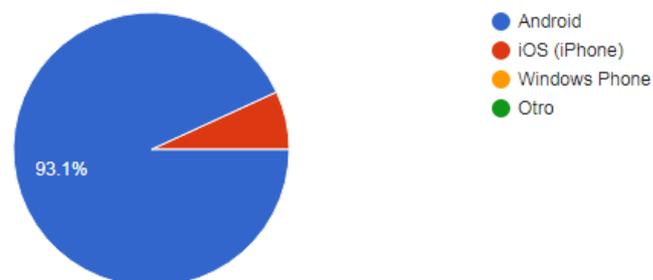


Gráfico 7. Encuesta: Sistema operativo de su dispositivo móvil

En la siguiente sección de preguntas, se indaga a los encuestados por sus preferencias de diseño y funcionalidad en aplicaciones relacionadas con la búsqueda de información o educación.

Pregunta 5:

Interfaz gráfica
116 respuestas

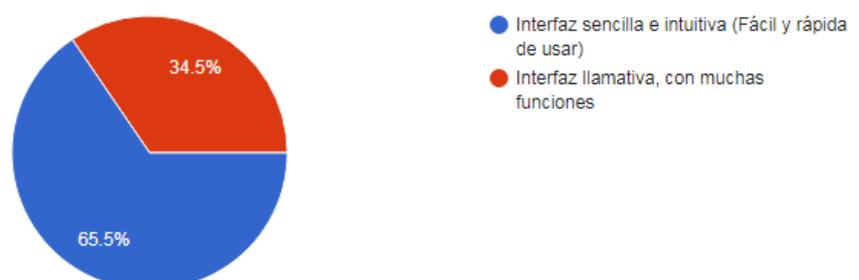


Gráfico 8. Encuesta: Tipo de interfaz preferida

Pregunta 6:

Conectividad

116 respuestas

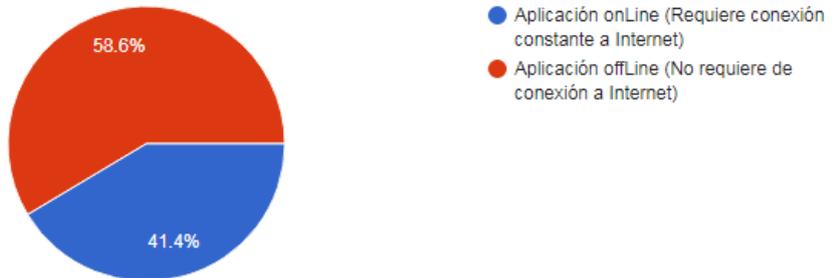


Gráfico 9. Encuesta: Tipo de conectividad preferida

Pregunta 7:

Lugar de uso (¿Generalmente en donde hace uso de este tipo de aplicaciones?). Puede seleccionar varias opciones

116 respuestas

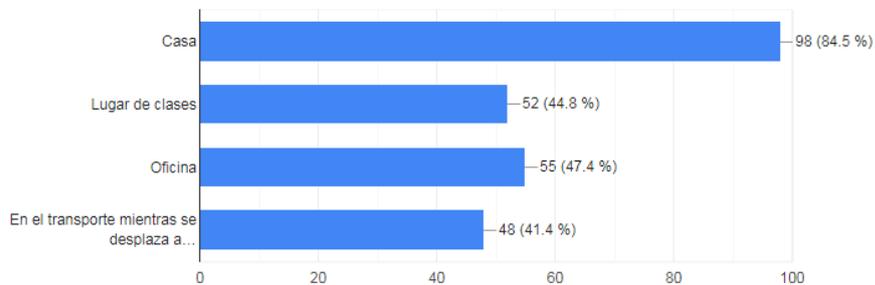


Gráfico 10. Encuesta: Lugar de uso de las aplicaciones

En la siguiente sección de preguntas, se indagó por el uso que los encuestados han hecho de aplicaciones de realidad aumentada.

Pregunta 8:

¿Ha utilizado alguna aplicación de realidad aumentada?

116 respuestas

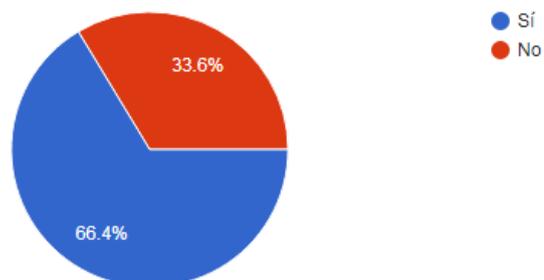


Gráfico 11. Encuesta: Ha usado aplicaciones de realidad aumentada

Pregunta 9:

En caso de haber respondido "sí" a la pregunta anterior. ¿Con que fin las ha utilizado? Puede seleccionar varias opciones

81 respuestas

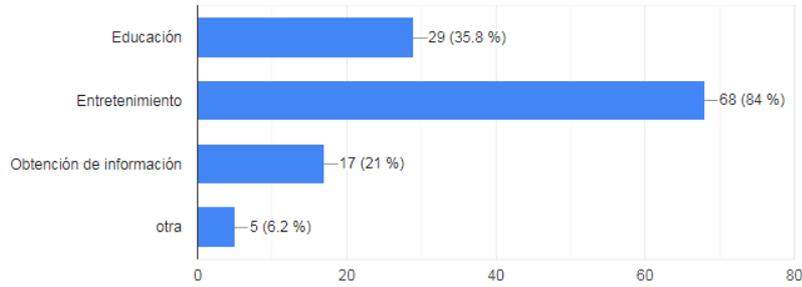


Gráfico 12. Encuesta: finalidad de uso de aplicaciones de realidad aumentada

Pregunta 10:

¿Cree usted que el poder observar y manipular modelos 3D de los dispositivos electrónicos, puede facilitar el proceso de aprendizaje?

116 respuestas

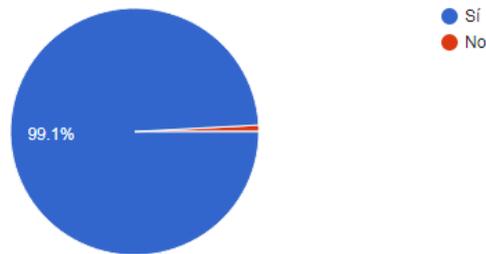


Gráfico 13. Encuesta: ¿Cree que los objetos 3D pueden facilitar el proceso de aprendizaje?

Pregunta 11:

¿Estaría dispuesto a utilizar una aplicación de realidad aumentada como apoyo en su proceso de enseñanza-aprendizaje de la electrónica?

116 respuestas

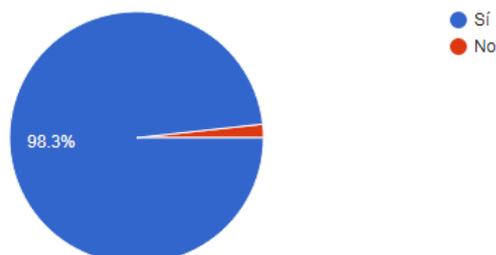


Gráfico 14. Encuesta: Disposición de usar aplicación de realidad aumentada en proceso de enseñanza-aprendizaje

Los anteriores resultados corresponden a las respuestas de 116 encuestados que pertenecen a un segmento de personas interesadas en la electrónica o áreas afines.

En las preguntas 1 a 4, se consultan datos básicos del encuestado, con el fin de determinar a grandes rasgos el público objetivo de la aplicación, a partir de las respuestas obtenidas se pudo establecer lo siguiente:

- El 65.5% de los encuestados corresponden al rango de edad de 20 a 29 años, el 24.1% al rango de edad de 30 a 39 años y el 10.3% restante corresponde a rangos de 40 a 50 años y menores de 20 años.
- El 76.7% de los encuestados son estudiantes, 12.9% profesores y 10.3% aficionados a la electrónica u otra ocupación diferente a las 2 anteriores.
- El 93.1% de los encuestados, cuenta con un dispositivo móvil con sistema operativo Android.

En las preguntas 5 a 7, se consulta sobre preferencias en temas relacionados con la interfaz, conectividad y uso de aplicaciones móviles enfocadas a la educación u obtención de información, de lo cual se obtiene lo siguiente:

- El 65.5% prefiere una interfaz de usuario sencilla y fácil de usar
- El 58.6% prefiere una aplicación offline
- El lugar de uso de este tipo de aplicaciones no se restringe a un solo sitio específico, aunque el uso en casa es el que obtuvo el mayor porcentaje, las otras tres opciones también obtuvieron un porcentaje mayor al 40%. Hay que tener en cuenta que en esta pregunta el encuestado podía seleccionar varias opciones de respuesta.

En las preguntas 8 a 11, se indaga si el usuario ha tenido experiencia de uso con aplicaciones de realidad aumentada, con qué fin las ha utilizado, si cree que la manipulación de modelos 3D puede facilitar el proceso de aprendizaje y si estaría dispuesto a utilizar una aplicación de realidad aumentada como parte de su proceso de enseñanza-aprendizaje. Preguntas de las que se obtuvieron los siguientes resultados.

- El 66.4% ha utilizado en algún momento una aplicación de realidad aumentada.
- De los encuestados que han utilizado este tipo de aplicaciones, el 84% lo ha hecho con fines de entretenimiento.

- El 99.1% de los encuestados cree que la manipulación de modelos 3D pueden facilitar el proceso de enseñanza-aprendizaje.
- El 98.3% de los encuestados estaría dispuesto a usar una aplicación de realidad aumentada como parte de su proceso de enseñanza aprendizaje.

Conclusión a partir de los resultados:

De acuerdo a los resultados presentados, se puede inferir que una aplicación de realidad aumentada como apoyo del proceso de enseñanza-aprendizaje de conceptos básicos de la electrónica, gozaría con un alto grado de aceptación, además que el público dentro del que más aceptación encontraría sería los estudiantes y profesores de esta área del conocimiento en rangos de edades desde los 20 a los 40 años.

Además, se infiere que para estos grupos poblacionales en específico, la realidad aumentada no es algo totalmente nuevo y ya han tenido alguna clase de acercamiento a este tipo de tecnología, razón por la cual no implicaría que se tenga que hacer un gran esfuerzo de adaptación tecnológica de estos usuarios para poder hacer uso de la aplicación.

3.1.2. Parámetros de diseño de la aplicación móvil

A partir de los resultados obtenidos en la encuesta realizada, se decidió que la aplicación móvil debía cumplir con los siguientes parámetros para el diseño y desarrollo de la misma.

- **Interfaz gráfica sencilla e intuitiva:** la interfaz gráfica debe permitir que el usuario interactúe con la aplicación de forma fluida, desde el primer momento de uso, debe ser una interfaz que no cuente con una curva de aprendizaje de uso elevada.
- **Aplicación Off-line:** aunque esto resulte en una aplicación con un tamaño mayor a su contraparte On-Line, la mayor cantidad de usuarios van a ser estudiantes que por diferentes razones puede que no cuenten con conexión estable a internet. Además, al ser una aplicación enfocada a un proceso educativo, se le debe garantizar al estudiante que puede utilizar la aplicación en el momento que lo requiera.

En primera instancia se había propuesto que esta aplicación debería funcionar en conjunto con un libro o cartilla que contendría todas explicaciones teóricas sobre conceptos básicos de electrónica, en este caso tanto aplicación como cartilla formarían parte de un solo producto y el software desarrollado no podría funcionar si el usuario no cuenta con el texto físico producido exclusivamente para este fin.

Sin embargo, se observó que esta propuesta no es conveniente y no significa un aporte relevante en temas como el funcionamiento de la aplicación y la experiencia de usuario.

Además, se tuvo en cuenta lo siguiente:

- En el mercado existe una gran cantidad de libros que abordan los temas de electrónica y que ya han pasado por un estricto proceso editorial, garantizando la calidad del material.
- La simbología de los elementos electrónicos se encuentra normalizada y los libros mencionados anteriormente, así como material disponible en internet, generalmente utilizan esta simbología.
- El público objetivo de la aplicación (personas interesadas en el área de la electrónica) ya cuentan con acceso a libros o material especializado sobre el tema.

Teniendo en cuenta lo anterior, se decidió no realizar el libro o cartilla que se tenía planeado en primera instancia y solamente realizar la aplicación de realidad aumentada, la cual debe ser capaz de reconocer como marcadores a los símbolos normalizados de diferentes elementos electrónicos y sobre estos mostrar en pantalla los modelos 3D de dichos elementos, modelos con los cuales el usuario podrá interactuar en diferentes medidas, dependiendo del componente que se esté mostrando.

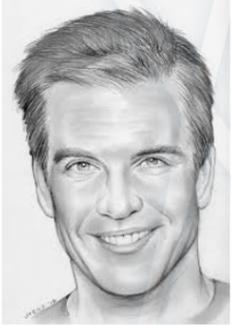
Con esta decisión, se busca aumentar el alcance de la aplicación, permitiendo que más personas que pertenecen al público objetivo, puedan utilizar la aplicación con los libros o material académico de consulta que ya poseen y sin la limitación de verse obligados a adquirir un nuevo libro para poder hacer uso del sistema.

3.1.3. Fichas de usuario

Las fichas de usuario, son elementos que hacen parte de la metodología denominada como diseño centrado en el usuario (DCU), en los cuales se crean personajes ficticios a partir de la observación de los posibles usuarios del producto. Esta herramienta busca agregarle una personalidad a esos posibles usuarios y de esta forma hacer que el diseñador tenga cierto grado de conexión o empatía con ellos.

Para este proyecto se realizaron dos fichas de usuario, las cuales se pueden ver a continuación y son resultado de la observación realizada a las respuestas de la encuesta presentada anteriormente.

Juan Rodríguez



La vida, un constante aprendizaje

Edad: 45
 Trabajo: Profesor
 Familia: Esposa, hijos, padres
 Ubicación: Bogotá D.C
 Carácter: Serio

Serio

Estudioso

Disciplinado

Objetivos

- Actualizar sus conocimientos día a día.
- Lograr que sus estudiantes comprendan los temas explicados, de una forma sencilla y divertida.
- Pasar tiempo con su familia

Necesidades

- Conseguir herramientas o materiales educativos de diferente índole que apoyen el proceso de enseñanza para sus estudiantes.
- Estar al día con el uso de las nuevas tecnologías aplicadas a la educación.

Biografía

Juan es un Licenciado en Electrónica que cuenta con varios años de experiencia como profesor en un colegio técnico de la ciudad de Bogotá, tiene a su cargo varios grupos de estudiantes entre los 12 y 17 años a los cuales les enseña los fundamentos de electrónica y circuitos básicos.

Al finalizar la jornada en el colegio, se dirige a su casa donde dedica unas horas a preparar clases y consultar diferentes materiales educativos que pueda usar en ellas. Finalmente cuando su esposa e hijos llegan del trabajo y colegio respectivamente, dedican este poco tiempo que les queda a compartir en familia.

Motivaciones

Familia 

Miedo 

Crecimiento 

Poder 

Social 

Marcas e influencias






Canales preferidos

Anuncios tradicionales 

Redes sociales 

Artículos de Internet y blogs 

Aplicaciones móviles 

Conocimientos Tecnológicos

Computador 

Dispositivos Móviles 

Ilustración 1. Ficha de usuario 1

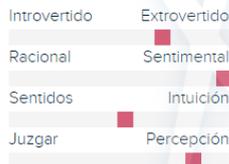
Andrea Jimenez



"No tengo ningún talento especial, solo soy apasionadamente curioso"
Albert Einstein

Edad: 25
Trabajo: Líder proyectos ingeniería
Estudio: Estudiante Ingeniería Mecatrónica.
Familia: Padres, hermanos, novio.
Ubicación: Bogotá D.C.
Carácter: Amable

Personalidad



- Amable
- Inteligente
- Emprendedora
- Curiosa

Objetivos

- Graduarse como ingeniera y destacarse en el campo laboral.
- Crear una empresa de desarrollo de proyectos de ingeniería
- Colaborar a su familia, amigos y personas que lo necesiten en general
- Viajar

Necesidades

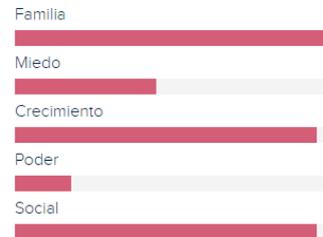
- Información técnica de temas o elementos de ingeniería.
- Acceso a material educativo que contribuya con su proceso de formación.
- Acceso a productos digitales que simplifiquen sus tareas diarias.

Biografía

Andrea es una estudiante de ingeniería Mecatrónica, trabaja en una empresa que lleva a cabo proyectos de ingeniería en el área de las telecomunicaciones. Suele ser una persona alegre y extrovertida que en sus momentos libres disfruta de compartir con su familia, escuchar música, leer o asistir a eventos de entretenimiento.

En la mañana y en la tarde desempeña sus labores dentro de la empresa en la cual labora y luego en la noche se dirige a la Universidad en donde adelanta sus estudios, por esta razón el tiempo del que dispone para dedicarle a sus tareas es limitado y requiere de diferentes materiales de estudio que le ayuden a comprender los temas estudiados en el menor tiempo posible, para así optimizar el tiempo dedicado a estas labores.

Motivaciones



Marcas e influencias



Canales preferidos



Conocimientos Tecnológicos

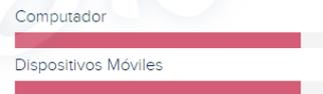


Ilustración 2. Ficha de usuario 2

3.1.4. Contextos de uso del sistema

Teniendo en cuenta las dos fichas de usuario presentadas, se desarrollan dos contextos de uso (uno para cada personaje).

Contexto de uso – caso Juan Rodríguez:

Juan se encuentra preparando las clases del día siguiente, la temática que les va a explicar a sus estudiantes son conceptos y elementos básicos usados en la electrónica. Una vez ya tiene preparada toda la explicación teórica, comienza a pensar de qué manera puede mostrarles cómo funcionan estos elementos de una forma práctica para que entiendan de una forma más fácil y profundizar lo explicado teóricamente.

Indagando en internet y consultando con amigos profesores de electrónica, descubre la aplicación Geek Electrónica AR la cual permite escanear símbolos de elementos electrónicos y por medio de realidad aumentada sobrepone modelos 3D de los dispositivos, de esta forma los estudiantes van a poder manipular los modelos conocer los dispositivos y hasta ver algunas funciones de los mismos.

A Juan le parece interesante la aplicación y decide que la va a utilizar en su clase. Al día siguiente llega al salón y luego de explicar el tema le solicita a los estudiantes que descarguen la aplicación Geek Electrónica AR y les enseña a utilizarla, los estudiantes terminan la clase bastante motivados debido a la experimentación que pudieron hacer con la aplicación. De esta forma Juan decide que va a seguir implementando la aplicación para sus futuras clases.

Contexto de uso – caso Andrea Jimenez:

Andrea llega a la universidad a su clase de instrumentación, es día de laboratorio y tienen que presentar un circuito que el profesor les ha propuesto.

Cuando empieza a revisar los elementos que requiere para realizar el montaje del circuito, se da cuenta que hay varios símbolos que no reconoce, en ese momento recuerda que hace unos días le había llamado la atención una aplicación de realidad aumentada para electrónica y que le podría servir en ese momento.

Toma su celular, abre la aplicación Geek Electrónica AR y escanea los símbolos que no reconocía del diagrama del laboratorio propuesto, aprovecha para revisar su funcionamiento y ya teniendo más claros todos los conceptos, se dispone a realizar su trabajo. Al terminar sus demás compañeros aún no habían terminado porque al igual que ella, no se acordaban de algunos símbolos y se demoraron más tiempo buscando la información, Andrea al darse cuenta de eso les recomienda la aplicación que ella utilizó y que le ayudo a obtener la información de forma más rápida y con buenos resultados.

3.1.5. Diagrama general de casos de uso

Los diagramas UML de caso uso de uso, son una forma simplificada de representar las diferentes partes que componen un sistema informático y como son las interacciones que tiene el usuario con cada una de estas partes o incluso las interacciones entre ellas mismas. A continuación se presenta el diagrama UML de casos de uso para la aplicación Geek Electrónica AR.

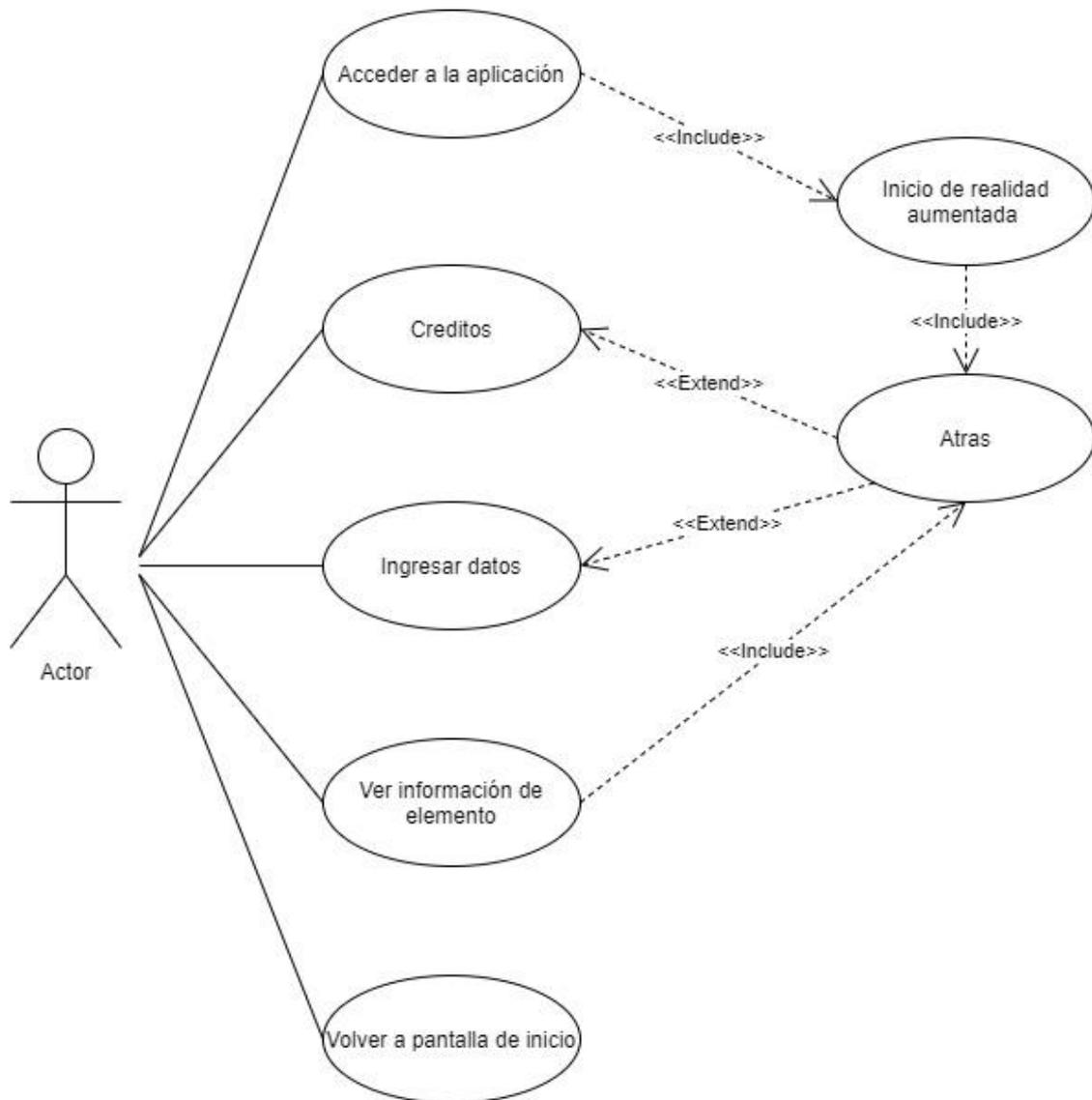


Gráfico 15. Diagrama UML de casos de uso de aplicación Geek Electrónica AR

3.1.6. Diagrama de flujos de interacción del sistema

Los diagramas de flujo de interacción del sistema, representan de forma básica el recorrido de un usuario por el sistema dependiendo de la forma que él interactúe con la aplicación. En el siguiente gráfico, se puede observar el recorrido básico que un usuario de la aplicación experimentaría, de acuerdo a las decisiones tomadas o a las detecciones que el sistema haga de los marcadores de realidad aumentada.

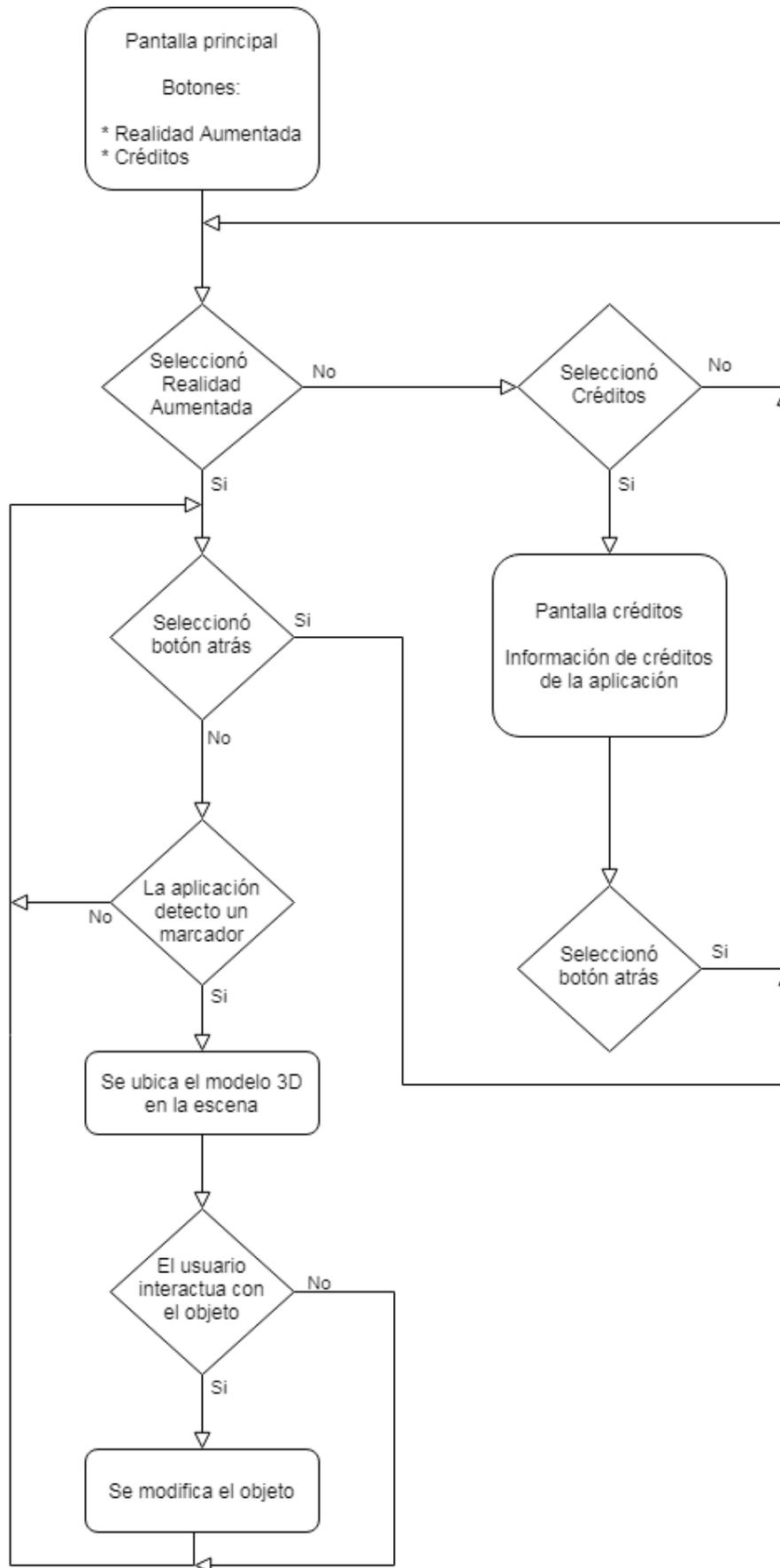


Gráfico 16. Diagrama de flujo de interacción del sistema

3.1.5. Prototipo de alta fidelidad

Los prototipos de alta fidelidad, son una representación gráfica con alto detalle de lo que va a ser la presentación final del producto, suelen incorporar interactividad con el usuario, con esto se observa cual es la impresión del usuario al producto y de esta forma se pueden corregir los últimos detalles antes de continuar con la fase de desarrollo.

Para este proyecto se utilizó como herramienta de prototipado MarvelApp, el cual es un producto en línea y se destaca por la facilidad de uso.

Para observar el funcionamiento de este prototipo se puede hacer por medio del siguiente link: <https://marvelapp.com/61cc2b0>

A continuación se presentan y describen las pantallas que componen al prototipo de alta fidelidad de esta aplicación.

Pantalla principal:

En ella se presenta el logo de la aplicación y dos botones de navegación con los cuales el usuario puede decidir si quiere seguir al modo de realidad aumentada de la aplicación o si quiere observar los créditos de la misma.

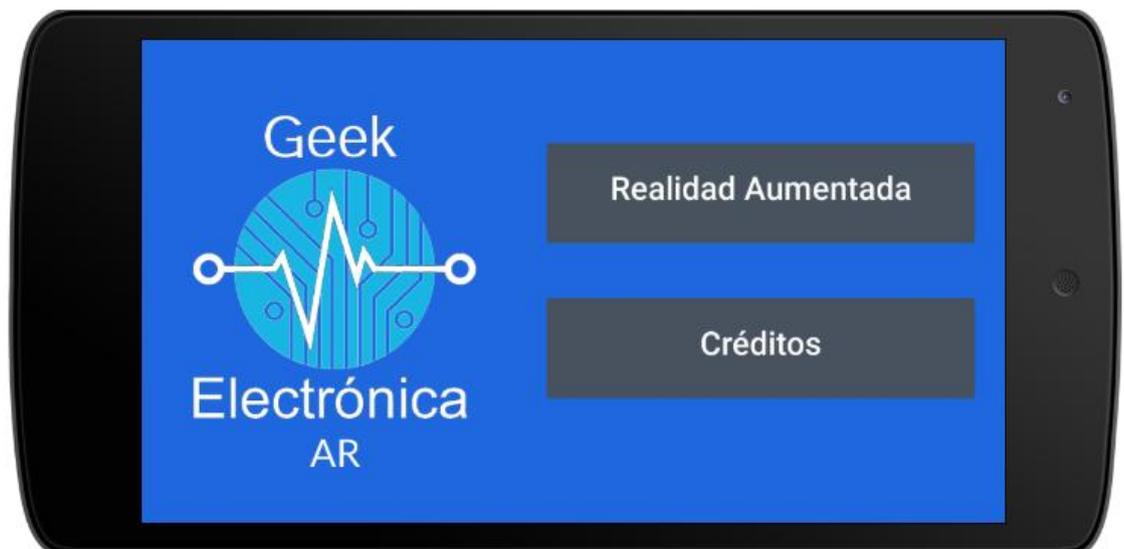


Ilustración 3. Pantalla principal de la aplicación

Pantalla realidad aumentada:

En ella se presenta el modo principal de la aplicación y se activa la cámara del dispositivo móvil, la aplicación escanea en busca de un marcador de realidad aumentada valido y en caso de encontrarlo muestra el modelo 3D del elemento electrónico y sus respectivas opciones de manipulación.

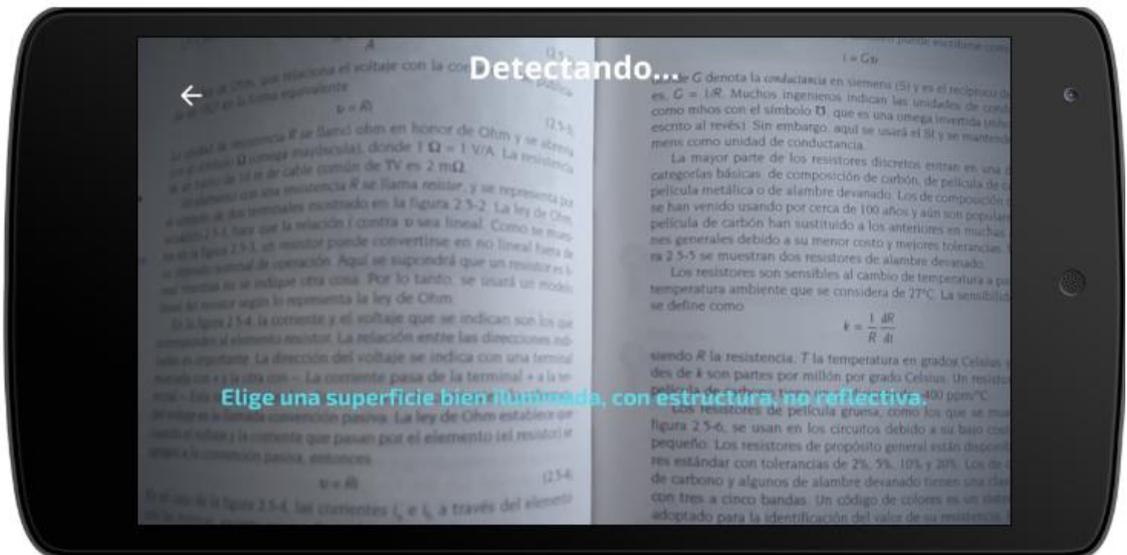


Ilustración 4. Pantalla de realidad aumentada - Detectando marcador

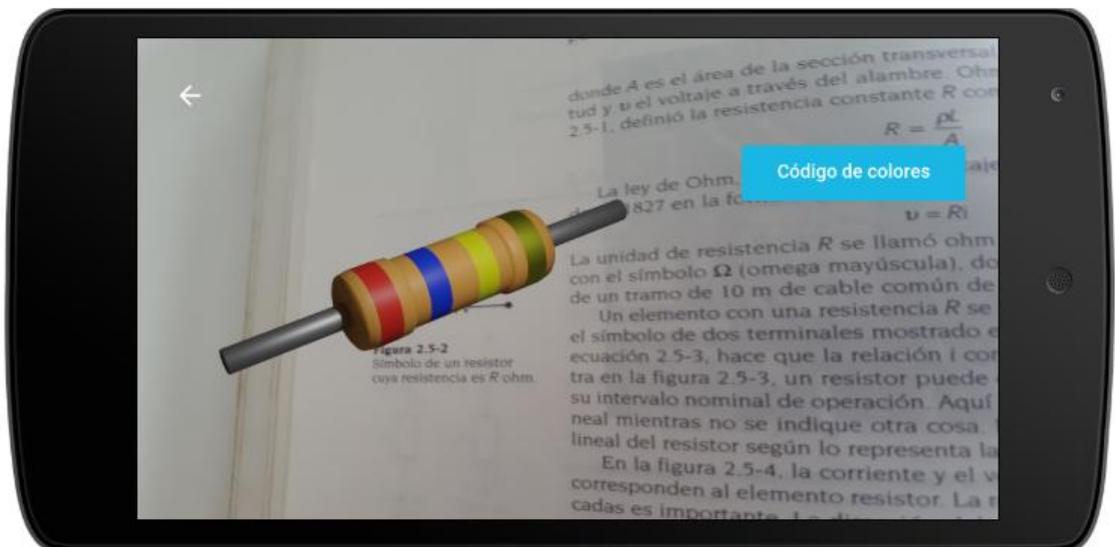


Ilustración 5. Pantalla de realidad aumentada - Modelo 3D sobrepuesto al marcador

Pantalla de funcionalidad por elemento:

Cada uno de los elementos electrónicos disponibles en la aplicación y que aparecen al ser escaneado un marcador válido, tiene una funcionalidad diferente con la que el usuario podrá explorar el funcionamiento o como está compuesto dicho dispositivo. En la siguiente imagen, se puede observar la funcionalidad del modelo de resistencia eléctrica la cual es una aplicación del código de colores de resistencias.

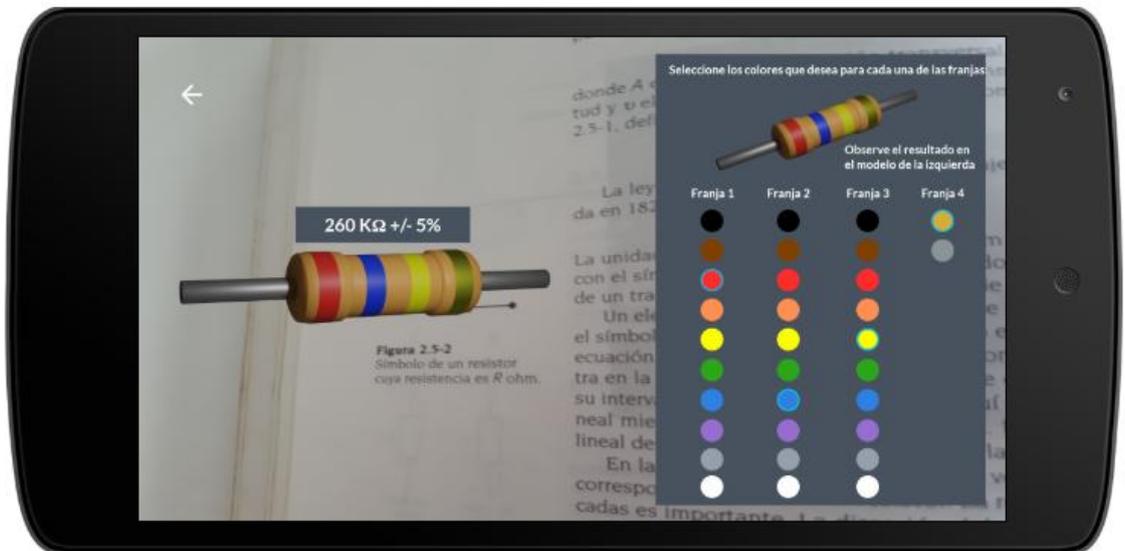


Ilustración 6. Pantalla funcionalidad del modelo resistencia

Pantalla de créditos:

En esta pantalla, se presentan los créditos de los recursos editados, si son recursos propios o recursos de terceros (siempre y cuando estén bajo una licencia que permitan su uso).

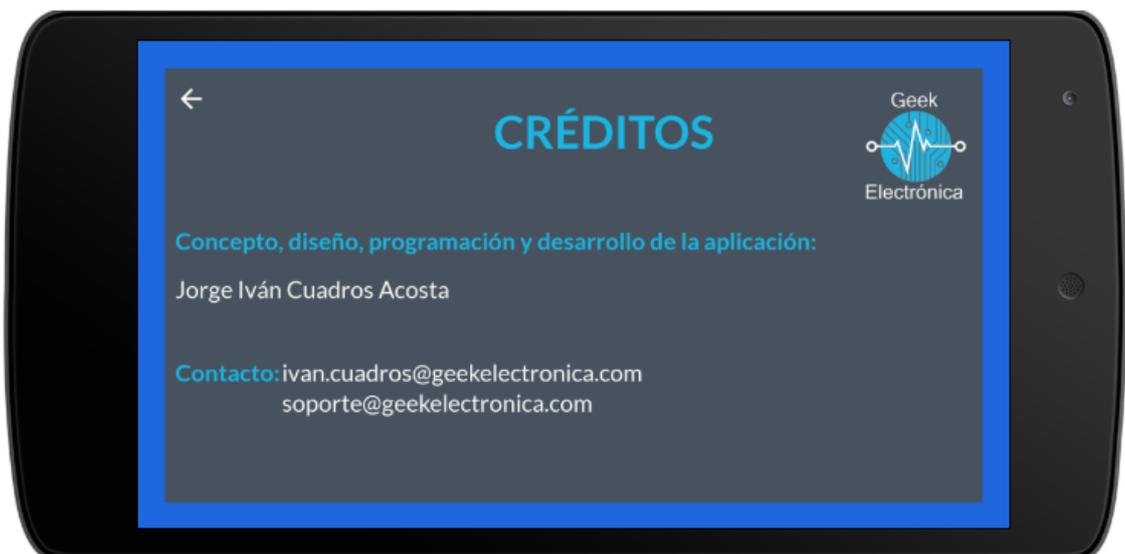


Ilustración 7. Pantalla de créditos de la aplicación

3.2. Implementación de la aplicación móvil

En esta sección del documento, se presenta todo el proceso de desarrollo e implementación de la aplicación móvil propuesta. Desde el proceso de diseño y modelado de los objetos 3D a utilizar, hasta la obtención final de la aplicación que se entrega en conjunto con este documento y objetivo principal del presente trabajo.

3.2.1. Desarrollo de modelos 3D

Geek Electrónica AR, al ser una aplicación que incorpora realidad aumentada dentro de su funcionamiento, requiere que el aspecto gráfico de los elementos a utilizar sea cuidado de tal forma que los objetos que se muestren dentro de la aplicación se vean lo más reales posibles enriqueciendo la experiencia de realidad aumentada.

Además, al ser una aplicación enfocada al aprendizaje de conceptos básicos de electrónica, entre más parecidos sean los objetos virtuales a los objetos reales, el usuario obtendrá una mejor experiencia ya que podrá asociar rápidamente el objeto virtual con el real y asimilar el concepto abordado en la aplicación para cada elemento electrónico expuesto.

Para este fin, se optó por desarrollar modelos 3D propios, que se ajustaran a las necesidades propias del proyecto, en temas de formas, materiales e incluso partes o componentes de cada uno de los objetos.

El software utilizado para llevar a cabo el modelado 3D de cada uno de los elementos fue Blender, el cual es un programa de uso libre y código abierto. La selección de este programa informático para llevar a cabo el trabajo de modelado estuvo dada por las siguientes características:

- Es un programa de uso libre y código abierto.
- Amplia documentación y guías de desarrollo.
- No requiere de grandes especificaciones de hardware para su correcto funcionamiento.
- Fácil exportación e integración de sus modelos con el entorno de desarrollo Unity 3D.

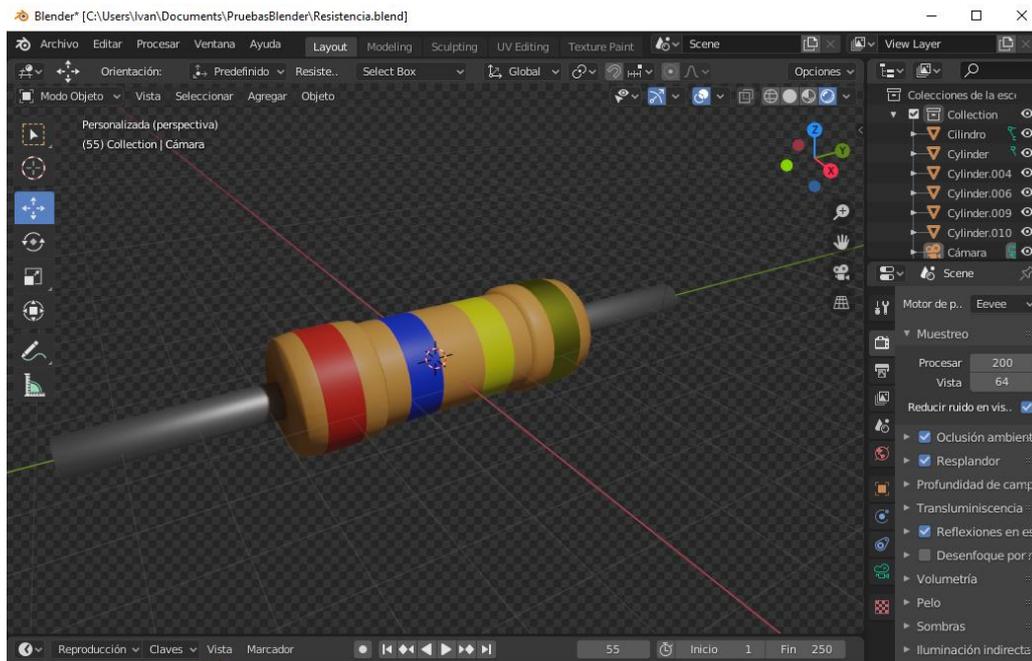


Ilustración 8. Entorno de software Blender. Desarrollo del modelo de una resistencia

Para esta aplicación se desarrollaron cinco modelos 3D de elementos electrónicos que se utilizan de forma frecuente durante el proceso de aprendizaje de conceptos básicos de electrónica, dichos modelos son resistencia, condensador electrolítico, diodo rectificador, display 7 segmentos y motor DC. En las siguientes imágenes se puede observar el resultado del modelado de estos componentes en Blender.

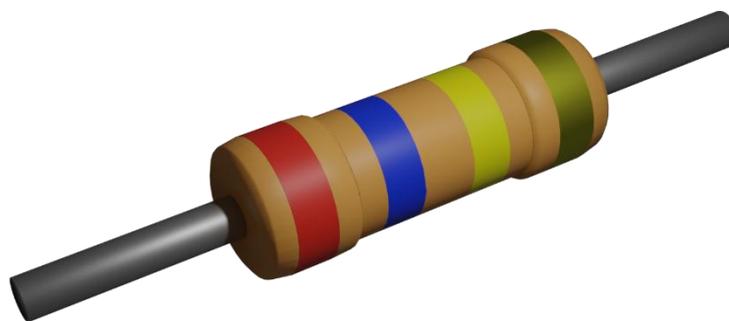


Ilustración 9. Resultado del renderizado del modelo 3D de la resistencia

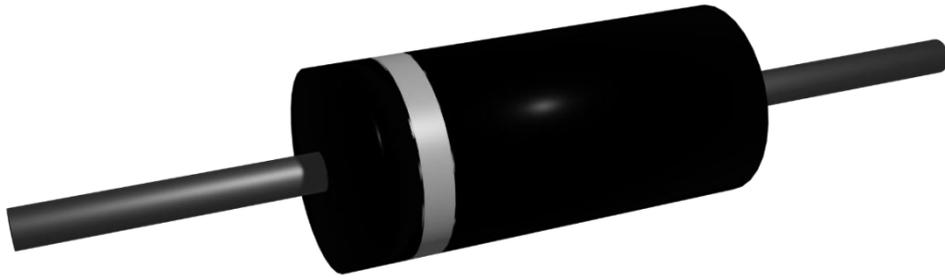


Ilustración 10. Resultado del renderizado del modelo 3D del diodo rectificador



Ilustración 11. Resultado del renderizado del modelo 3D del condensador

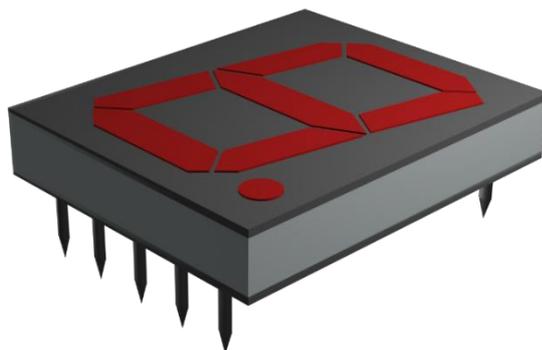


Ilustración 12. Resultado del renderizado del modelo 3D del display 7 Segmentos

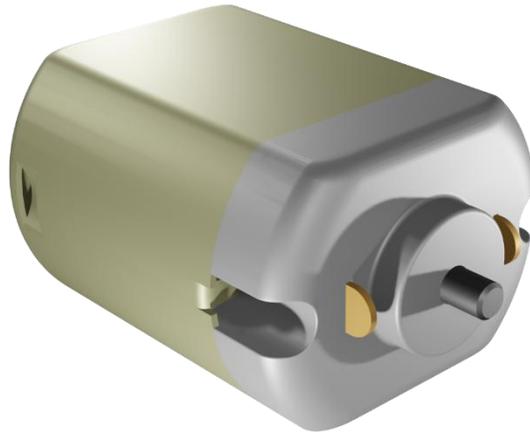


Ilustración 13. Resultado del renderizado del modelo 3D del motor (Armado)

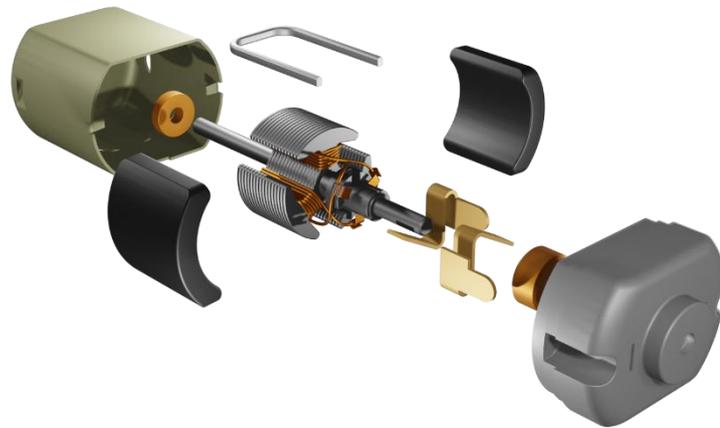


Ilustración 14. Resultado del renderizado del modelo 3D del motor (Desarmado)

3.2.2. Implementación de SDK de realidad aumentada

Para proceso de implementación de realidad aumentada, se decidió utilizar el SDK de vuforia con una licencia de desarrollador, la cual permite el uso de este SDK de manera gratuita con algunas restricciones como lo son por ejemplo el no poder publicar y distribuir la aplicación y limitaciones de almacenamiento en nube. Restricciones que para el alcance del presente proyecto no significan ningún problema.

Algunas razones por las que se optó por el uso de este SDK son las siguientes:

- Licencia desarrollador gratuita
- Amplia documentación
- Sencillez de uso en cada una de las etapas de implementación de realidad aumentada.
- Compatibilidad y fácil integración con el entorno de desarrollo Unity 3D.

Creación de marcadores de realidad aumentada:

Teniendo en cuenta que los objetivos a reconocer son los símbolos ANSI de los componentes electrónicos que se desarrollaron en Blender (a excepción del display siete segmentos que se explicara más adelante), se tomó como patrón dichos símbolos para realizarlos completamente en un software de edición de imágenes como lo es GIMP. Como resultado se obtuvieron las siguientes cinco imágenes que se pueden observar en la siguiente ilustración.

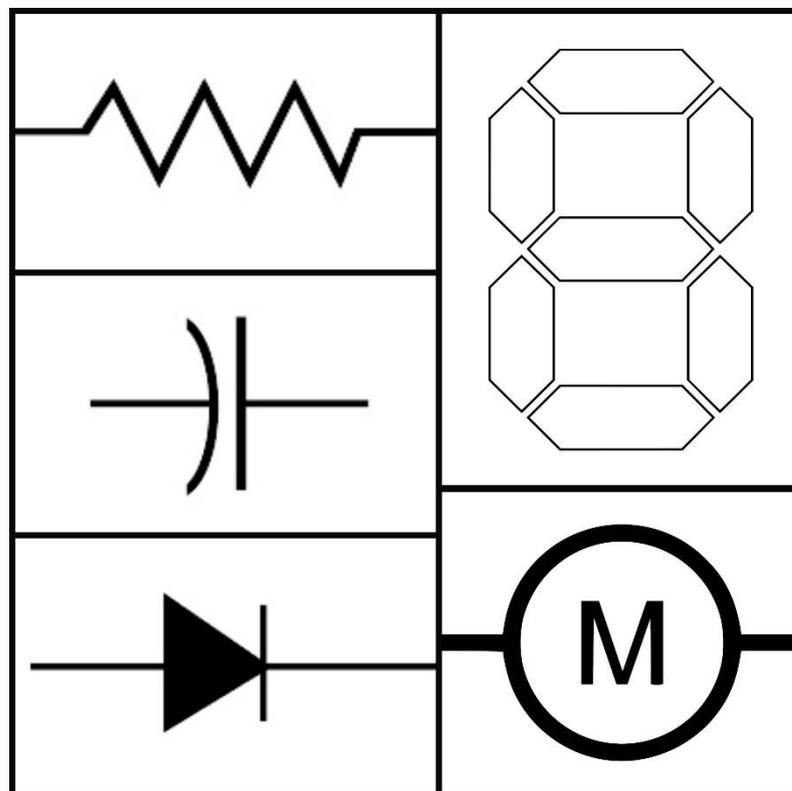


Ilustración 15. Imágenes diseñadas como marcadores de realidad aumentada

Para el caso del display siete segmentos que no cuenta con un símbolo ANSI claramente especificado, se optó por además generar dos marcadores adicionales al ya presentado en la ilustración anterior, con esto se busca ampliar la cantidad de patrones que el usuario pueda escanear para la posterior visualización del modelo 3D del display siete

segmentos. Los dos marcadores adicionales son los que se muestran en la siguiente ilustración.

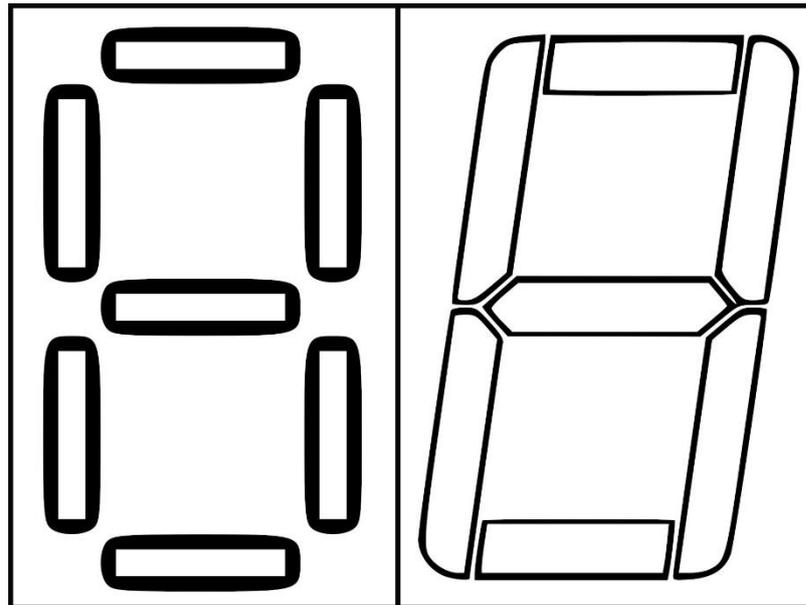


Ilustración 16. Imágenes adicionales generadas para el display siete segmentos

Luego de este proceso se procedió a generar una licencia y una base de datos en el sitio web de vuforia, en dicha base de datos se procedió a generar cada uno de los marcadores necesarios con las imágenes presentadas anteriormente.

GeekElectronicaAR [Edit Name](#)
Type: Device

Targets (8)

Add Target

<input type="checkbox"/>	Target Name	Type	Rating ⓘ	Status ▾
<input type="checkbox"/>	 Display7SegTarget3	Single Image	★★★★★	Active
<input type="checkbox"/>	 Display7SegTarget2	Single Image	★★★★★	Active
<input type="checkbox"/>	 Display7SegTarget1	Single Image	★★★★★	Active
<input type="checkbox"/>	 MotorTarget1	Single Image	★★★★★	Active
<input type="checkbox"/>	 CapacitorTarget2	Single Image	★★★★★	Active
<input type="checkbox"/>	 DiodeTarget1	Single Image	★★★★★	Active
<input type="checkbox"/>	 CapacitorTarget1	Single Image	★★★★★	Active
<input type="checkbox"/>	 ResistorTarget1	Single Image	★★★★★	Active

Ilustración 17. Base de datos creada con marcadores creada en vuforia

El sistema automáticamente detecta puntos específicos de cada imagen para establecerlos como puntos de marcado y los cuales servirán como comparación con los objetivos que el usuario escanee posteriormente,

de esta forma establece cuál de los marcadores ha sido detectado y posteriormente muestra el modelo 3D correspondiente a dicho marcador.

ResistorTarget1

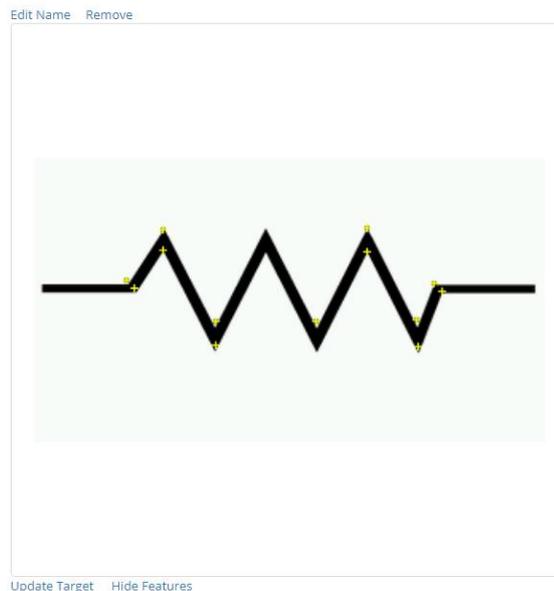


Ilustración 18. Marcador de resistencia con puntos detectados por el sistema de vuforia

Luego de completar la base de datos con todos los marcadores que requiere la aplicación, se procede a descargar la base de datos, obteniendo un paquete de Unity que posteriormente se agrega al proyecto.

En el anexo 6.1 del presente documento puede encontrar muestras de marcadores de los distintos elementos electrónicos, con los cuales puede hacer pruebas de la aplicación.

Integración del SDK de vuforia con el entorno Unity 3D:

Como se mencionó anteriormente, una de las razones por la que se escogió el SDK de vuforia para este proyecto, además de la amplia documentación, fue la facilidad de integración con el entorno de desarrollo Unity 3D. Esto es debido a que solo bastan pocas configuraciones en el entorno de desarrollo para poder implementar una aplicación de realidad aumentada con vuforia.

A continuación se describen los pasos de configuración para llevar a cabo dicho procedimiento.

1. En el proyecto de Unity 3D que en donde se va a utilizar realidad aumentada, abrir el Project Settings, en la sección se Player ubicar la opción XR Settings y habilitar la opción “Vuforia Augmented Reality Supported”.

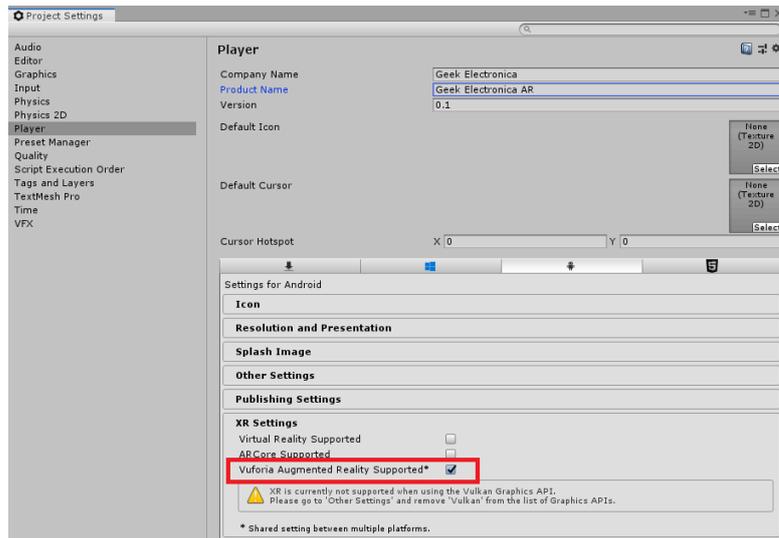


Ilustración 19. Habilitación del soporte a realidad aumentada con vuforia en Unity 3D

2. En la opción Other Settings y Graphics APIs, seleccionar la opción Vulkan y eliminarla. Este proceso es requerido ya que Vulkan Graphics genera conflictos con el SDK de vuforia utilizado para la realidad aumentada.

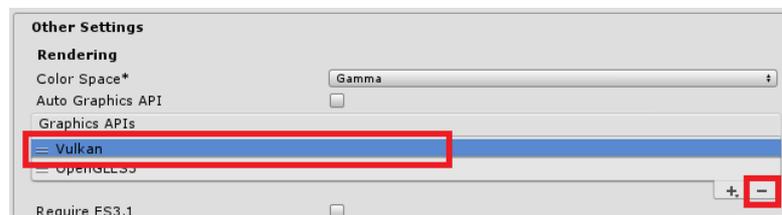


Ilustración 20. Eliminar Vulkan Graphics de las configuraciones del proyecto

3. A partir de este punto ya se puede hacer uso de las herramientas del SDK de vuforia para la realización de proyectos con realidad aumentada.

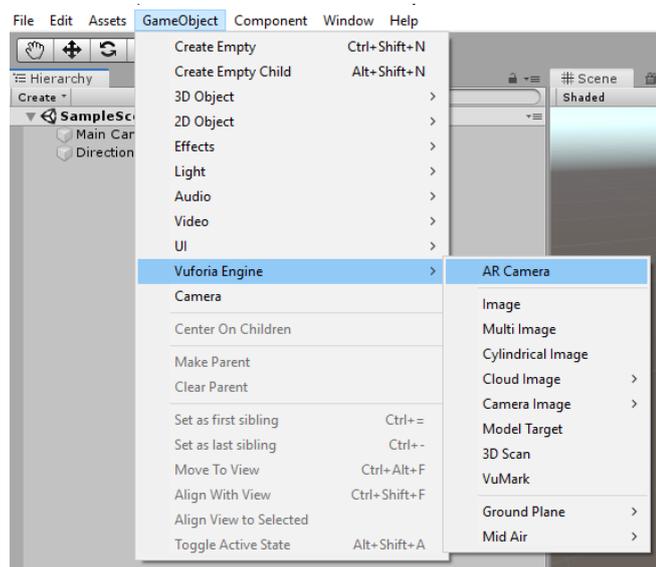


Ilustración 21. Herramientas del SDK de vuforia en Unity 3D

4. Para importar la base de datos que contiene los marcadores y que fue descargada previamente. Basta con hacer doble click (teniendo el proyecto de unity abierto), posteriormente se presenta un cuadro de dialogo mostrando los elementos que se van a importar y solicitando la verificación de la importación.

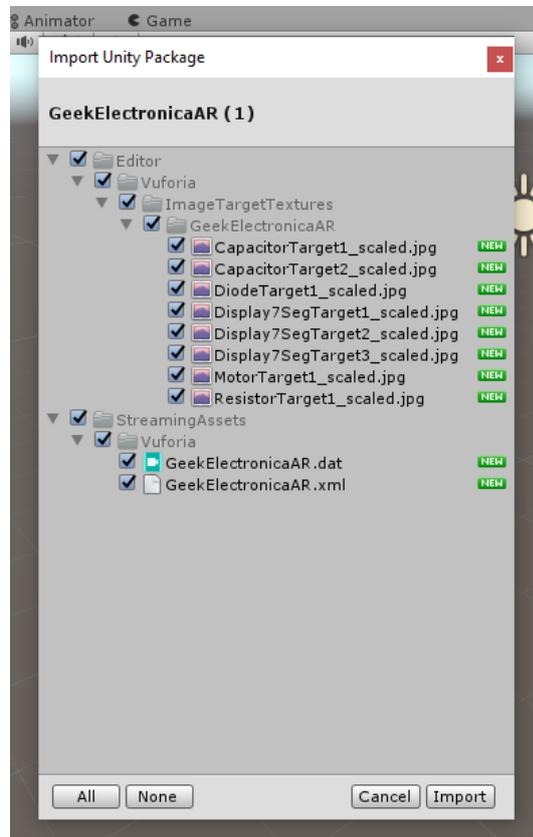


Ilustración 22. Importación de base de datos con los marcadores AR

3.2.3. Desarrollo de la aplicación Geek electrónica AR

En esta sección del documento se presenta el proceso de desarrollo de la aplicación denominada Geek Electrónica AR, teniendo en cuenta tanto los aspectos gráficos implementados dentro del entorno de desarrollo y que ya se habían planteado en el diseño de la aplicación y prototipado de alta fidelidad, hasta los aspectos relacionados con el código de programación generado para esta aplicación para su correcto funcionamiento.

Para el desarrollo de la aplicación se crearon en total ocho escenas diferentes, la cual cada una corresponde a una pantalla de la aplicación y tiene diferentes funciones, así mismo se generaron once scripts de programación en lenguaje C#, los cuales controlan desde la transición entre escenas hasta la modificación de diferentes parámetros de los objetos que participan en cada una de ellas.

A continuación se describe más a fondo cada uno de estos aspectos de acuerdo a cada una de las escenas presentes en la aplicación.

Escena Inicio:

Es la pantalla de presentación de la aplicación, tiene un aspecto gráfico sencillo que consta del logo de la aplicación y dos botones que redirigen al usuario a dos escenas distintas, la escena AR y la escena Credits.



Ilustración 23. Aspecto escena Inicio

Para realizar la transición entre escenas se desarrolla una clase llamada "SceneChanger", la cual contiene un método que es invocado por medio del evento onClick de cada uno de los botones.

Dicho método recibe como parámetro el nombre de la escena a la que se debe redirigir la aplicación, posteriormente hace una llamada a una corutina que se encarga de mostrar un spinner de carga para simbolizar que se está realizando el proceso y por último carga la escena seleccionada por el usuario.

```
public class SceneChanger : MonoBehaviour
{
    IEnumerator LoadIndicator(string sceneName)
    {
        Handheld.SetActivityIndicatorStyle(AndroidActivityIndicatorStyle.Large);

        Handheld.StartActivityIndicator();
        yield return new WaitForSeconds(0);
        SceneManager.LoadScene(sceneName);
    }

    public void ChangeSceneTo(string sceneName)
    {
        StartCoroutine(LoadIndicator(sceneName));
    }
}
```

Ilustración 24. Muestra del código realizado para la transición entre escenas

Escena Credits:

A esta escena solo es posible llegar desde la pantalla de presentación de la aplicación. Al igual que la escena anterior, tiene un aspecto gráfico sencillo en donde se presenta la información del desarrollador de la aplicación incluyendo correos de contacto y un botón atrás para regresar a la pantalla de presentación de la aplicación.



Ilustración 25. Aspecto escena Credits

Respecto a código de programación utilizado para esta escena, se utilizó la misma clase “SceneChanger” que se relacionó en la descripción de la escena anterior para realizar la transición de retorno a la escena principal.

Escena AR:

En esta escena se lleva a cabo el proceso de reconocimiento de marcadores y presentación de los modelos virtuales 3D en realidad aumentada. Para esta escena se tienen dos presentaciones diferentes dependiendo si el sistema ha reconocido o no un marcador válido.

Si no se ha reconocido un marcador válido, en pantalla solo se puede observar la visualización obtenida por la cámara AR de vuforia y un botón de atrás el cual permite regresar a la pantalla de presentación de la aplicación.

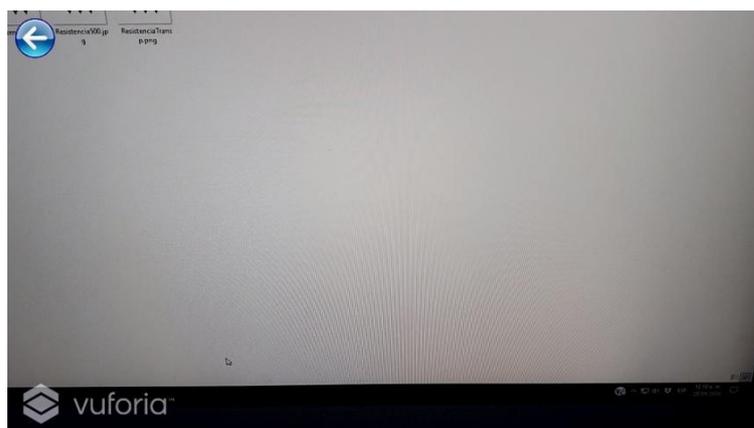


Ilustración 26. Aspecto escena AR sin detección de marcador

En caso que el sistema detecte algún marcador de realidad aumentada válido, se puede observar en pantalla el modelo virtual 3D correspondiente al marcador detectado y un botón en la parte superior derecha el cual tiene como función redirigir al usuario a la escena específica que contiene la aplicación de cada elemento.

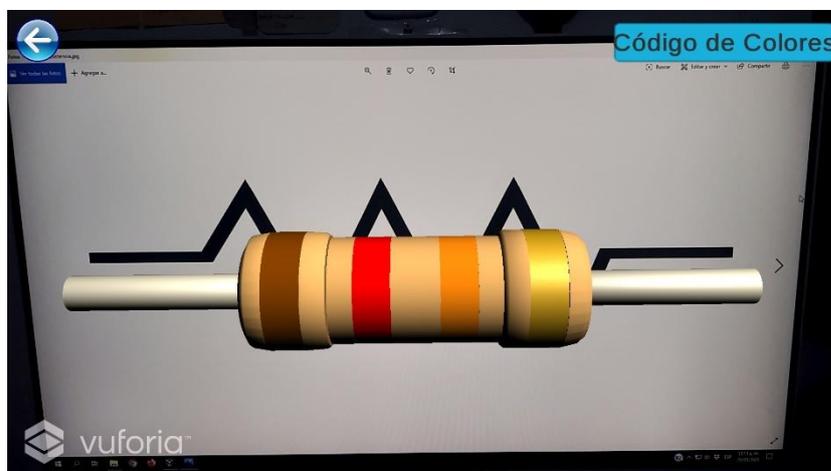


Ilustración 27. Aspecto escena AR con marcador de resistencia detectado

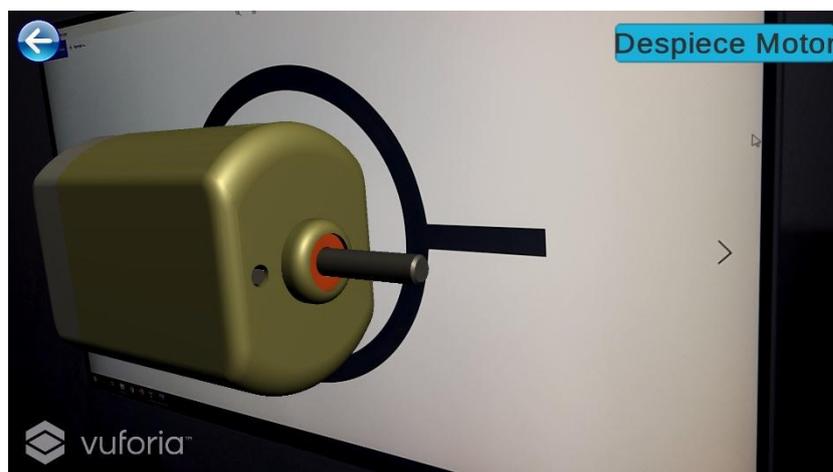


Ilustración 28. Aspecto escena AR con marcador de motor detectado

Para obtener el funcionamiento descrito, cada uno de los modelos virtuales 3D tienen un componente "Mesh Renderer" el cual se activa cada vez que el modelo 3D se visualiza en pantalla y se desactiva cuando se pierde la visualización del marcador y por tanto del objeto virtual.

Esta activación o desactivación del Mesh Renderer de cada objeto, es evaluado por medio de la clase llamada "ShowMenu" que dependiendo

del objeto que es mostrado en pantalla, cambia el texto del botón y habilita la visualización del mismo.

Cuando el usuario oprime el boton superior derecho que redirige a la pantalla que contiene la aplicación realizada para cada elemento, por medio del evento `OnClick` se hace el llamado al método `ChangeSceneToApplication` que hace parte de la clase llamada "SceneChangerToApplication". El objetivo de este método es en primera instancia capturar el texto del botón para con ese valor redirigir a la escena correspondiente para cada modelo 3D.

```
public void ChangeSceneToApplication()
{
    Scene = textButton.text;

    switch(Scene)
    {
        case "Código de Colores":
            Scene = "Resistor";
            break;
        case "Carga del condensador":
            Scene = "Capacitor";
            break;
        case "Diodo Rectificador":
            Scene = "Diode";
            break;
        case "Display 7 Segmentos":
            Scene = "Display7Seg";
            break;
        case "Despiece Motor":
            Scene = "Motor";
            break;
    }

    SceneManager.LoadScene(Scene);
}
```

Ilustración 29. Muestra del código realizado para la transición de la escena AR a la escena correspondiente para cada elemento electrónico

En el siguiente gráfico, se muestra el diagrama de clases para la escena AR en el cual se representan las distintas clases que intervienen en la escena y su relación entre ellas.

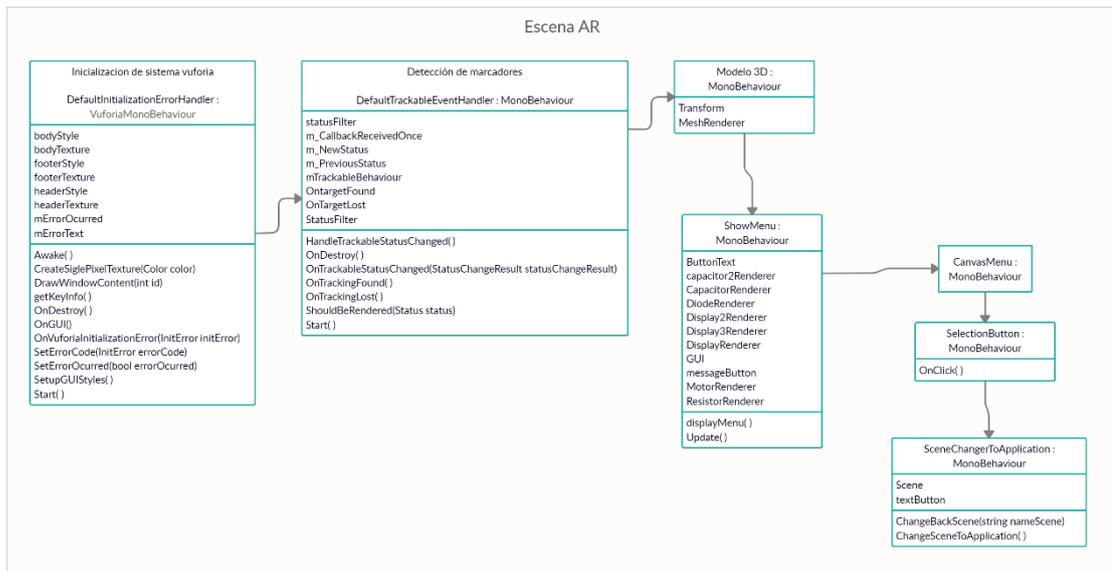


Gráfico 17. Diagrama de clases escena AR

Escena Resistor:

En esta escena se presenta la aplicación desarrollada para dispositivo electrónico llamado resistencia, la cual consiste en una herramienta que sirve como ayuda para comprender el código de colores con el que se determina el valor en ohmios de una resistencia.

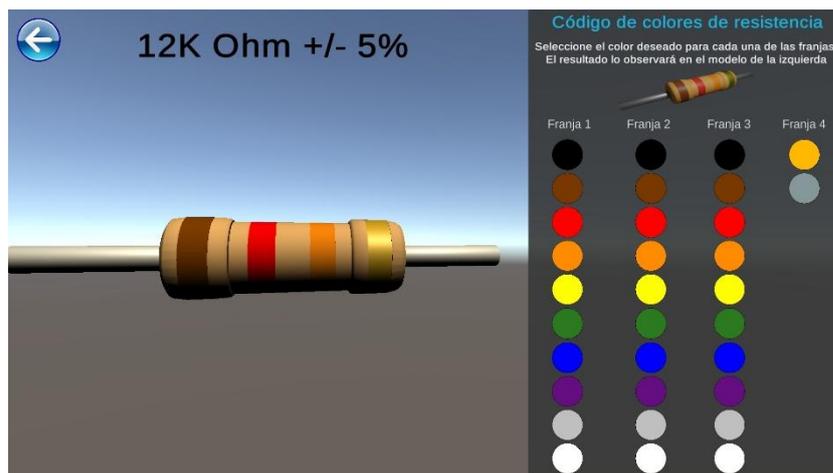


Ilustración 30. Aspecto de la escena Resistor

En la esquina superior izquierda, se encuentra un botón atrás el cual redirige al usuario a la pantalla de realidad aumentada nuevamente. Además de esto la pantalla se divide en dos grandes secciones con elementos y funcionalidades diferentes (Diseño que se mantiene constante para cada una de las aplicaciones de los distintos elementos).

Sección izquierda de la pantalla:

En esta sección se muestra el modelo virtual de una resistencia, el usuario puede interactuar con dicho modelo a través de gestos táctiles

en su dispositivo móvil con lo cual realizara la rotación del modelo, esta opción permite que el usuario pueda reconocer el dispositivo electrónico desde distintos ángulos y se familiarice con su aspecto físico.

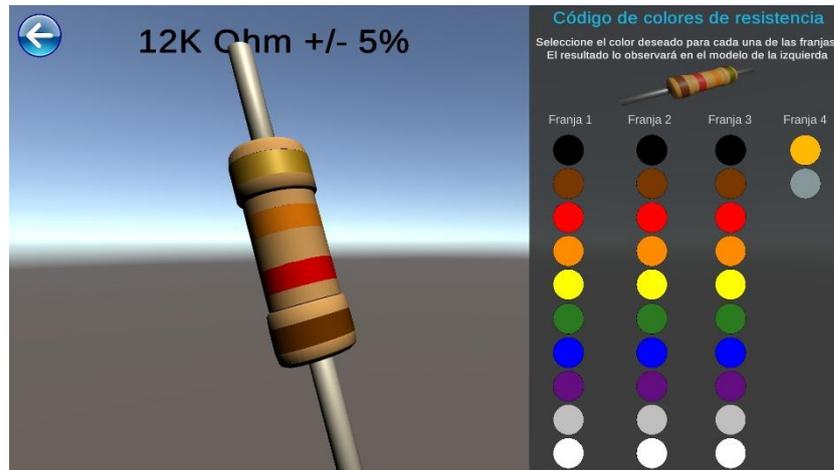


Ilustración 31. Aspecto de la resistencia luego de rotar el modelo

La rotación del modelo 3D se realiza mediante la clase llamada “RotateComponent” la cual es agregada al modelo en cuestión y su función es determinar los cambios de valor en las coordenadas en las que se realiza el gesto táctil en pantalla para posteriormente modificar el valor del atributo rotate del componente transform del modelo 3D.

```
transform.Rotate(Vector3.forward * speed * Time.deltaTime);
```

Ilustración 32. Muestra de modificación del atributo rotate del componente transform del modelo 3D

En esta misma sección izquierda, en la parte superior se encuentra un texto que indica el valor actual de la resistencia para la combinación de colores que se presenta en el modelo y el cual se verá afectado dependiendo de las selecciones que haga el usuario en la sección derecha de la pantalla.

Sección derecha de la pantalla:

En esta sección se puede encontrar un panel que contiene en la parte superior un texto que tiene la función de instructivo para el usuario y luego se encuentran una serie de botones presentados como círculos de colores con los cuales el usuario puede modificar el color de cada una de las franjas del modelo 3D.

Cada vez que el usuario oprime uno de los botones presentes en el panel derecho, por medio del evento OnClick del botón, se hace el llamado al método setColor de la clase llamada “ChangeResistorValue”.

Dicho método recibe como parámetro el número correspondiente a cada color y dependiendo del número obtenido, modifica el valor del atributo color para la franja correspondiente.

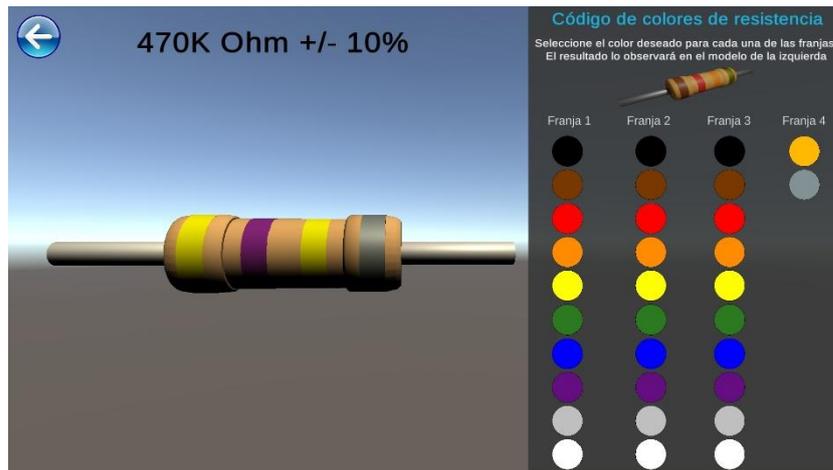


Ilustración 33. Aspecto de la resistencia luego de modificar sus valores

Los diferentes colores con los que se modifica el atributo color de cada una de las franjas, se encuentran pre-establecidos en un arreglo mediante una configuración RGB.

```
Color[] colores = { new Color(0,0,0,1), //Negro
                   new Color(0.5019f,0.2519f,0,1), //Cafe
                   new Color(1,0,0,1), //Rojo
                   new Color(1,0.5019f,0,1), //Naranja
                   new Color(1,1,0,1), //Amarillo
                   new Color(0.2039f,0.5647f,0.1882f,1), //Verde
                   new Color(0,0,1,1), //Azul
                   new Color(0.5450f,0.1568f,0.5764f,1), //Violeta
                   new Color(0.7921f,0.8f,0.8078f,1), //Gris
                   new Color(2,2,2,1), //Blanco
                   new Color(0.9372f,0.7215f,0.0627f,1), //Dorado
                   new Color(0.5411f,0.5843f,0.5921f,1)};
```

Ilustración 34. Arreglo que contiene todos los colores del código de colores de resistencias

```
renderFranja1.material.color = colores[color];
```

Ilustración 35. Muestra de modificación del atributo color para la franja 1 de la resistencia

Además, la clase “ChangeResistorValue”, también se encarga de modificar el texto del valor actual de la resistencia que se encuentra en la parte superior de la sección izquierda antes mencionada.

Esta pantalla donde se muestra la aplicación para cada dispositivo electrónico, presentó una modificación respecto al prototipo de alta fidelidad, ya que en dicho prototipo se utiliza como fondo la visualización de la imagen obtenida por la cámara del dispositivo, sin embargo para la aplicación final no se utiliza la cámara del dispositivo sino que se muestra un fondo estático. Esta decisión se tomó teniendo en cuenta que el uso de la cámara representa un consumo considerable de

recursos y no realiza un aporte significativo al funcionamiento de esta pantalla.

En el siguiente gráfico, se muestra el diagrama de clases para la escena Resistor en el cual se representan las distintas clases que intervienen en la escena y su relación entre ellas.

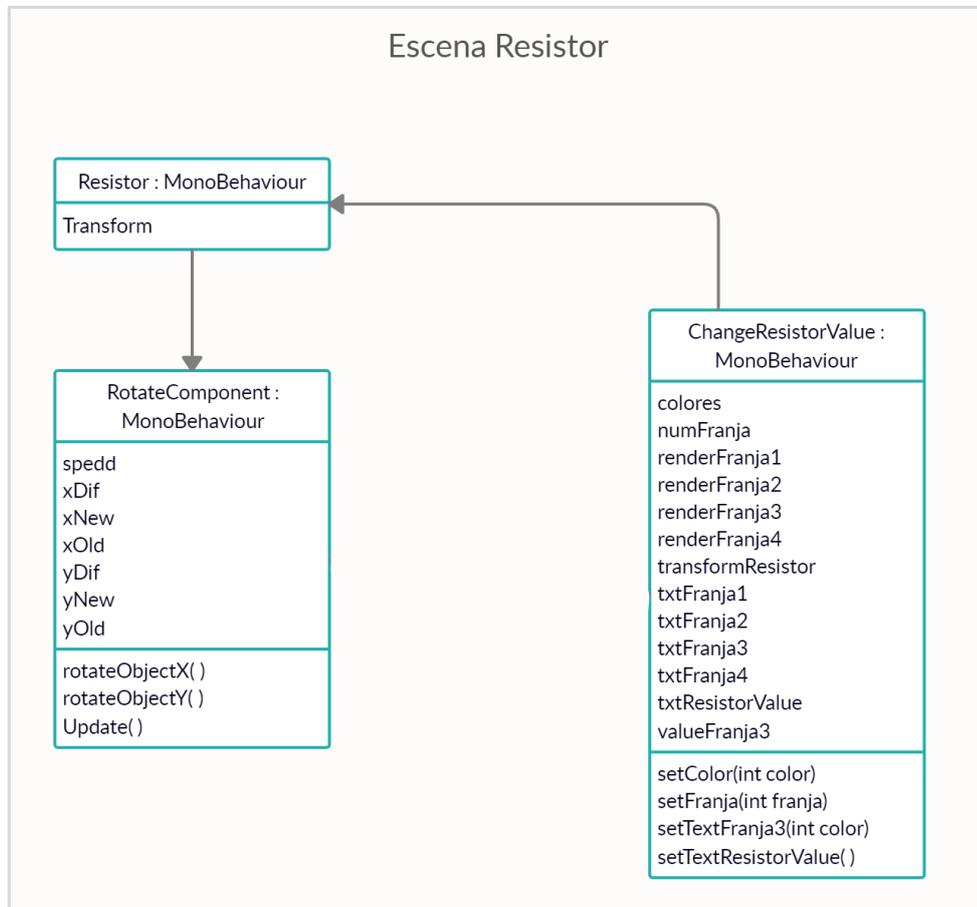


Gráfico 18. Diagrama de clases de la escena Resistor

Escena Display7Seg:

Esta es la escena correspondiente a la aplicación para el dispositivo electrónico display siete segmentos, el cual es un elemento de visualización compuesto por varios LED. En esta pantalla, se presenta una herramienta en la que el usuario puede simular el funcionamiento de un display de este tipo modificando los valores que le ingresan a cada uno de los terminales entre un 1 y 0 lógicos.

La sección izquierda de la pantalla presenta la misma funcionalidad que el caso anterior en donde se presenta el modelo 3D del dispositivo y este puede ser rotado a partir de gestos táctiles del usuario en su dispositivo móvil.

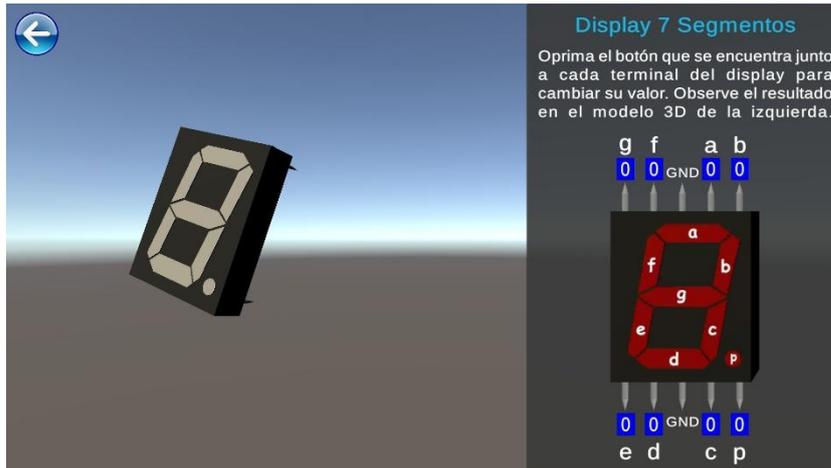


Ilustración 36. Aspecto escena Display7Seg

Sección derecha de la pantalla:

En esta sección se presenta en la parte superior un texto en el cual se explica el funcionamiento de la herramienta y en la parte inferior muestra una imagen que representa al display en conjunto con botones ubicados junto a cada terminal.

Al oprimir cada botón, por medio del evento OnClick del mismo se hace la invocación a la método ChangeSegment de la clase "Change7Segment" que recibe como parámetro un string que corresponde al nombre del segmento que el usuario quiere modificar, posteriormente se obtiene el texto que tiene el botón al momento de ser pulsado por el usuario para modificar los valores de los atributos text y color del botón y además el atributo color del componente material del segmento del modelo 3D que se quiere modificar.

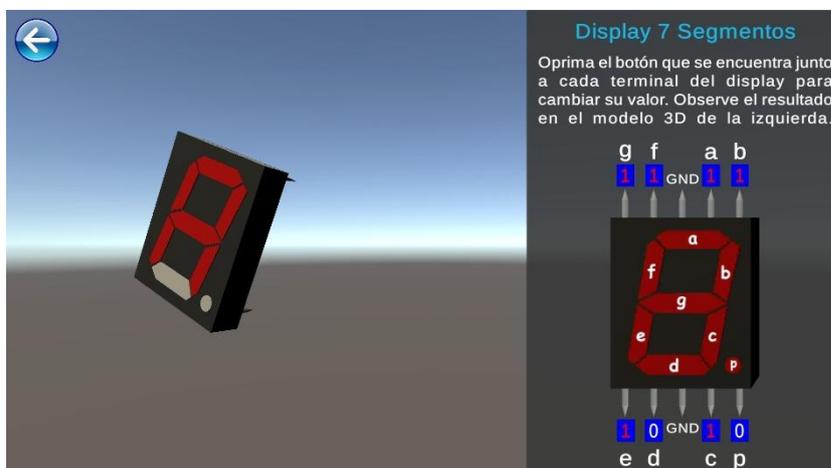


Ilustración 37. Aspecto de la escena Display7Seg al hacer modificaciones sobre el modelo

```

switch (segmento)
{
    case "a":
        if (txtButtonA.text == "0")
        {
            txtButtonA.text = "1";
            txtButtonA.color = Color.red;
            aSegment.material.color = segmentOn;
        }
        else
        {
            txtButtonA.text = "0";
            txtButtonA.color = Color.white;
            aSegment.material.color = segmentOff;
        }
}

```

Ilustración 38. Muestra de fragmento de código utilizado para realizar los cambios en botones y modelo 3D

En el siguiente gráfico, se muestra el diagrama de clases para la escena Display7Seg en el cual se representan las distintas clases que intervienen en la escena y su relación entre ellas.

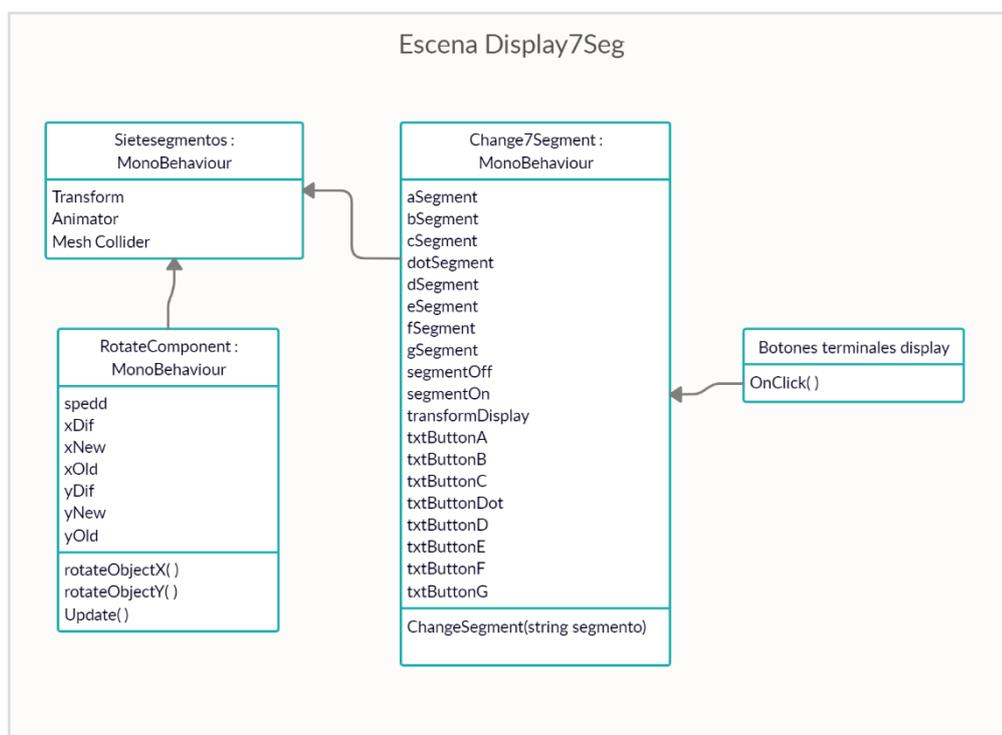


Gráfico 19. Diagrama de clases de la escena Display7Seg

Escena Motor:

En esta se escena se presenta una herramienta en donde el usuario puede observar un motor DC genérico e interactuar con él por medio de la rotación del modelo. Además por medio del botón que se encuentra en el panel derecho de la pantalla puede generar el despiece del motor o

su rearmado (dependiendo del estado actual en que se encuentre el modelo).

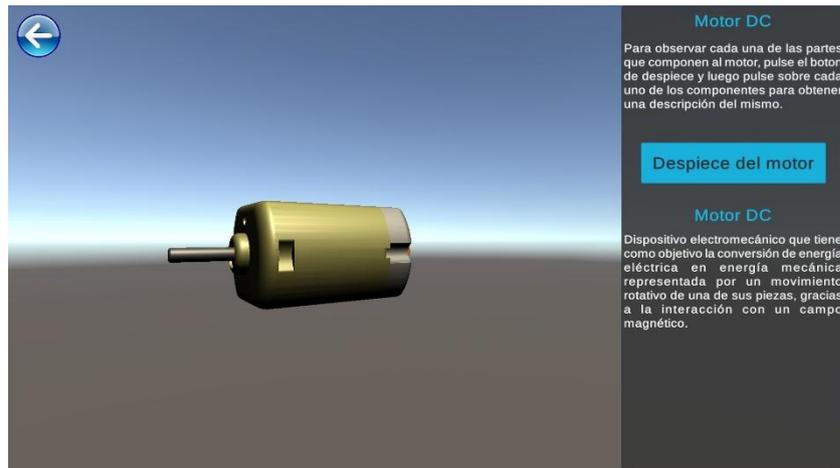


Ilustración 39. Aspecto escena Motor con modelo 3D armado

La animación del despiece del motor se realizó directamente en el entorno Unity 3D creando un clip de animación en el cual se modifican los atributos del componente transform de cada una de las piezas del motor, dicho clip de animación es controlado por medio de un componente animator que hace parte del modelo del motor para esta escena.

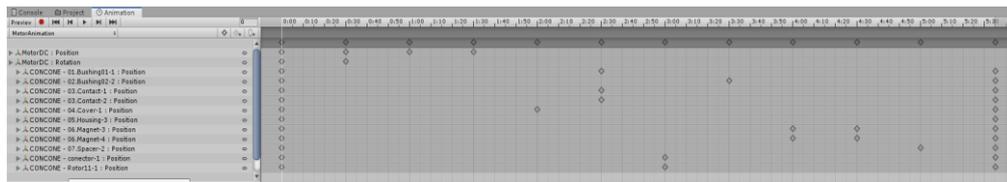


Ilustración 40. Línea de tiempo de animación de despiece del motor

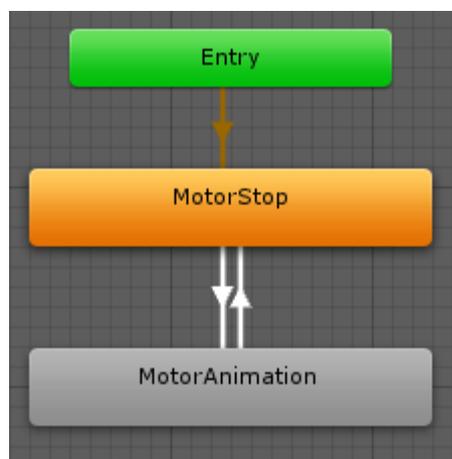


Ilustración 41. Controlador de animaciones del motor

Una vez que el motor se encuentra completamente desarmado, el usuario puede presionar cada una de las piezas que lo componen y en la

parte inferior del panel de la derecha observara una corta descripción sobre la pieza correspondiente.

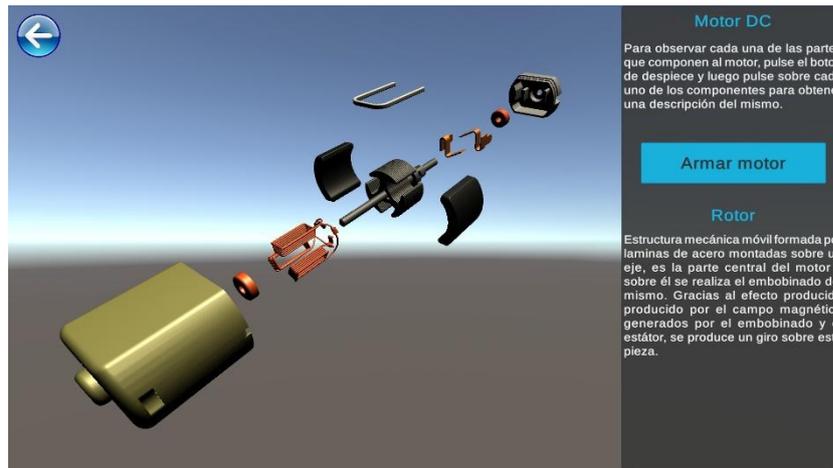


Ilustración 42. Aspecto escena Motor con despiece del motor y descripción de la pieza rotor

Para la activación de la animación del motor, se utiliza el método `PlayAnimation` de la clase llamada "PlayAnimationMotor" el cual es invocado por medio del evento `OnClick` que se encuentra en el panel derecho de la pantalla, este método captura el valor del atributo `text` del botón y con base es este valor determina si se debe activar la animación de despiece del motor o la de armado.

La muestra de la descripción de cada una de las piezas que componen el motor, se hace por medio del evento `OnMouseUpAsButton` de la clase "ShowDescriptionsMotor", este método compara el `tag` (etiqueta) de la pieza seleccionada y dependiendo el resultado obtiene la definición de un diccionario donde se encuentran los valores clave de piezas y definiciones de cada parte del motor, finalmente modifica el atributo `text` del texto que se encuentra en la parte inferior del panel derecho mostrando al usuario la definición de la pieza seleccionada.

```
Dictionary<string, string> descriptionsParts = new Dictionary<string, string>();
Dictionary<string, string> nameParts = new Dictionary<string, string>();

descriptionsParts.Add("bushing", "Elementos mecánicos generalmente fabricados en bronce, permiten el libre giro del rotor y evitan el desgaste y pérdida de energía. Adicionalmente sirven como puntos de conexión para las partes móviles del motor.");
descriptionsParts.Add("contact", "Piezas metálicas, fabricadas generalmente en cobre, están conectadas al embobinado y sirven como terminales externos del motor para conectar a una fuente de energía eléctrica.");
descriptionsParts.Add("cover", "Pieza que hace conjunto con la carcasa y tiene como función el sellar y proteger todos los componentes internos del motor.");
descriptionsParts.Add("magnet", "Pieza fija que rodea al rotor, genera un campo magnético que interactúa con el generado por el bobinado del rotor. En algunos casos este estator pueden ser imanes o bobinados.");
descriptionsParts.Add("spacer", "Pieza fija utilizada en motores que tiene imanes como estator para mantener los imanes separados y a una distancia adecuada para generar el campo magnético.");
descriptionsParts.Add("conector", "Arrollamiento de alambre de cobre que se realiza alrededor del rotor. Su función es generar un campo magnético que interactúa con el campo magnético generado por el estator.");
descriptionsParts.Add("rotor", "Estructura mecánica móvil formada por laminas de acero montadas sobre un eje, es la parte central del motor y sobre él se realiza el embobinado del mismo. Gracias al efecto producido por el campo magnético generado por el embobinado y el estátor, se produce un giro sobre esta pieza.");
descriptionsParts.Add("housing", "Pieza que tiene como función el alojar y proteger todos los componentes internos del motor.");
```

Ilustración 43. Muestra del diccionario creado para las descripciones de las piezas del motor

En el siguiente gráfico, se muestra el diagrama de clases para la escena Motor en el cual se representan las distintas clases que intervienen en la escena y su relación entre ellas.

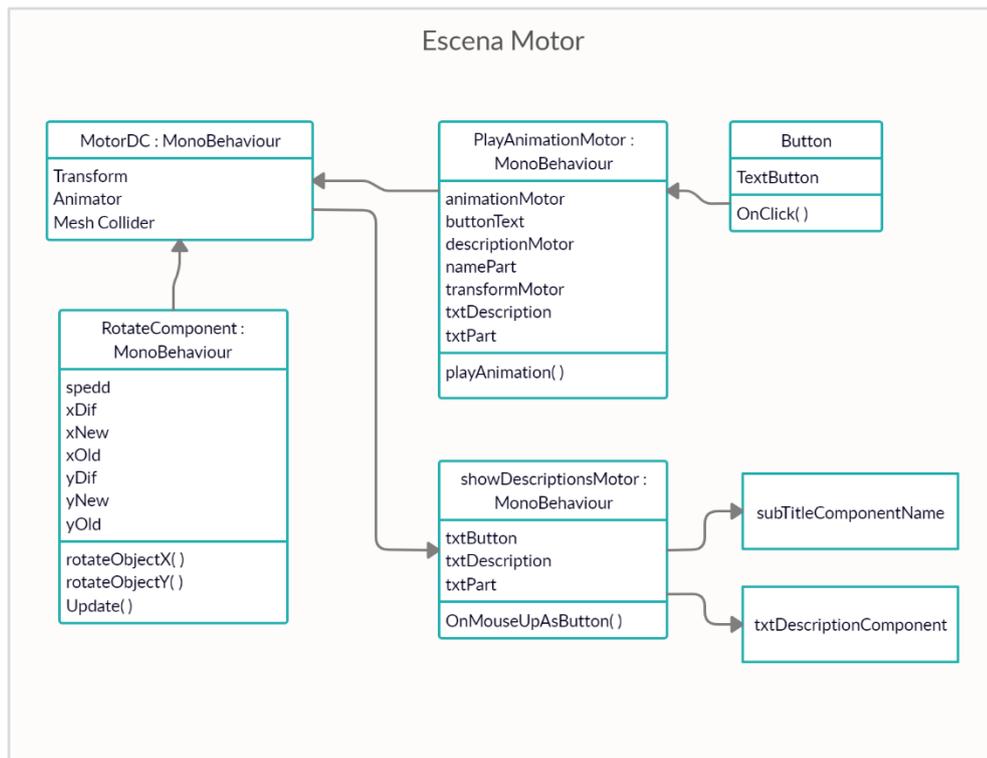


Gráfico 20. Diagrama de clases de la escena Motor

Escena Diode:

En esta escena se presenta el modelo y aplicación desarrollada para el diodo rectificador, al igual que en las escenas ya presentadas, el usuario puede rotar el modelo por medio de gestos táctiles en su dispositivo móvil.

En el panel derecho se ubican dos botones entre los cuales el usuario puede seleccionar entre dos de los circuitos más comunes con los que se utilizan los diodos rectificadores (rectificación media onda y rectificación onda completa). Luego que el usuario realiza su selección, en la sección izquierda de la pantalla se reubica el modelo del diodo a la parte superior de la pantalla y en la parte inferior se muestra una imagen animada del circuito seleccionado así como una representación gráfica de las señales de entrada y de salida del respectivo circuito.

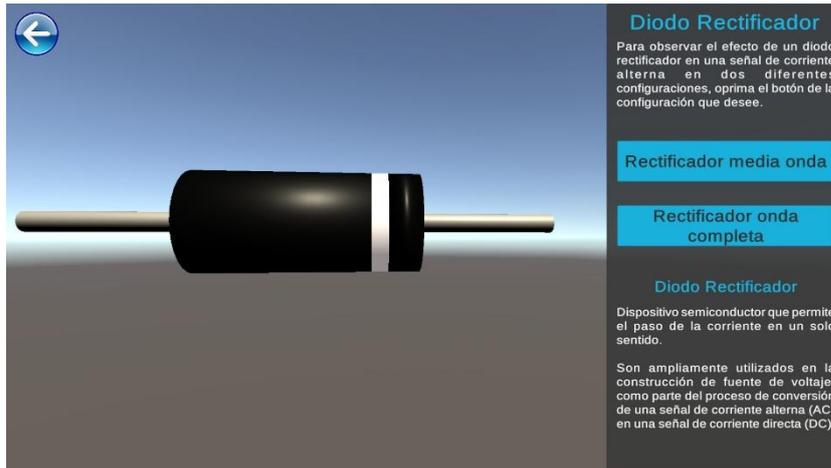


Ilustración 44. Aspecto escena Diode

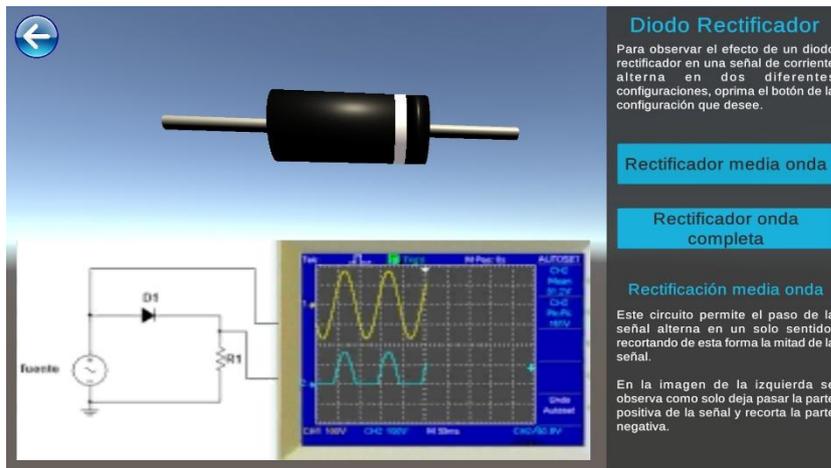


Ilustración 45. Aspecto escena Diode con circuito rectificador de media onda seleccionado

El entorno de desarrollo Unity 3D no cuenta con un objeto o componente nativo que permita mostrar imágenes GIF animadas, por lo cual para mostrar las imágenes animadas de cada circuito, fue necesario descomponer la imagen GIF en cada uno de los cuadros que la componen y posteriormente crear una clase llamada "AnimationGifs" que guarda cada uno de las imágenes en un vector y luego realiza el recorrido del vector mostrando imagen por imagen, realizando la animación.

```

public class AnimationGifs : MonoBehaviour
{
    public Texture2D[] frames;
    public int fps = 0;

    // Update is called once per frame
    void Update()
    {
        int index = (int)(Time.time * fps) % frames.Length;
        GetComponent<RawImage>().texture = frames[index];
    }
}

```

Ilustración 46. Muestra de la clase AnimationGifs

En el siguiente gráfico, se muestra el diagrama de clases para la escena Diode en el cual se representan las distintas clases que intervienen en la escena y su relación entre ellas.

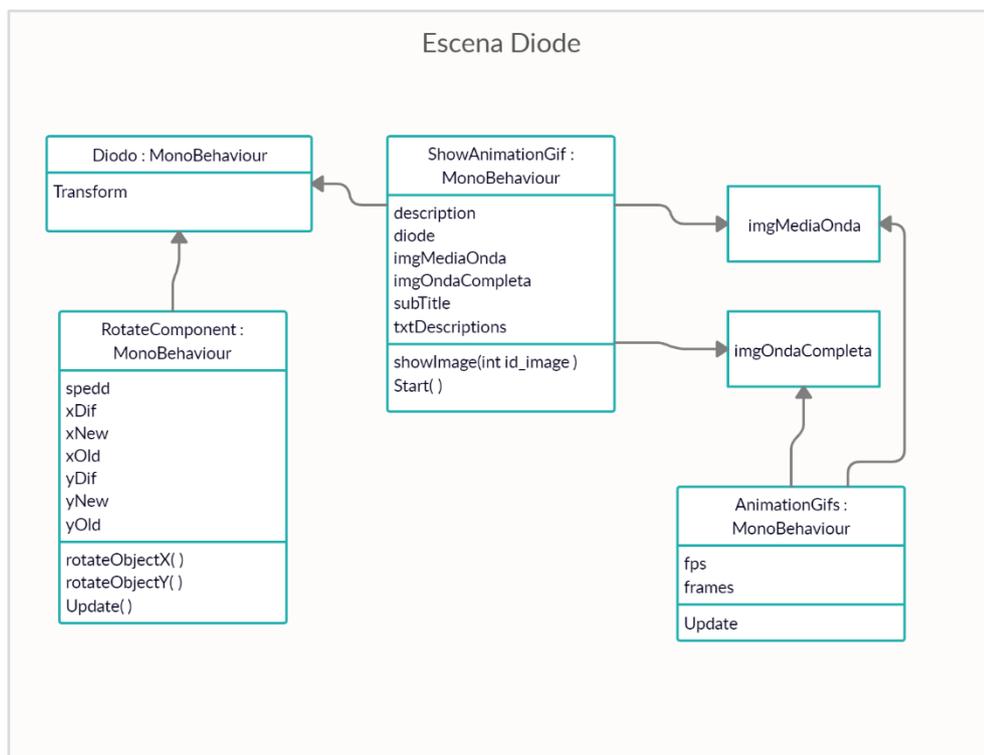


Gráfico 21. Diagrama de clases de la escena Diode

Escena Capacitor:

Por último, la escena capacitor guarda una estructura grafica similar a la presentada en las escenas anteriores.

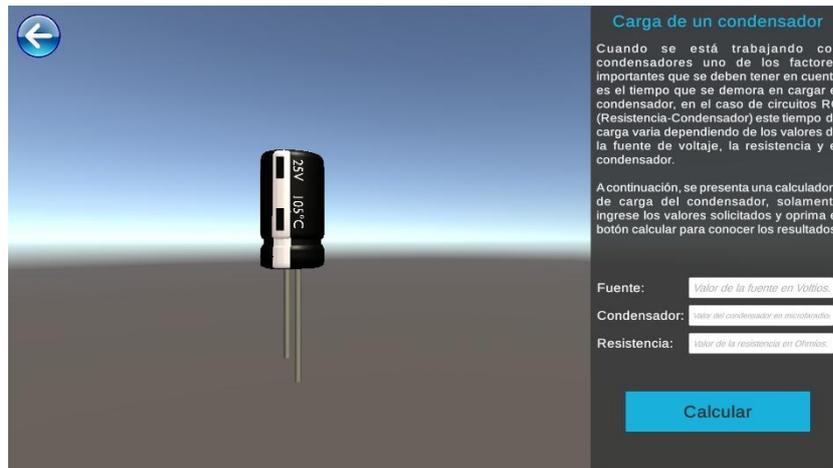


Ilustración 47. Aspecto escena Capacitor

En el panel derecho de la aplicación, se encuentran tres cuadros de entrada de texto y un botón. En los cuadros de texto el usuario podrá ingresar los valores de voltaje de fuente, capacitancia y resistencia de un circuito RC. Además, en la parte inferior del panel se observa un botón con el cual el usuario podrá obtener los tiempos de carga del condensador para un circuito como el antes mencionado.

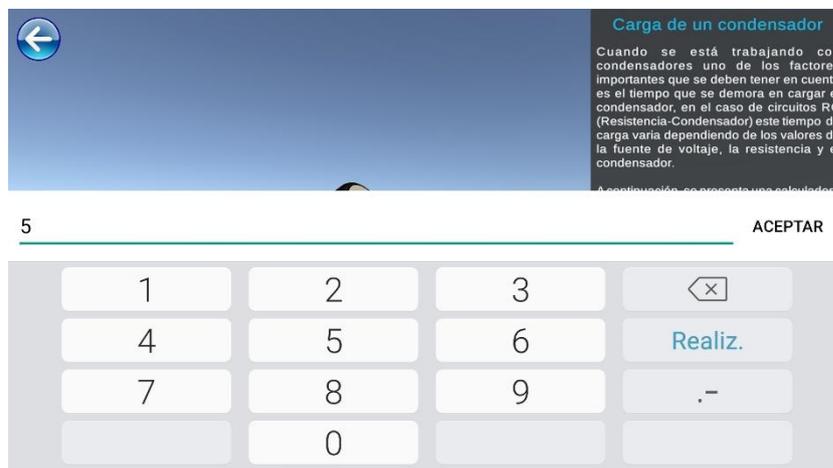


Ilustración 48. Aspecto teclado para el ingreso de datos

Para el cálculo de carga se tienen en cuenta la constante de tiempo del circuito (τ) y la ecuación de carga del condensador. Este proceso de cálculo se realiza por medio del método calculate de la clase "CalculateChargeCapacitor". El método en mención es invocado a partir del evento OnClick del botón que se encuentra en la parte inferior del panel.

Finalmente desde este mismo método se activa la visualización de un panel en la parte inferior de la sección izquierda de la pantalla en donde se muestran todos los valores de carga calculados de acuerdo a la información ingresada por el usuario.

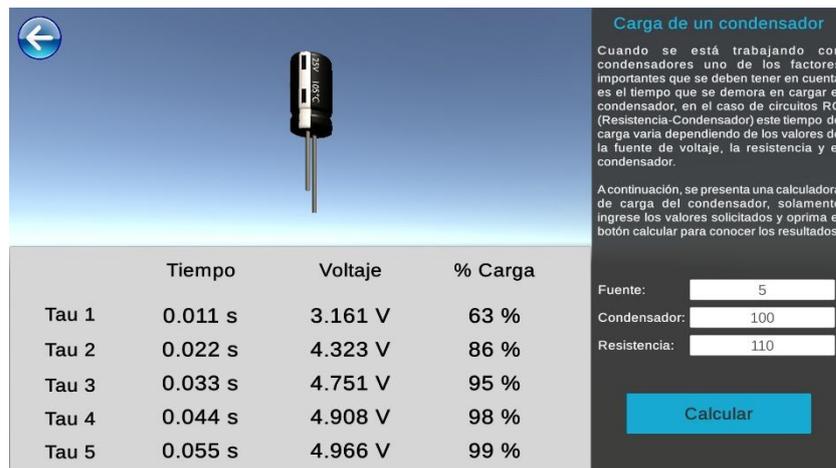


Ilustración 49. Aspecto escena Capacitor con muestra de resultados del calculo

En el siguiente gráfico, se muestra el diagrama de clases para la escena Capacitor en el cual se representan las distintas clases que intervienen en la escena y su relación entre ellas.

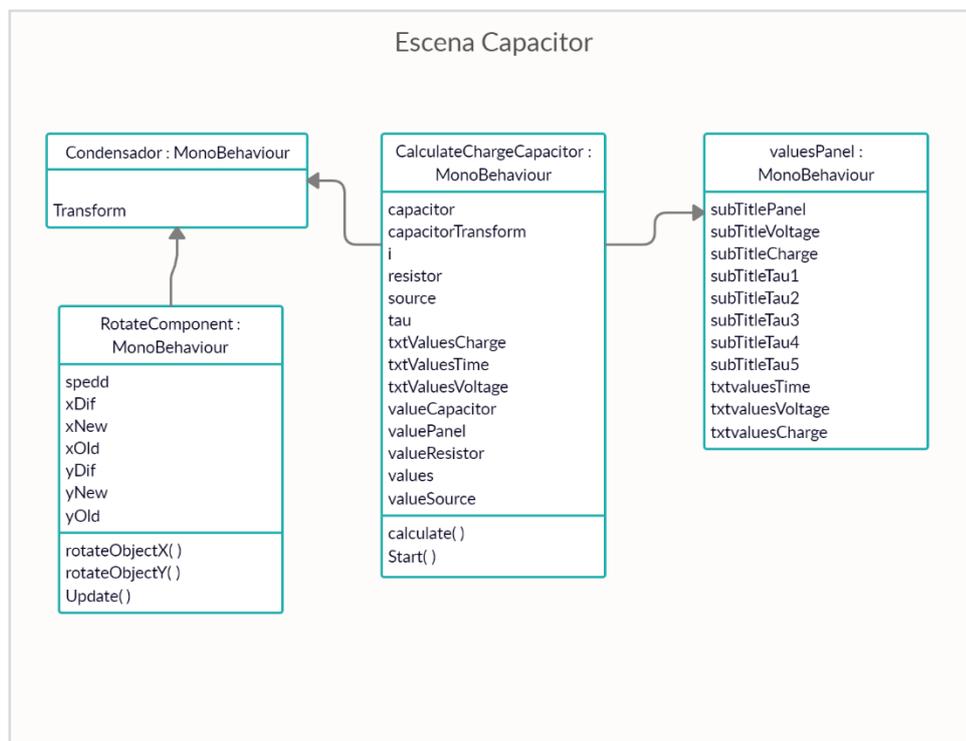


Gráfico 22. Diagrama de clases de la escena Capacitor

3.2.4. Evaluación de la aplicación

Al culminar el desarrollo de la aplicación, construir el proyecto y generar el archivo .apk para la posterior instalación en dispositivos Android. Se obtuvo un archivo de instalación cuyo tamaño es de 40.5 MB y que al ser instalada la aplicación ocupa un espacio de almacenamiento de 80 MB.

La aplicación fue puesta a prueba en distintos dispositivos con sistema operativo Android, de distintas marcas o referencias. La evaluación se centró en el reconocimiento de los marcadores AR a partir de los símbolos encontrados en distintos medios como libros, revistas, páginas de internet e incluso simuladores electrónicos.

Como resultado a esta prueba se obtuvo que la capacidad de reconocimiento de los marcadores varía mucho de la calidad de la cámara con la que cuente el dispositivo móvil, lo cual genera que en algunos dispositivos el reconocimiento se haga de una forma rápida y eficaz y en otros con características más limitadas se obtuvieran problemas de reconocimiento con algunos marcadores.

A continuación se presentan los resultados de reconocimiento obtenidos con símbolos encontrados en distintos medios.

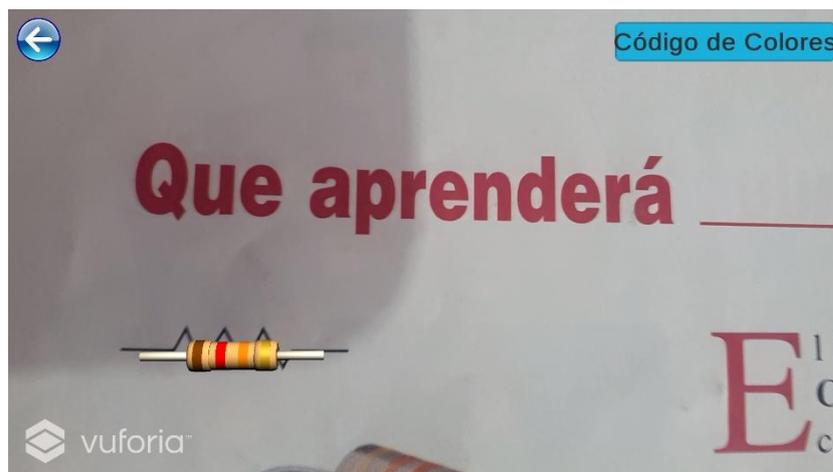


Ilustración 50. Reconocimiento símbolo resistencia de una revista

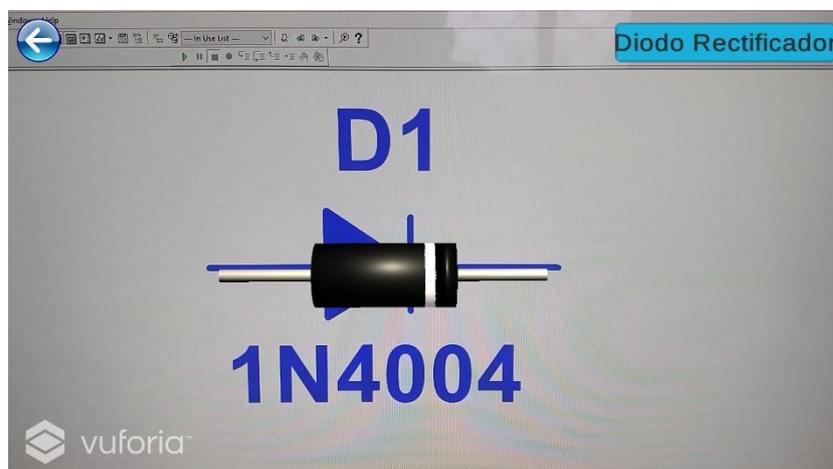


Ilustración 51. Reconocimiento símbolo diodo del simulador electrónico multisim 12

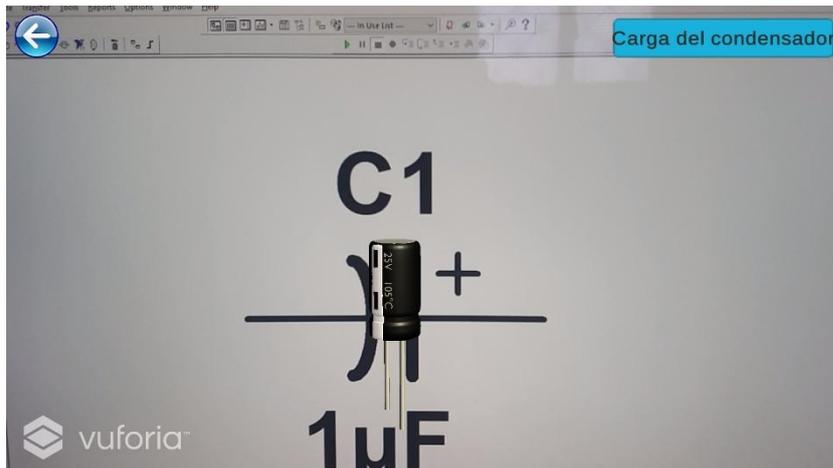


Ilustración 52. Reconocimiento símbolo condensador del simulador electrónico multisim 12

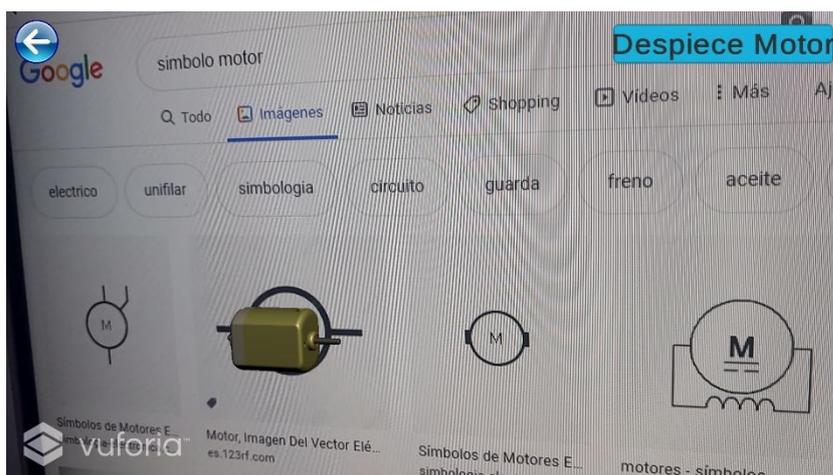


Ilustración 53. Reconocimiento símbolo motor de los resultados de búsquedas de Google

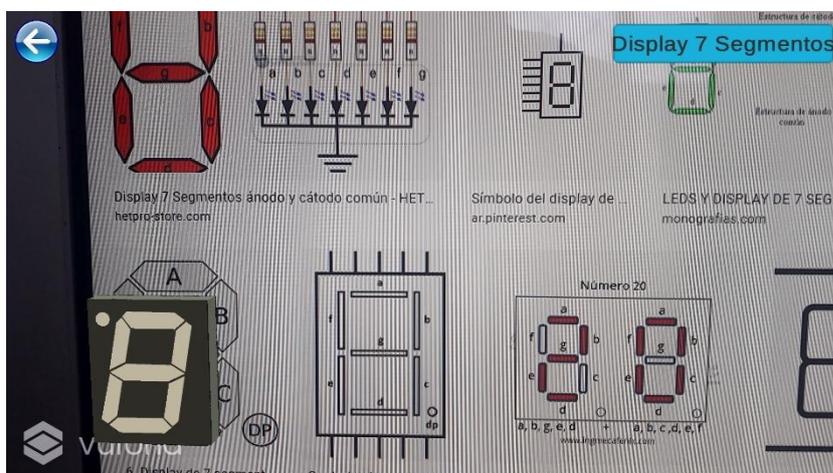


Ilustración 54. Reconocimiento símbolo display 7 segmentos de los resultados de búsqueda de Google

En las escenas de aplicación de cada uno de los componentes electrónicos no fue detectado ningún problema, esto debido a que durante todo el proceso de desarrollo, a estas aplicaciones se les realizaron constantes pruebas de funcionamiento.

Por último el mayor inconveniente que presentó la aplicación en la primera instalación y puesta a prueba fue un retardo considerable en la transición de cualquier escena a la escena de realidad aumentada, esto debido al tipo de procesamiento de los modelos 3D que se estaba realizando. El retardo mencionado era de alrededor de 70 segundos para poder cargar la escena de realidad aumentada.

3.2.5. Corrección de fallos

Luego de establecer los fallos y puntos de mejora necesarios para la aplicación que se mencionaron en la sección anterior. Se procedió a realizar algunas acciones con el fin de solucionar completamente el fallo o mitigarlo en alguna medida.

Problema de reconocimiento de marcadores:

Para mitigar este problema, se optó por mejorar la calidad de las imágenes utilizadas como marcadores y actualizar la base de datos de vuforia.

Para el mejoramiento de las imágenes, se utilizó el programa inkscape para vectorizar las imágenes que ya se tenían, con este proceso se obtuvieron imágenes mejor definidas que el SDK de vuforia reconoce más fácilmente.

La solución anterior como se mencionó en un principio, tan solo mitiga o reduce el problema de reconocimiento de marcadores, razón por la cual en futuras versiones de aplicación se hace necesario el estudio de otros SDK de realidad aumentada para su implementación, o el desarrollo de un algoritmo de visión artificial propio que se adapte a las necesidades de específicas de este proyecto.

Problema de retardo en carga de escena AR:

Para este caso específico, el problema era generado por la configuración del proyecto de Unity para dispositivos Android en el procesamiento y renderizado de las mallas (Meshes) de los modelos 3D.

Por lo tanto se cambiaron las configuraciones de optimización del proyecto, con lo cual se corrigió completamente el problema mencionado anteriormente, pasando de los aproximadamente 70 segundos de retardo para la carga de la escena a un máximo de 5 segundos.



Ilustración 55. Configuraciones de optimización utilizadas para corregir fallo

3. Conclusiones

Al finalizar el proceso de desarrollo de la aplicación Geek Electrónica AR, se puede llegar a concluir lo siguiente:

- Se lograron conseguir todos los objetivos planteados para el desarrollo de la aplicación, esto se logró gracias a la planeación realizada minuciosamente durante la primera etapa de trabajo, pero en mayor medida el éxito en la consecución de los objetivos estuvo determinado por la correcta selección del software a utilizar para este prototipo, donde se optó por utilizar software con amplia documentación lo que permitió solucionar los problemas presentados de forma ágil. Además, que ya se contaba con alguna experiencia previa en el uso de estos programas informáticos, lo cual produjo que la curva de aprendizaje del manejo de los mismo no fuera elevada.
- El haber utilizado la metodología de desarrollo en cascada en este proyecto, generó los resultados esperados ya que permitió seguir a cabalidad con cada una de las etapas planteadas durante la fase de planeación de trabajo. Esta metodología a pesar de ser completamente rígida y secuencial, fue la adecuada ya que debido a las limitaciones de tiempo para el desarrollo de la aplicación y cantidad de personas trabajando en el desarrollo del proyecto (1 persona), el utilizar alguna metodología iterativa como por ejemplo desarrollo ágil, acarrearía una carga extra de trabajo al tener que realizar proceso de desarrollo, evaluación y corrección de fallos en cada una las distintas etapas del proyecto.
- Aunque se consiguieron los objetivos planteados, para futuras implementaciones con fines de distribución, es necesario replantear el uso del SDK de realidad aumentada Vuforia por la siguientes razones:
 - Los fallos de reconocimiento de marcadores que se evidenciaron y documentaron durante el proceso de evaluación de la aplicación.
 - La licencia de Vuforia que se utilizó en este proyecto, es una licencia de desarrollador, en tal caso que se desee realizar la distribución de la aplicación es necesario adquirir alguna licencia de pago.

Por lo tanto en posteriores implementaciones de la aplicación se evaluarán los SDK de realidad aumentada ARToolkit y ARCore.

- El fallo encontrado en el reconocimiento de algunos símbolos electrónicos, no solamente es debido al SDK de realidad

aumentada utilizado, parte de este fallo también en generado por la falta de uniformidad o de seguimiento de la normatividad existente en los símbolos que se encuentran en diferentes materiales educativos.

Por lo tanto de ser necesario, para futuras implementaciones puede ser viable el retomar el concepto inicial del proyecto, en donde la aplicación funciona en conjunto con un libro creado por el mismo desarrollador y que contiene todos los símbolos reconocibles en la aplicación y una explicación teórica de cada uno de los elementos electrónicos.

- Para completar la aplicación y obtener un producto listo para distribución, es necesario ampliar la base de datos de componentes electrónicos que puede reconocer. Además, crear estrategias de distribución y monetización de la aplicación.

4. Glosario

ANDROID: Sistema operativo para dispositivos móviles.

API: interfaz de programación de aplicaciones

DCU: Diseño centrado en el usuario

ENTORNO DE DESARRO INTEGRADO: software que provee distintas herramientas, las cuales son de utilidad en programación y que le facilitan el trabajo al programador o desarrollador de software.

IDE: Entorno de desarrollo integrado

KIT DE DESARROLLO DE SOFTWARE: es un conjunto de herramientas que utiliza el programador para desarrollar aplicaciones en un lenguaje o para una plataforma en específico, tiene como finalidad simplificar el trabajo del programador además de permitir la integración entre distintos sistemas.

REALIDAD AUMENTADA: Tecnología perteneciente al área de la informática, que permite que por medios de visión artificial para el reconocimiento de marcadores o de geo-posicionamiento para determinación de ubicación del usuario, se puedan sobreponer en una pantalla objetos virtuales a un entorno físico real.

SDK: kit de desarrollo de software

5. Bibliografía

- Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4), 355-385.
- Cabero Almenara, J., & Llorente Cejudo, M. D. C. (2019). Evaluación de software de producción de objetos en Realidad Aumentada con fines educativos.
- Flamarich, J. (2017). Diseño de productos interactivos multidispositivo. Universitat Oberta de Catalunya.
- Gómez García, G., Rodríguez Jiménez, C., & Marín Marín, J. A. (2020). La trascendencia de la Realidad Aumentada en la motivación estudiantil. Una revisión sistemática y meta-análisis. *ALTERIDAD. Revista de Educación*, 15(1), 36-46.
- González-Moreno, M. S. E., Montalvo, J. A. C., & Palma-Ruiz, J. M. (2019). 3 La industria cultural y la industria de los videojuegos. *JUEGOS Y SOCIEDAD:: DESDE LA INTERACCIÓN A LA INMERSIÓN PARA EL CAMBIO SOCIAL*, 19, 20 – 23.
- Kellems, R. O., Cacciatore, G., & Osborne, K. (2019). Using an augmented reality–based teaching strategy to teach mathematics to secondary students with disabilities. *Career Development and Transition for Exceptional Individuals*, 42(4), 253-258.
- Montecé-Mosquera, F., Verdesoto-Arguello, A., Montecé-Mosquera, C., & CaicedoCamposano, C. (2017). Impacto de la realidad aumentada en la educación del siglo XXI. *European Scientific Journal, ESJ*, 13(25), 129-137.
- Peres Irving. (2019). El valor de la industria de los videojuegos en 2019. Forbes México. Recuperado de: <https://www.forbes.com.mx/el-valor-de-la-industria-de-los-videojuegos-en-2019/>
- Prendes Espinosa, C. (2015). Realidad aumentada y educación: análisis de experiencias prácticas. *Píxel-Bit. Revista de Medios y Educación*, 46, 187-203.
- Ramírez, R. Metodologías para el desarrollo de aplicaciones móviles. Universitat Oberta de Catalunya. 21-36.
- Rodríguez Deivis. (2020). Industrial de los videojuegos arrasó en 2019 con ganancias sobre los USD \$120.000 millones. FayerWayer. Recuperado de: <https://www.fayerwayer.com/2020/01/videojuegos-2019-ganancias/>
- Rovira, J. (2017). Diseño avanzado de productos interactivos multidispositivo. Universitat Oberta de Catalunya.
- Telefónica, F. (2011). *Realidad Aumentada: una nueva lente para ver el mundo*. Fundación Telefónica, 10-13.
- Wang, Y. H. (2017). Using augmented reality to support a software editing course for college students. *Journal of Computer Assisted Learning*, 33(5), 532-546.

6. Anexos

6.1. Ejemplos de imágenes válidas para prueba de la aplicación

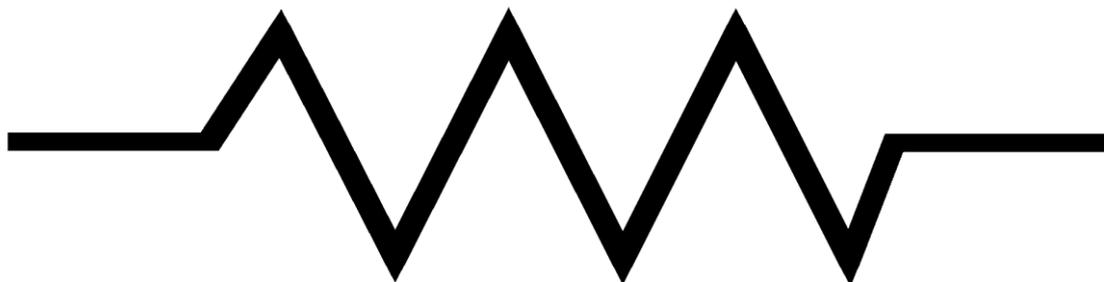


Ilustración 56. Imagen Marcador resistencia

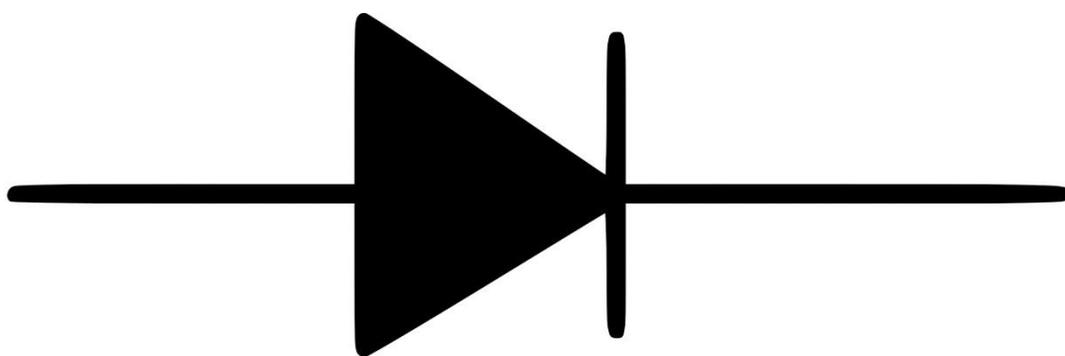


Ilustración 57. Imagen Marcador diodo rectificador

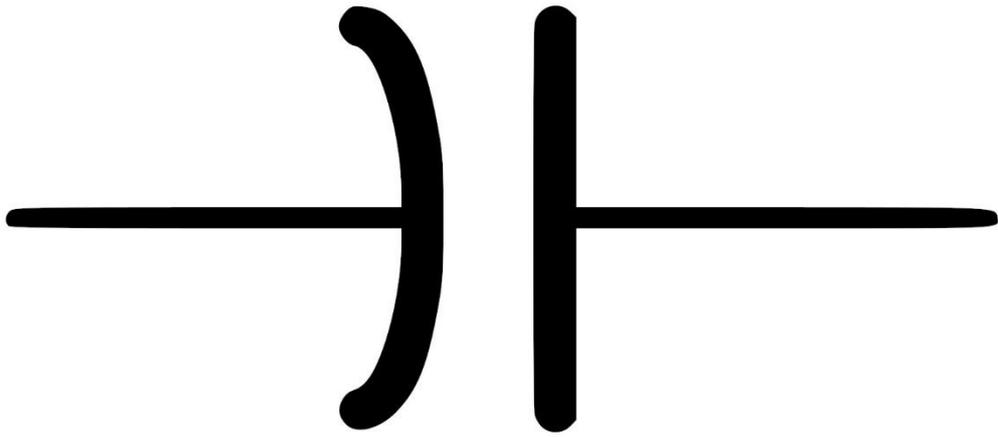


Ilustración 58. Imagen Marcador condensador

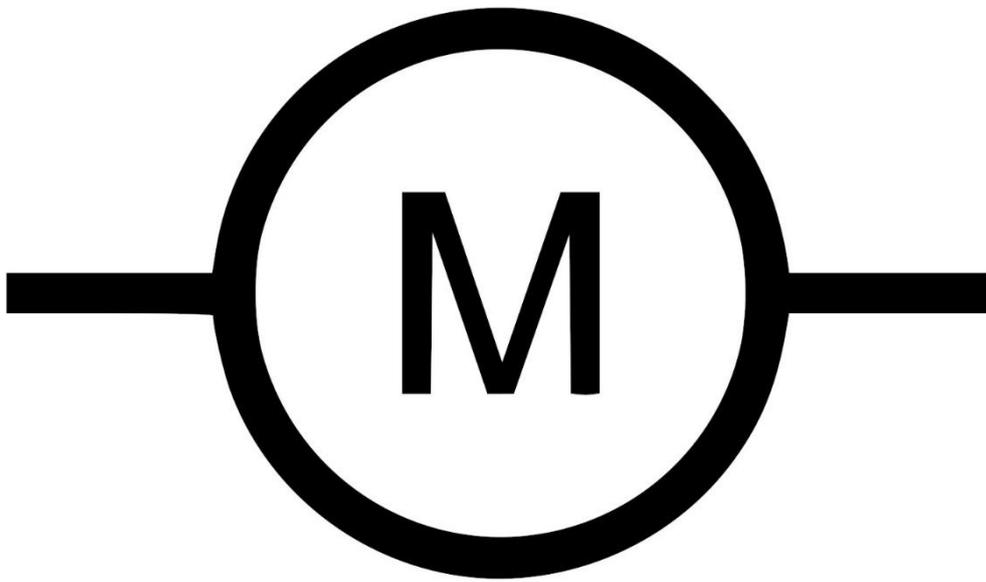


Ilustración 59. Imagen Marcador motor

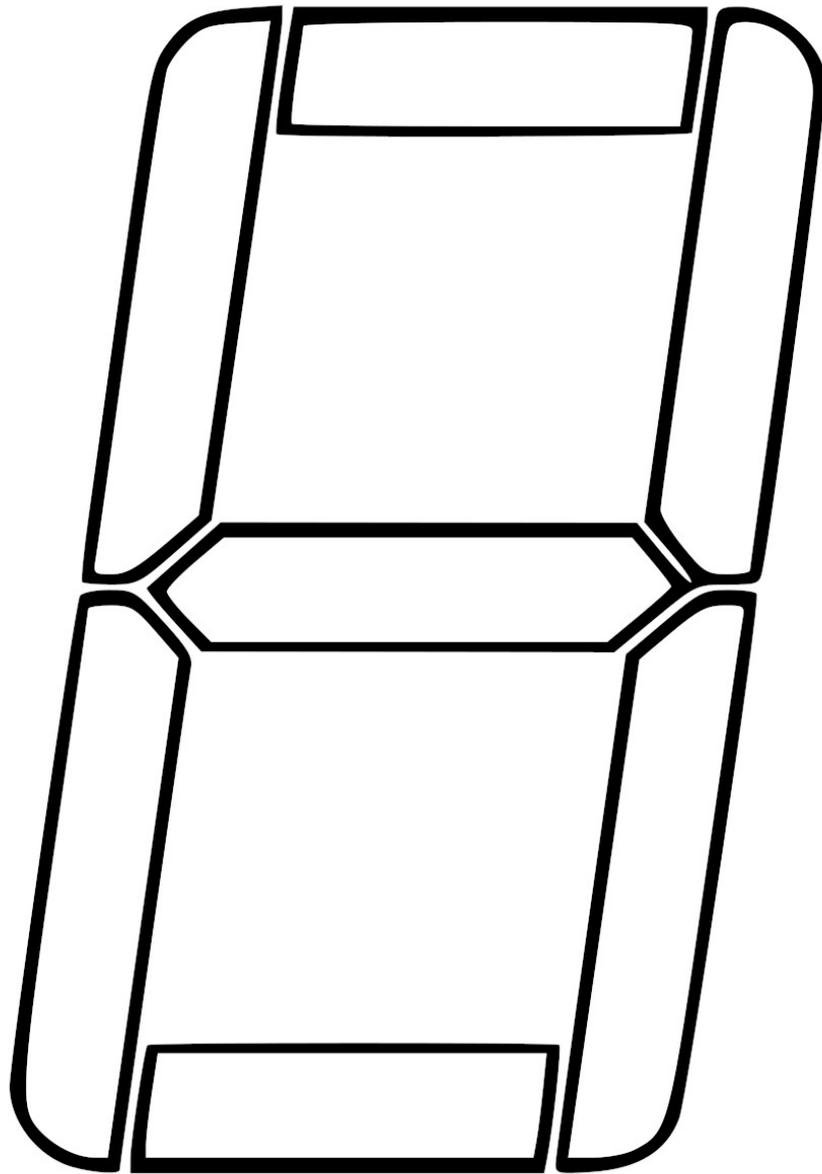


Ilustración 60. Imagen Marcador display siete segmentos