

TECNOLÓGICO NACIONAL DE MÉXICO



TECNOLÓGICO
NACIONAL DE MÉXICO



SEP
SECRETARÍA
DE EDUCACIÓN
PÚBLICA



INSTITUTO TECNOLÓGICO DE CIUDAD GUZMÁN
TÍTULO:

**ON THE USE OF AUGMENTED REALITY TO
REINFORCE THE LEARNING OF POWER
ELECTRONICS FOR BEGINNERS**

**(Sobre el uso de la Realidad Aumentada para reforzar
el aprendizaje de Electrónica de potencia para
principiantes)**

**QUE PARA OBTENER EL TÍTULO DE:
MAESTRO EN INGENIERÍA ELECTRÓNICA**

PRESENTA:

VILLA BARBA MIGUEL ÁNGEL

NÚMERO DE CONTROL: M20291054

DIRECTOR: DR. SERGIO SANDOVAL PÉREZ

CODIRECTOR: DR. JUAN MIGUEL GONZÁLEZ LÓPEZ

CD. GUZMÁN, JALISCO, AGOSTO 2022

Instituto Tecnológico de Ciudad Guzmán
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Ciudad Guzmán, Jalisco, **03/agosto/2022**
Asunto: Autorización de impresión de Tesis

ING. MIGUEL ÁNGEL VILLA BARBA
CANDIDATO AL GRADO DE MAESTRO EN INGENIERÍA ELECTRÓNICA
PRESENTE

De acuerdo con los Lineamientos para la Operación de los Estudios de Posgrado en el Tecnológico Nacional de México y las disposiciones en este Instituto, habiendo cumplido con todas las indicaciones que la Comisión Revisora realizó con respecto a su Trabajo de Tesis titulado **"Sobre el uso de la realidad aumentada para reforzar el aprendizaje de electrónica de potencia para principiantes"**, la División de Estudios de Posgrado e Investigación de este Instituto, concede la Autorización para que proceda a la impresión del mismo.

Sin otro particular, quedo de Usted.

ATENTAMENTE

Excelencia en Educación Tecnológica*

2022, AÑO DEL CINCUENTA ANIVERSARIO DEL INSTITUTO TECNOLÓGICO DE CIUDAD GUZMÁN*



MARÍA GUADALUPE SÁNCHEZ CERVANTES
JEFA DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN

ccp. Archivo

MGSC/megg



<https://cdguzman.tecnm.mx/sgciltog/>

Av. Tecnológico #100 C.P. 49100 Ciudad Guzmán, Jal. Tel. (341) 5752050

tecnm.mx | cdguzman.tecnm.mx



AGRADECIMIENTOS

¡Gracias a Dios!

A mis tutores, Dr. Sergio Sandoval Pérez y Dr. Juan Miguel González López, sin ustedes y sus virtudes, su paciencia y constancia este trabajo no lo hubiese logrado. Ustedes formaron parte importante de esta historia con sus aportes profesionales que los caracterizan. Muchas gracias por sus múltiples palabras de aliento, cuando más las necesite; por estar allí cuando mis horas de trabajo se hacían confusas. Gracias por sus orientaciones.

A mis docentes, sus palabras fueron sabias, sus conocimientos rigurosos y precisos, a ustedes mis profesores queridos, les debo mis conocimientos. Su semilla de conocimientos, germinó en el alma y el espíritu. Gracias por su paciencia, por compartir sus conocimientos de manera profesional e invaluable, por su dedicación perseverancia y tolerancia.

A mis padres, esposa e hijo, ustedes han sido siempre el motor que impulsa mis sueños y esperanzas, quienes estuvieron siempre a mi lado en los días y noches más difíciles durante mis horas de estudio. Siempre han sido mis mejores guías de vida. Hoy cuando concluyo mis estudios, les dedico a ustedes este logro, como una meta más conquistada. Orgulloso de que sean mi familia y que estén a mi lado en este momento tan importante. Gracias por ser quienes son y por creer en mí.

A mis amigos y compañeros de la maestría en Electrónica, hoy nos toca cerrar un capítulo maravilloso en esta historia de vida y no puedo dejar de agradecerles por su apoyo y constancia, al estar en las horas más difíciles, por compartir horas de estudio. Gracias por estar siempre allí.

TABLA DE CONTENIDO

AGRADECIMIENTOS	3
TABLA DE FIGURAS	8
Abstract:	12
Resumen:	12
1 CAPÍTULO I	14
INTRODUCCIÓN	14
1.1. Introducción	14
1.2. Trabajo relacionado	15
CAPÍTULO 2	18
MARCO TEÓRICO	18
¿Qué es Unity 3d?	18
2.1 Instalación de Unity 3d	18
2.1.1 Descargar e instalar Unity HUB	18
2.1.2 Instalar versión del motor Unity	19
2.1.3 Añadir módulos a la versión de Unity	20
2.1.4 Finalizar instalación y crear nuevo proyecto	22
2.2 Integración de ARcore	24
2.2.1 ¿Qué es ARCore?	24
2.2.2 ¿Cómo funciona ARCore?	25
2.2.3 ¿Por qué se opta por ARCore?	26
2.3 Modelado 3d Blender	27
2.3.1 Introducción al modelado en Blender	27
2.3.2 Modos	27

2.3.3	Modo de edición	28
2.4	Ensamble de modelos 3d	30
2.4.1	Agregar componentes al espacio 3d	30
2.4.2	Texturas	31
2.4.3	Archivos 3D exportados	31
2.4.4	Modelos	31
2.4.5	Posicionar protoboard	31
2.4.6	Colocar modelos en posiciones coincidentes	32
2.4.7	Revisar coincidencias	32
2.5	Creación de animaciones con timeline	32
2.5.1	Ventana de Timeline	32
2.5.2	Ventana de animación	34
2.5.3	Creando un nuevo clip.	35
2.5.4	Herramientas de la ventana de animación	36
2.5.5	Herramientas de la ventana de animación de unity	36
2.6	Creación de UIs	38
2.6.1	Diseño básico	38
2.6.2	La Rect tool	38
2.6.3	Rect transform	39
2.6.4	Cambiar el Tamaño Versus Escalando	40
2.6.5	Pivot (Pivote)	40
2.6.6	Anclajes	40
2.6.7	Presets de la Ancla	42
2.6.8	Los campos de Anchor (ancla) y position (posición) en el inspector	43

2.7	Interacción UIs y animaciones	45
2.7.1	Componentes de Interacción	45
2.7.2	Funcionalidad Común	45
2.7.3	Button (Botón)	45
2.7.4	Toggle	47
2.7.5	Slider (Deslizador)	50
2.7.6	Animación	51
2.8	Scripting	54
2.8.1	Creando y usando scripts	54
2.8.2	Creando Scripts	54
2.8.3	Anatomía de un archivo Script.	55
2.8.4	Controlando un GameObject	57
2.8.5	Scripts usados en este proyecto.	58
2.8.6	Ejemplo de uso frecuente de los scripts en el proyecto.	59
2.9	Build	60
2.9.1	Configurando su ambiente de desarrollo Android	60
2.9.2	Agregar la ruta del SDK de Android a Unity	61
2.9.3	Hacer que el dispositivo sea reconocido por el sistema	61
2.9.4	Construcción	62
2.10	Pruebas	63
2.11	Resultados	63
CAPÍTULO 3		64
PROCESO METODOLÓGICO PARA EL DESARROLLO DE LA APP AR-ELECTRONICS		64

3.1	Descripción del desarrollo y diseño de la app AR-Electronics	64
3.2	Descripción de la App Electrónica Diseñada y Desarrollada	66
3.3	Desarrollo del AR para Circuitos RLC	69
3.4	Desarrollo de AR para el convertidor DC-DC Buck-Boost	71
CAPÍTULO 4.		74
PROCESO EXPERIMENTAL		74
4.1	Participantes	74
4.2	Percepciones y Experiencias de las Tecnologías	76
4.3	Realidad Mixta aplicada al control no lineal de la autonomía del tractor	80
CAPÍTULO 5.		96
RESULTADOS		96
5.1	Resultados y discusiones	96
5.2	Contribuciones	98
5.3	Conclusiones	100
ANEXOS		100
ANEXO A. REGISTRO DE CAMBIOS		100
BIBLIOGRAFÍA		106

TABLA DE FIGURAS

Figura. 1. En el enlace de arriba se pueden descargar la aplicación Unity HUB. .	18
Figura. 2. En la sección «Installs» se pueden añadir nuevas versiones de Unity y agregar módulos a versiones ya instaladas.	19
Figura. 3. Con el botón Add se puede instalar una nueva versión del motor Unity.	19
Figura. 4. En esta ventana se selecciona la versión del motor Unity que se desea instalar.	20
Figura. 5. Ventana de selección de módulos para añadir al motor Unity. Módulo Visual Studio para editar Scripts seleccionado.	21
Figura. 6. Módulo Android para exportar juegos y aplicaciones Android con Unity.	21
Figura. 7. Módulo WebGL para exportar juegos y aplicaciones para navegadores con Unity.	22
Figura. 8. Ventana de gestión de proyectos Unity.	23
Figura. 9. Botón para crear un nuevo proyecto seleccionando la versión del motor Unity a utilizar.	23
Figura. 10. Configuración inicial de un proyecto en Unity.	24
Figura. 11. Configuración de AR Core para Unity.	26
Figura. 12. Ventana principal de Blender.	27
Figura. 13. Modos en la edición de modelos 3D en Blender.	28
Figura. 14. Objeto 3D a exportar.	29
Figura. 15. Ventanas para exportar el modelo 3D.	29
Figura. 16. Ventana de dialogo para confirmar el directorio donde se guardar el archivo en FBX	30
Figura. 17. Componentes requeridos en las practicas posicionados en el espacio 3D en Unity.	31
Figura. 18. Protoboard posicionado en el espacio 3D en Unity.	31
Figura. 19. Coincidencia de los pines de componentes con el protoboard	32

Figura. 20. Guardando secuencia de Timeline para un objeto 3D	33
Figura. 21. Pestaña "Activation Track" en Timeline.	33
Figura. 22. Ingresar a la ventana de animación de Unity.	34
Figura. 23. Creando un nuevo animator y clip de animacion.	35
Figura. 24. Elegir la carpeta de destino y el nombre de tu clip de animación.	35
Figura. 25. Clip de animación vacío	36
Figura. 26. Boton "Preview" de la ventana de animacion.	36
Figura. 27. Visualización del frame en cada parte de la animación.	37
Figura. 28. Crear múltiples animaciones para el mismo objeto.	37
Figura. 29. Añadir una nueva keyframe en la linea de tiempo de la animación.	37
Figura. 30. Añadir un evento.....	37
Figura. 31. Tipo de visualización de la animación.....	38
Figura. 32. Botones de la pestaña de herramientas con la Rect Tool seleccionada.	38
Figura. 33. Botones de la barra de herramientas configuradas a Pivot y Local.	39
Figura. 34. Componente Rect Transform, característico de los elementos UI	39
Figura. 35. Pivote.....	40
Figura. 36. Los anclajes de UI	41
Figura. 37. Elemento UI anclado a la esquina	41
Figura. 38. Tipos de anclajes más comunes.....	43
Figura. 39. Componente Rect Transform UI	44
Figura. 40. Imagen de un Boton en Unity3D	45
Figura. 41. Propiedades del botón en unity	46
Figura. 42. Imagem de un Toggle unity 3D	48
Figura. 43. Propiedades del Toggle en unity	49
Figura. 44. Imagen de en Slider Unity 3D	50
Figura. 45. Componente Animation adjunto a Botón	51
Figura. 46. Ventana de animación	52
Figura. 47. Menú Animation Clip.....	52
Figura. 48. Imagen de un Script al ser creado	55

Figura. 49. Ejemplo de script utilizado en el proyecto.....	56
Figura. 50. Script en la ventana Inspector	57
Figura. 51. Ejemplo de Script instanciado en un GameObject.....	58
Figura. 52. Signal Emitters.....	60
Figura. 53. Escenas agregadas a la Build de una APK	62
Figura. 54. Power electronic course.....	65
Figura. 55. Diseño de la aplicación AR—electrónica en UNITY.....	66
Figura. 56. Ventana de animaciones en Timeline	68
Figura. 57. AR—resistor GameObject.	68
Figura. 58. Script UIHandler para activar/desactivar la ayuda visual o los cambios de escena.	69
Figura. 59. Circuito en serie RLC.....	70
Figura. 60. AR - Análisis matemático interactivo del circuito RLC	70
Figura. 61. GUI del circuito AR–RLC para ensamblar dispositivos en una protoboard electrónica.....	71
Figura. 62. Diagrama eléctrico del convertidor Buck-Boost bidireccional	72
Figura. 63. Topología de convertidor DC-DC bidireccional.....	72
Figura. 64. AR GUI para ensamblar el convertidor Buck/Boost. (a) Guía de conexión para Buck; (b) Ayuda de montaje para Boost; (c) Usar un osciloscopio para medir el voltaje de las baterías.	73
Figura. 65. Usando la aplicación AR-electronics.....	75
Figura. 66. Evaluación general de la aplicación AR-electronics.....	77
Figura. 67. Encuesta de circuitos AR-RLC.	79
Figura. 68. Encuesta AR-DC-DC	80
Figura. 69. Interfaz para login en Firebase	81
Figura. 70. Módulos requeridos en Unity - Firebase	82
Figura. 71. Venta para agregar proyecto en Firebase	83
Figura. 72. Ventana para agregar APP en Firebase	83
Figura. 73. Registro de APP para Android.....	84
Figura. 74. Vista de la carpeta Assets en Unity	85

Figura. 75. Sección de código encontrado en Firebase.....	86
Figura. 76. Tabla de productos Firebase - Unity.....	87
Figura. 77. Ubicación del script FirebaseManager en Assets>Scripts	87
Figura. 78. UI para realizar login Unity - Firebase.....	88
Figura. 79. Sección para introducir url de la base de datos	88
Figura. 80. Ejemplo de UI Text Mesh Pro	89
Figura. 81. Sección de script FirebaseManager.....	90
Figura. 82. Sección de script SteeringWheel	91
Figura. 83. Script UIChanger	92
Figura. 84. Método publico SaveSteeringWheel.....	92
Figura. 85. Scripts.....	93
Figura. 86. UI con volante y valores.....	94
Figura. 87. Ejemplo de envío de datos Unity - Firebase	94
Figura. 88. Pruebas de plataforma de tractor y Realidad Mixta	95
Figura. 89. Envío de datos a través de Firebase a plataforma de tractor autónoma	95
Figura. 90. Medición de la mejora del conocimiento.	97
Figura. 91. Certificado de aceptación de artículo.....	99
Figura. 92. Aplicación RLC y Buck - Boost,	99

ABSTRACT:

This study aims to develop and evaluate an Augmented Reality (AR) application to teach power electronics to beginners. For this purpose, two topics were presented: The first was the design of a series-connected Resistance–Inductor–Capacitor (RLC) circuit in AR, the space-state equations of which were analyzed in an interactive way, and its assembly in a virtual protoboard to analyze the voltage and currents as measured by an oscilloscope. The second presented topic in AR was about Bidirectional Direct Current (DC)–DC converters, known as Buck–Boost; the aim was to study their behavior when energy is exchanged between two systems, usually photovoltaic panels, electric vehicles, and storage systems. The attitudes of the students towards the AR application was significantly better than those towards traditional teaching. The measurements of the developed skills indicated better cognitive performance when using AR technology. The designed AR tool was used in an industry course to explore the students' opinions, who provided valuable feedback.

RESUMEN:

Este estudio tiene como objetivo desarrollar y evaluar una aplicación de Realidad Aumentada (AR) para enseñar electrónica de potencia para principiantes. Para ello se presentan dos temas: Se diseñan circuitos Resistencia-Inductor-Condensador (RLC) conectados en serie en AR, analizando las ecuaciones espacio-estado de forma interactiva y su montaje en un protoboard virtual para analizar tensiones y corrientes. Como podría medirse en un osciloscopio. El segundo tema presentado en AR es sobre Bidireccional Directo

Convertidores de corriente (CC)-CC conocidos como Buck-Boost; el objetivo es estudiar su comportamiento cuando se intercambia energía entre dos sistemas, normalmente fotovoltaicos, vehículos eléctricos y sistemas de almacenamiento.

Las actitudes de los estudiantes hacia las aplicaciones de AR fueron significativamente más altas que las de la enseñanza tradicional. Las medidas de las habilidades desarrolladas indican un mejor rendimiento cognitivo cuando se

utiliza la tecnología AR. Ambas herramientas AR diseñadas se utilizaron en un curso de la industria para explorar su opinión, proporcionando comentarios valiosos.

Palabras clave: realidad aumentada, simulación de circuitos electrónicos, educación en ingeniería, aprendizaje móvil, Unity, Vuforia.

CAPÍTULO I

INTRODUCCIÓN

1.1. INTRODUCCIÓN

La formación-enseñanza particularmente en las universidades debe abordar diversos desafíos principalmente en ingeniería debido al equipo de laboratorio costoso o insuficiente o la dificultad para simular ciertas condiciones experimentales.

El desafío en la formación y docencia está en promover las competencias profesionales y el incremento progresivo mediante el uso de recursos tecnológicos y educativos como las Tecnologías de la Información y las Comunicaciones (TIC). Estos materiales de aprendizaje digital deben ser efectivos y de alta calidad para la formación y prácticas de laboratorio a través de una perspectiva diferente de representación de contenidos, fenómenos y situaciones simples o complejas para abordar en un escenario que facilita el aprendizaje en áreas de la ingeniería.

La situación provocada por la pandemia del COVID19 ha provocado cambios y desafíos en la formación universitaria y restricciones en las actividades, poniendo así a prueba la capacidad de las universidades para adaptar estos cambios, y ha revivido fuertemente el enfoque del aprendizaje basado en competencias en un nuevo contexto. Por lo tanto, las TIC juegan un papel importante en el desarrollo de habilidades y la adquisición de conocimientos. Facilitar el intercambio de información y aumentar el acceso a contenidos diversos.

Antes de la pandemia ya se han utilizado algunas TIC que ayudan a mejorar el aprendizaje, unas implementan las estrategias con el uso de Internet de las Cosas (IoT) [1]-[3]. Mylonas en [1] reporta un kit-lab basado en IoT para motivar actividades

de ahorro energético mediante la integración de gamificación, componentes y sensores electrónicos, mientras Márquez [2] y Jiménez[3] exploran la utilidad IoT en el contexto académico. Mancilla [4] desarrolla un dispositivo IoT llamado Mecatrónica e Internet de las Cosas (MEIoT) estación meteorológica, combinando las habilidades necesarias en Mecatrónica Educativa para la Industria 4.0. Simiscuka [5] presenta un entorno de Realidad Virtual (VR) combinado con dispositivos IoT que permiten a usuarios operar objetos en un entorno virtual y objetos IoT reales para realizar algún laboratorio prácticas.

Los entornos de VR se han utilizado para la formación-enseñanza y también se ha demostrado su eficacia en los enfoques de aprendizaje. Simulador VR para entrenamiento de incendios con la adquisición de conceptos y entrenamiento procedimental se propone en [6]. Las características técnicas y detalles de instalación del sistema de instalaciones fotovoltaicas son proyectadas en un mundo virtual por [7]. Mientras que la capacitación de trabajadores de línea en el sistema de distribución de energía se presenta en [8], mientras que la evaluación en un entorno industrial real con la inclusión de VR se analiza en [9]. Radianti presenta una revisión de investigaciones existentes dedicadas a la aplicación de VR inmersiva en la educación superior en muchas disciplinas, donde la ingeniería fue la más popular identificada en un 24 % [10]. Del mismo modo, como se presenta en [11]-[12]. Sin embargo, la creación de contenido tiende a ser costosa debido a la compra de cascos y equipos informáticos especializados.

1.2. TRABAJO RELACIONADO

La tecnología de Realidad Aumentada se ha utilizado cada vez más en la educación. Los efectos de aprendizaje pueden ser más interesante si un objeto virtual se fusiona con un escenario real con AR y el comportamiento interactivo de los alumnos es el mismo que el de la condición experimental real, incluso en la interactividad en tiempo real bajo el marco de Teaching Factory entre el equipo industrial y AR propuesta como se sugiere en [13]. Wei [14] investiga los impactos

de AR en la enseñanza en cursos de diseño técnico creativo en alta escuelas para mejorar la motivación, la creatividad y la eficiencia docente, en deportes [15], matemáticas [16]-[17], ingeniería [18]-[19], industria [20] con clientes que se convierten en parte del proceso de fabricación en la industria de fabricación de células robóticas para personalizar sus necesidades [21], o guiar a los operadores en la industria [22]-[23].

Las tecnologías AR se han utilizado en el campo de la electrónica para ayudar a comprender conceptos y sus aplicaciones en laboratorios para aumentar las habilidades y la comprensión de algunos temas. [24–27] presentó un simulador de circuito eléctrico de resistencia mixta en serie-paralelo con componentes AR para enseñar conceptos de redes eléctricas a estudiantes de ingeniería de entre 20 y 21 años. De manera similar, [28] desarrolla una aplicación AR para enseñar a los estudiantes de quinto grado sobre las corrientes eléctricas en las bombillas.

Se presenta una aplicación móvil AR-Electronic-Lab para brindar una experiencia de circuitos electrónicos en la vida física real, creando algunas prácticas de laboratorio con el uso de un osciloscopio, un circuito recortador de diodos, y un circuito amplificador operacional inversor [29]. El entrenamiento para operar un osciloscopio y un generador de función se presenta en AR por [30], luego se convierte en VR [31].

Los conceptos fundamentales de los dispositivos electrónicos digitales, como el álgebra booleana y las puertas lógicas, se destacan en AR propuesto por [32] para ayudar en la identificación de 7 puertas lógicas básicas AND gate, OR gate, NOT gate, NAND gate, NOR puerta, puerta XOR y puerta XNOR. [33] describe el diseño e implementación de un AR con IoT para controlar y monitorear sistemas mecatrónicos.

Teniendo en cuenta la necesidad de desarrollar infraestructura experimental y teniendo en cuenta el trabajo relacionado, existe un amplio campo de mejora en la formación-enseñanza de la electrónica. Por esta razón, las siguientes herramientas

de aprendizaje basadas en electrónica AR se proponen para principiantes. Discutimos el diseño y la implementación de circuitos AR-RLC y un convertidor DC-DC bidireccional conocido como Buck-Boost; la herramienta propuesta se utiliza en cursos de electrónica de potencia para estudiantes universitarios y en un curso de capacitación en la industria.

CAPÍTULO 2

MARCO TEÓRICO

¿QUÉ ES UNITY 3D?

Unity es mucho más que una plataforma de desarrollo en tiempo real: también es un ecosistema robusto diseñado para ayudarte a tener éxito. Con una comunidad dinámica de creadores donde podrás encontrar lo que necesitas para hacer realidad tu visión.

2.1 INSTALACIÓN DE UNITY 3D

Cómo instalar motor Unity y herramientas para distintas plataformas

2.1.1 Descargar e instalar Unity HUB

Unity HUB es una aplicación que permite gestionar las distintas versiones de Unity, sus módulos y los proyectos que hagamos. Unity HUB se puede descargar ingresando en este enlace: <https://unity3d.com/get-unity/download>. La siguiente imagen es una captura del enlace anterior, al pulsar el botón «Download Unity Hub» comienza la descarga de la aplicación.

Download Unity

Welcome! You're here because you want to download Unity, the world's most popular development platform for creating 2D and 3D multiplatform games and interactive experiences.

Before you download choose the version of Unity that's right for you.

Choose your Unity + download

Download Unity Hub

[Learn more about the new Unity Hub here.](#)

Figura. 1. En el enlace de arriba se pueden descargar la aplicación Unity HUB.

2.1.2 Instalar versión del motor Unity

Una vez instalado Unity HUB lo abrimos como se observa en la figura 1. Y vamos a la sección «Installs» donde podemos instalar nuevas versiones de motor Unity como se observa en la figura 2.

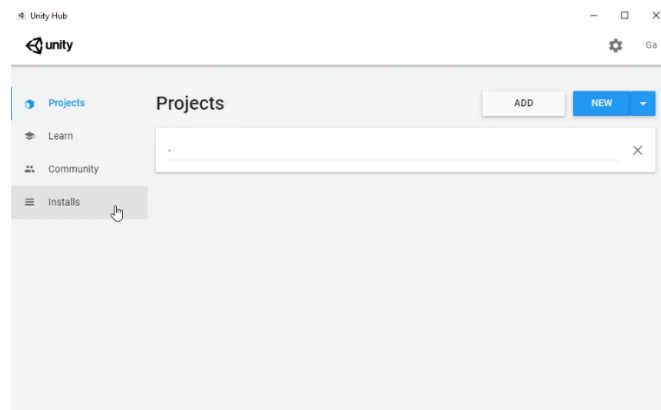


Figura. 2. En la sección «Installs» se pueden añadir nuevas versiones de Unity y agregar módulos a versiones ya instaladas.

En la sección «Installs» se encuentran las versiones ya instaladas del motor, podemos eliminarlas, añadirle módulos y también agregar nuevas versiones de Unity utilizando el botón «Add» como se observa en la figura 3.

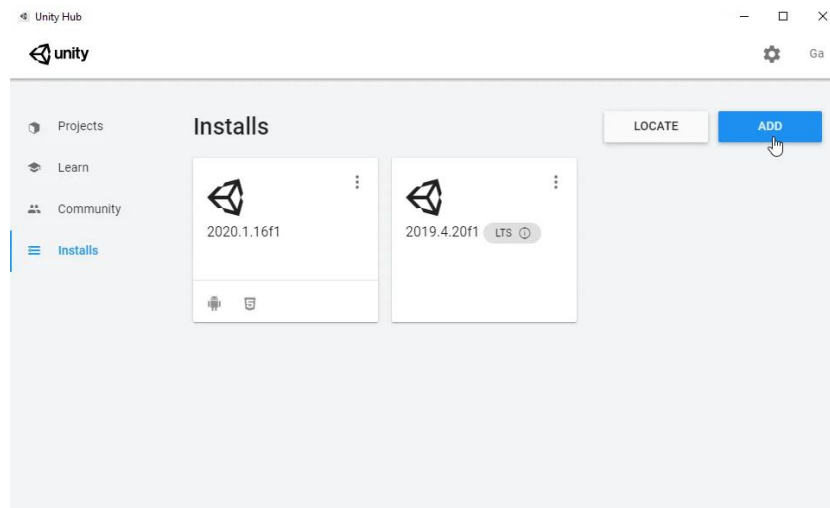


Figura. 3. Con el botón Add se puede instalar una nueva versión del motor Unity.

Esto nos despliega una ventana en la que se puede elegir la versión del motor (véase figura 4). Las versiones que tienen «(LTS)» son las versiones más estables del motor, usualmente siempre instalada la última versión oficial de Unity y además la última versión LTS, si es un proyecto importante utilizo esta última.

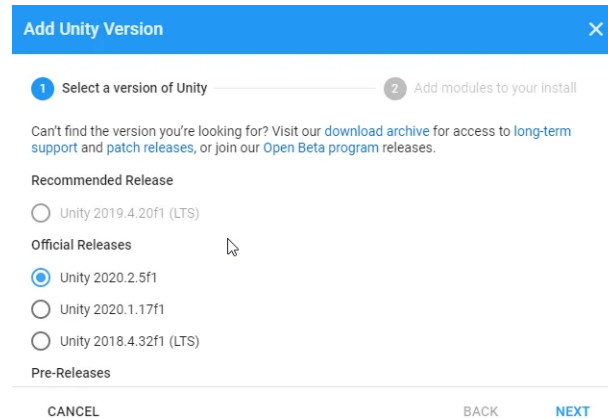


Figura. 4. En esta ventana se selecciona la versión del motor Unity que se desea instalar.

2.1.3 Añadir módulos a la versión de Unity

Los módulos extienden la funcionalidad de Unity y se pueden añadir en cualquier momento, no necesariamente al momento de instalar Unity como se observa en la figura 5.

El Visual Studio es el editor de Scripts oficial que se utiliza con Unity, aunque se pueden utilizar otros editores si se lo prefiere.

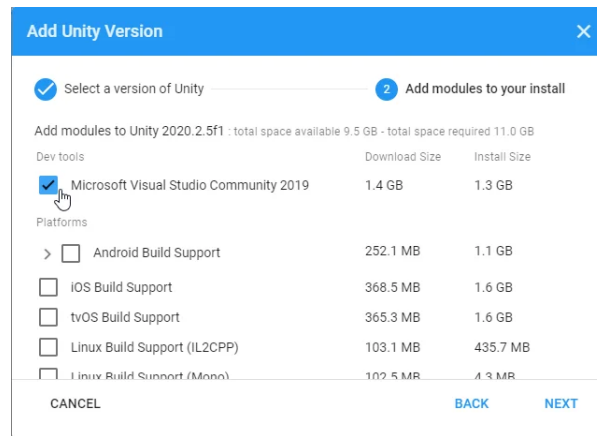


Figura. 5. Ventana de selección de módulos para añadir al motor Unity. Módulo Visual Studio para editar Scripts seleccionado.

Dependiendo de nuestra plataforma objetivo serán los módulos que tengamos que instalar, por defecto el motor exporta para el sistema operativo Windows.

Si se quiere exportar para móviles Android y publicar juegos y aplicaciones en Google Play se necesitan los módulos que se ven en la figura 6. Android Build Support, Android SDK & NDK Tools y el módulo OpenJDK. Si se quiere exportar juegos y aplicaciones que funcionen en navegadores, en páginas como Itch.io o en una página web de su propiedad, se necesita instalar el módulo WebGL Build Support seleccionado en la figura 7.

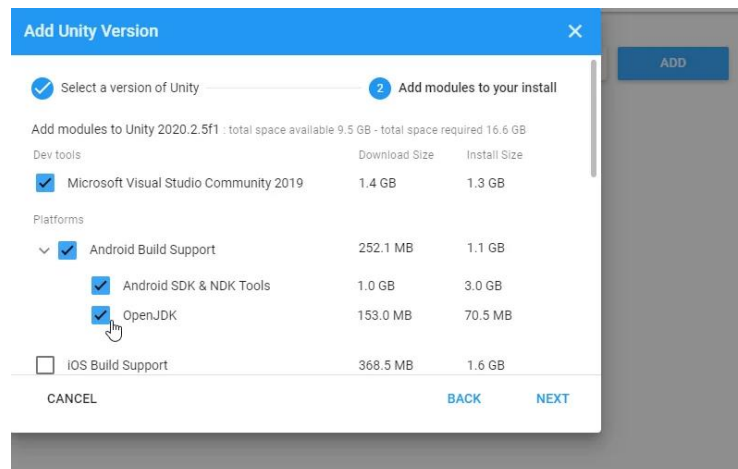


Figura. 6. Módulo Android para exportar juegos y aplicaciones Android con Unity.

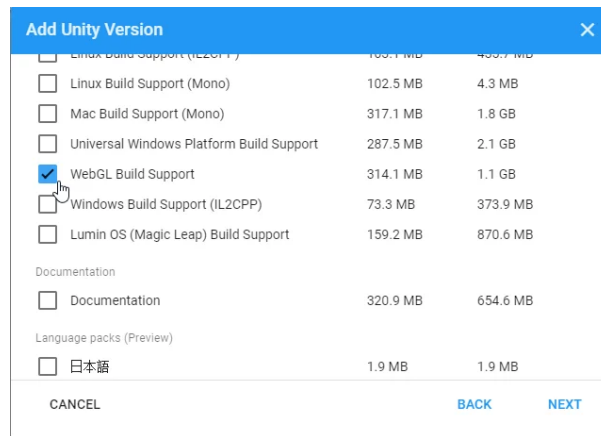


Figura. 7. Módulo WebGL para exportar juegos y aplicaciones para navegadores con Unity.

Una vez que se eligen los módulos, se pulsa el botón Next, se aceptan los términos y condiciones y la instalación comienza

2.1.4 Finalizar instalación y crear nuevo proyecto

Cuando la instalación termina, en la pestaña «Installs» nos aparece la nueva versión (figura 3), haciendo clic en los tres puntitos de la esquina superior derecha de cada versión se pueden añadir nuevos módulos o eliminar la versión.

Ahora si vamos a la sección «Projects» podemos crear un proyecto nuevo utilizando el botón «New» o la flecha que se encuentra a la derecha (figura 8), esta última acción nos permite elegir el motor para crear el nuevo proyecto, como se observa en la figura 9.

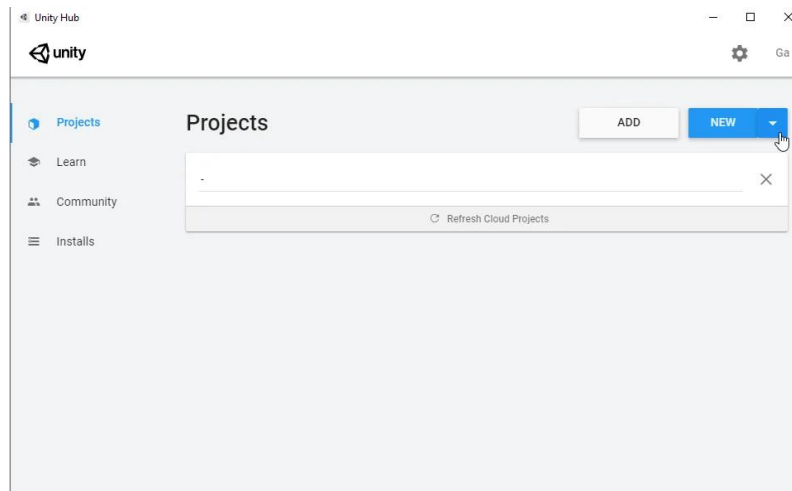


Figura. 8. Ventana de gestión de proyectos Unity.

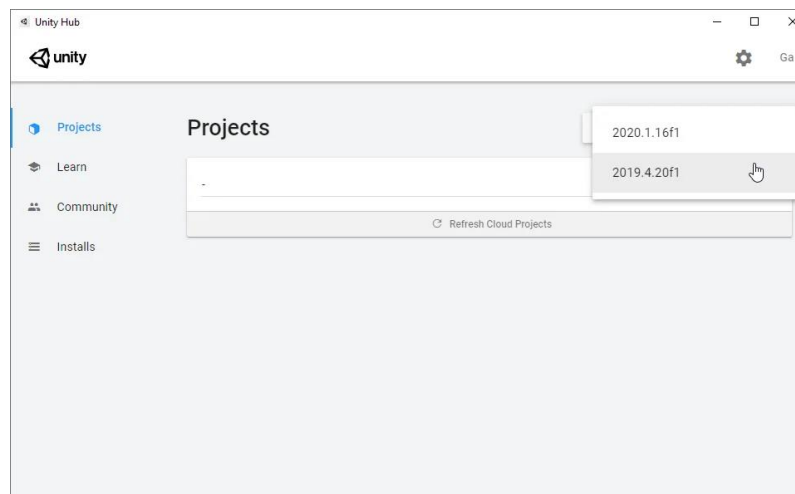


Figura. 9. Botón para crear un nuevo proyecto seleccionando la versión del motor Unity a utilizar.

Al crear un nuevo proyecto se abre la ventana que se ve en la figura 10, donde podemos darle un nombre al proyecto, elegir la plantilla que más se adapte a lo que queramos hacer, elegir la ubicación de los archivos del proyecto y finalmente crear el proyecto con el botón «Create».

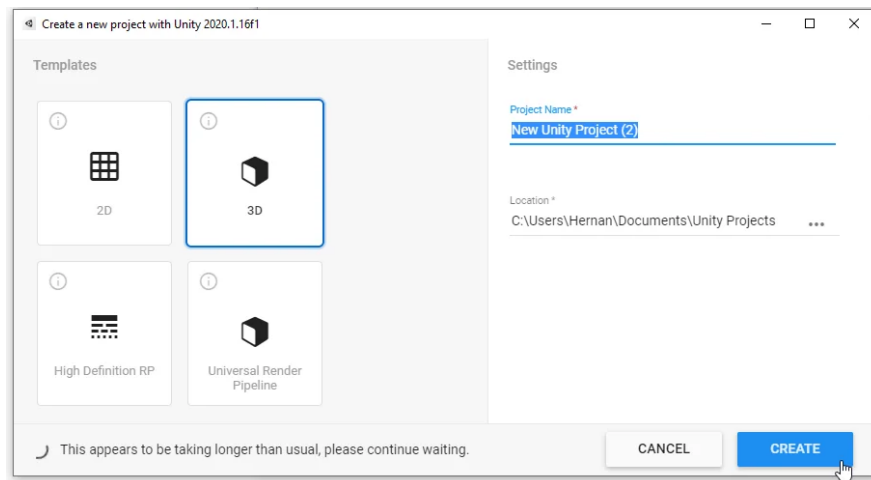


Figura. 10. Configuración inicial de un proyecto en Unity.

2.2 INTEGRACIÓN DE ARCORE

AR Core de Google.

2.2.1 ¿Qué es ARCore?

ARCore es la plataforma de Google para crear experiencias de realidad aumentada. Usando diferentes API, ARCore permite que su teléfono detecte su entorno, comprenda el mundo e interactúe con la información. Algunas de las API están disponibles en Android e iOS para permitir experiencias de RA compartidas.

ARCore utiliza tres capacidades clave para integrar contenido virtual con el mundo real como se ve a través de la cámara de su teléfono:

- El seguimiento del movimiento permite que el teléfono para entender y realizar un seguimiento de su posición en relación con el mundo.
- La comprensión del medio ambiente permite que el teléfono para detectar el tamaño y la ubicación de todo tipo de superficies: superficies horizontales, verticales y en ángulo como el suelo, una mesa de café o paredes.
- La estimación de la luz permite que el teléfono para estimar las condiciones actuales de iluminación del entorno.

2.2.2 ¿Cómo funciona ARCore?

Básicamente, ARCore está haciendo dos cosas: rastrear la posición del dispositivo móvil mientras se mueve y desarrollar su propia comprensión del mundo real.

La tecnología de seguimiento de movimiento de ARCore utiliza la cámara del teléfono para identificar puntos interesantes, denominados características, y rastrea cómo se mueven esos puntos a lo largo del tiempo. Con una combinación del movimiento de estos puntos y las lecturas de los sensores inerciales del teléfono, ARCore determina tanto la posición como la orientación del teléfono a medida que se mueve por el espacio.

Además de identificar puntos clave, ARCore puede detectar superficies planas, como una mesa o el piso, y también puede estimar la iluminación promedio en el área a su alrededor. Estas capacidades se combinan para permitir que ARCore construya su propia comprensión del mundo que lo rodea.

La comprensión de ARCore del mundo real le permite colocar objetos, anotaciones u otra información de una manera que se integra perfectamente con el mundo real.

ARCore proporciona SDK para muchos de los entornos de desarrollo más populares. Estos SDK proporcionan API nativas para todas las funciones esenciales de AR, como el seguimiento del movimiento, la comprensión del entorno y la estimación de la luz. Con estas capacidades, puede crear experiencias de RA completamente nuevas o mejorar las aplicaciones existentes con funciones de RA.

-Hardware necesario para poder usar el módulo de ARCore.

ARCore está diseñado para funcionar en una amplia variedad de teléfonos Android calificados que ejecutan Android 7.0 (Nougat) y versiones posteriores. También se precisa que el dispositivo móvil cuente con un sensor denominado giroscopio (no confundir con acelerómetro).

2.2.3 ¿Por qué se opta por ARCore?

Al momento de elegir una herramienta, no se puede hacer un juicio en cuanto a cuál es “la mejor”, ya que esto será siempre subjetivo; siempre hay que hacer una elección basándonos en los requerimientos del proyecto y si ambas opciones cubren estos, se procede a escoger la herramienta que más se le facilite implementar al desarrollador.

Se opta por usar ARCore debido a que este mantiene mapas más grandes en cuanto al seguimiento del entorno de la vida real en comparativa con otros SDK de Realidad Aumentada como Vuforia; sin dejar de lado la documentación y el soporte existente, tanto por parte de Google y de la comunidad en sí. Agregado a lo anterior, ARCore es totalmente gratis y la configuración e integración de ARCore en Unity es mucho más sencilla y no requiere de descargas ni registros adicionales. Ver figura 11.

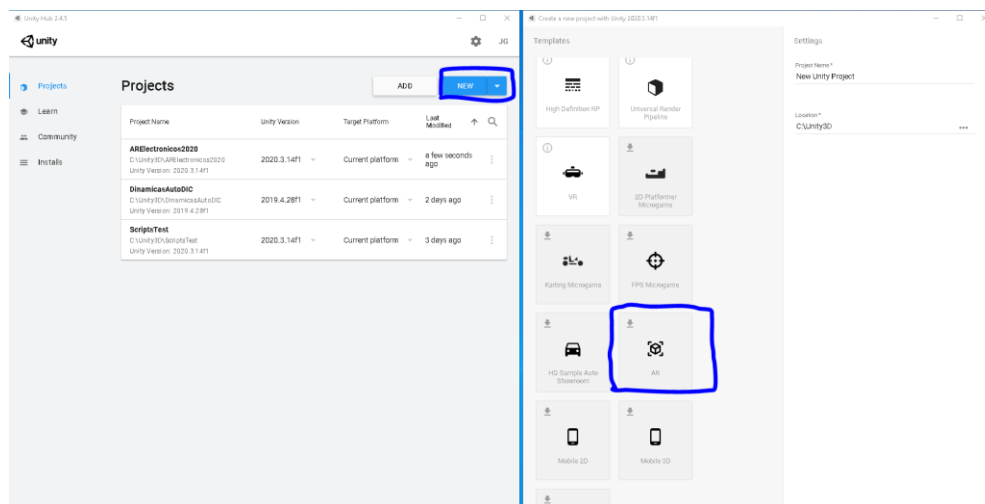


Figura. 11. Configuración de AR Core para Unity

En 3 sencillos pasos tendremos nuestro proyecto creado y configurado con ARFoundation (implementación de ARCore para Unity3D), sin necesidad de hacer ninguna clase de registro en línea ni estar configurando archivos del tipo JSON (en el caso de Vuforia).

2.3 MODELADO 3D BLENDER

2.3.1 Introducción al modelado en Blender

La creación de una escena 3D necesita al menos tres componentes clave: modelos, materiales y luces, estos componentes los encontramos en la ventana principal como se observa en la figura 12. En esta parte, se trata el primero de ellos, que es el modelado. El modelado es simplemente el arte y la ciencia de crear una superficie que imita la forma de un objeto del mundo real o expresa su imaginación de objetos abstractos.

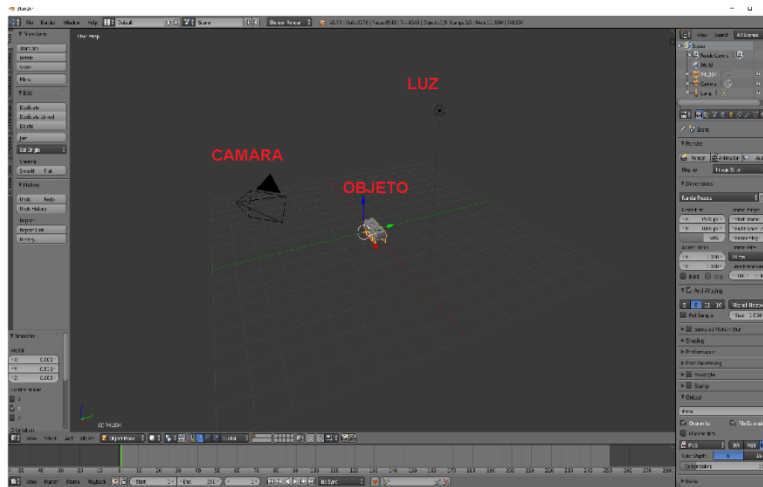


Figura. 12. Ventana principal de Blender.

2.3.2 Modos

Dependiendo del tipo de objeto que intente modelar, existen diferentes tipos de modos de modelado (figura 13) . Dado que los modos no son específicos del modelado, se tratan en diferentes partes del manual.

Es común cambiar entre modos durante el modelado. Algunas herramientas pueden estar disponibles en más de un modo, mientras que otras pueden ser exclusivas de un modo en particular.

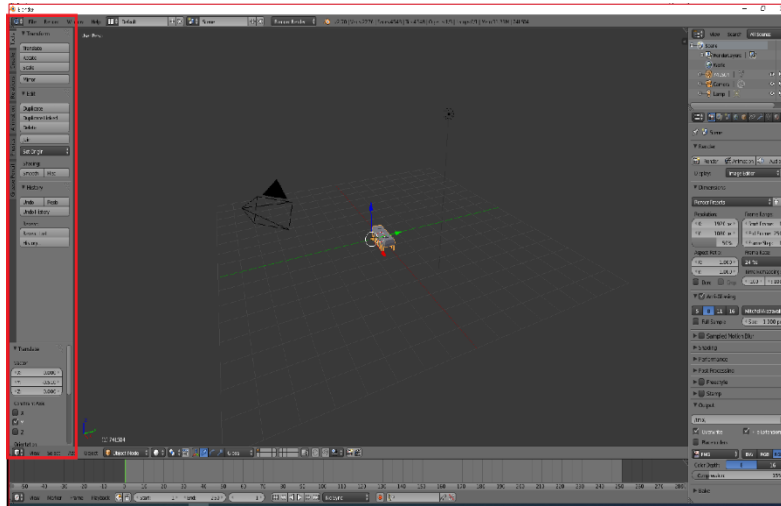


Figura. 13. Modos en la edición de modelos 3D en Blender.

2.3.3 Modo de edición

El modo de edición es el modo principal en el que se realiza el modelado. El modo de edición se utiliza para editar los siguientes tipos de objetos:

- Mallas
- Curvas
- Superficies
- Metaballs
- Objetos de texto
- Enrejado

Solo puede modificar la malla de los objetos que está editando. Para modificar otros objetos, puede salir del modo de edición, seleccionar otro objeto y entrar en el modo de edición, o utilizar la edición de objetos múltiples .

- Exportar archivos a formato FBX para ser usado en Unity 3D.

1. Dejar solamente el objeto 3D a exportar y seleccionarlo (figura 14).

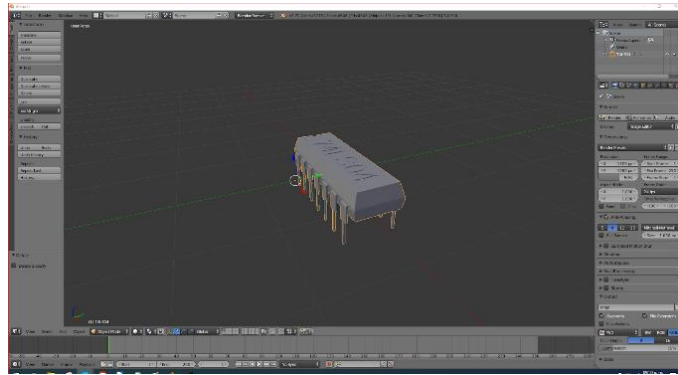


Figura. 14. Objeto 3D a exportar.

2. Seleccione Archivo → Exportar → Autodesk (FBX). Véase figura 15.

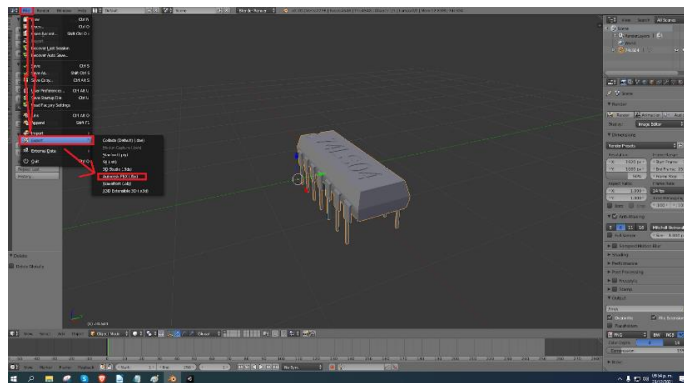


Figura. 15. Ventanas para exportar el modelo 3D.

3. Se abrirá una ventana de dialogo para confirmar el directorio donde se guardar el archivo en FBX, así como el nombre que se le desea asignar (figura 16).

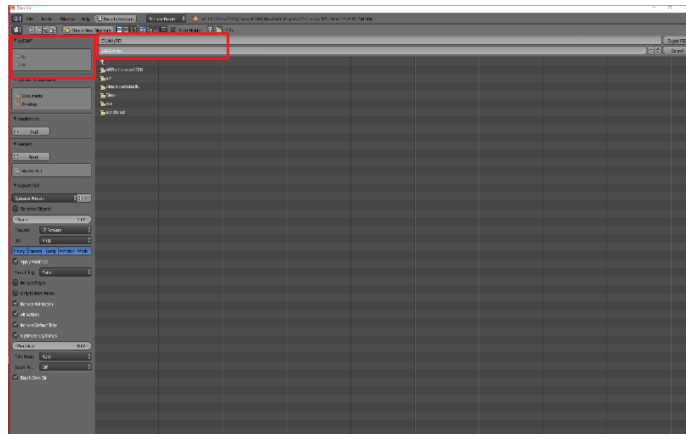


Figura. 16. Ventana de dialogo para confirmar el directorio donde se guardar el archivo en FBX

2.4 ENSAMBLE DE MODELOS 3D

2.4.1 Agregar componentes al espacio 3d

Para importar un modelo 3D a Unity puedes arrastrar un archivo y soltarlo en la ventana Project.

En la pestaña Inspector > Model Unity soporta la importación de modelos desde las aplicaciones 3D más populares.

Para páginas acerca de una importación específica de paquetes 3D, ver las siguientes:

- Maya
- Cinema 4D
- 3ds Max
- Cheetah3D
- Modo
- Lightwave
- Blender
- SketchUp

2.4.2 Texturas

Almacena las texturas en un directorio llamado Textures junto al mesh exportado dentro de tu proyecto Unity. Esto permitirá que Unity encuentre la textura, y la conecte automáticamente a los materiales generados.

2.4.3 Archivos 3D exportados

Unity puede leer archivos .FBX, .dae (Collada), .3DS, .dxf y .obj, los exportadores FBX se pueden encontrar aquí y exportadores obj o Collada se pueden encontrar en muchas aplicaciones.

2.4.4 Modelos

Los archivos modelo que son colocados en la carpeta Assets en su proyecto Unity son automáticamente importados y almacenados como assets de Unity.

Un archivo modelo puede contener un modelo 3D, como lo es un personaje, un edificio, o un mueble. El modelo es importado en múltiples assets. En la Vista de Proyecto, el objeto principal importado es un Modelo prefabricado. Usualmente hay también hasta varios objetos Mesh que son referenciados por el Modelo Prefabricado.

2.4.5 Posicionar protoboard

Una vez importado nuestro modelo del protoboard vamos a posicionarlo y escalarlo para poder ejecutar un ensamble con los componentes como se ve en las figuras 17 y 18, para que nos permita una mayor coincidencia entre las terminales del protoboard y los pines de nuestros modelos 3D de los componentes.

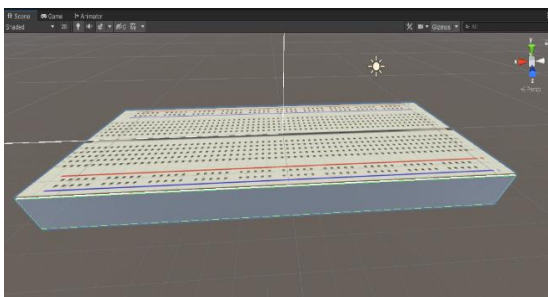


Figura. 18. Protoboard posicionado en el espacio 3D en Unity.

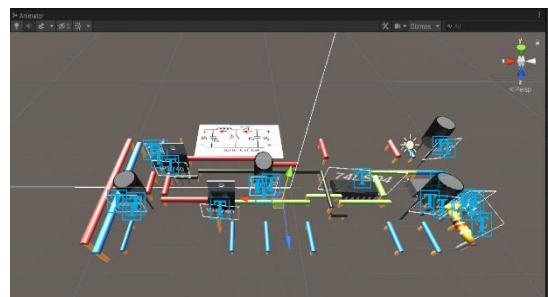


Figura. 17. Componentes requeridos en las practicas posicionados en el espacio 3D en Unity

2.4.6 Colocar modelos en posiciones coincidentes

Para este paso debemos de manipular los componentes individualmente y acercarlos de una manera muy precisa hacia el modelo 3d del protoboard, siempre revisando que exista una coincidencia entre la terminal y el pin del componente

2.4.7 Revisar coincidencias

Por último, una vez que hemos ensamblado por completo nuestros componentes realizamos un barrido a detalle de que realmente las coincidencias se den como se ve en el ejemplo de la figura 19, de lo contrario debemos ajustar el componente que se pueda encontrar desfasado

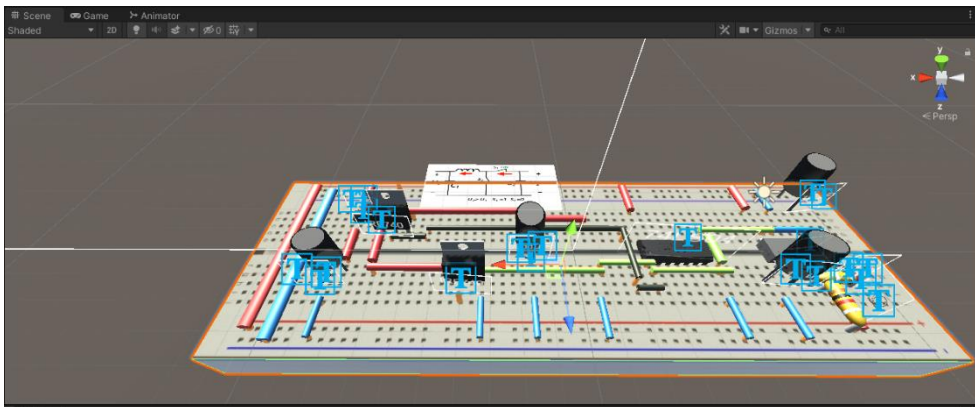


Figura. 19. Coincidencia de los pines de componentes con el protoboard

2.5 CREACIÓN DE ANIMACIONES CON TIMELINE

Para este caso utilizamos el secuenciador de Unity 3D (Time Line) para la activación de conjuntos de objetos animados, en conjunto con la herramienta de animación (esta para elementos individuales)

2.5.1 Ventana de Timeline

Seleccione la opción Window → Sequencing → Timeline.

I. Seleccione el objeto al cual desea animar la secuencia y seleccione la opción "Create" dentro de la pestaña Timeline. Se abrirá un cuadro de dialogo para guardar la secuencia de Timeline como se ve en la figura 20.

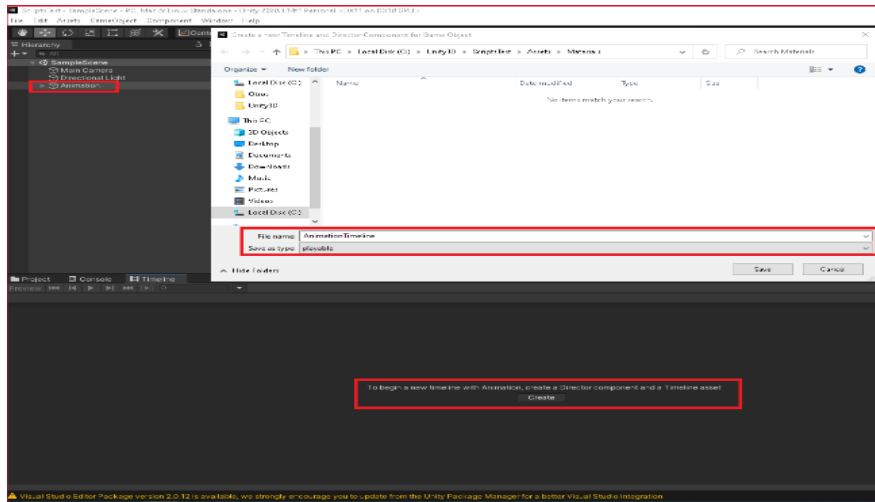


Figura 20. Guardando secuencia de Timeline para un objeto 3D

II. Arrastre los objetos que desea activar uno por uno, a la pestaña de Timeline y seleccione la opción "Activation Track"; subsecuentemente redimensione la barra que representa a cada objeto activo para tenga una duración conforme a sus requisitos, observe figura 21 .

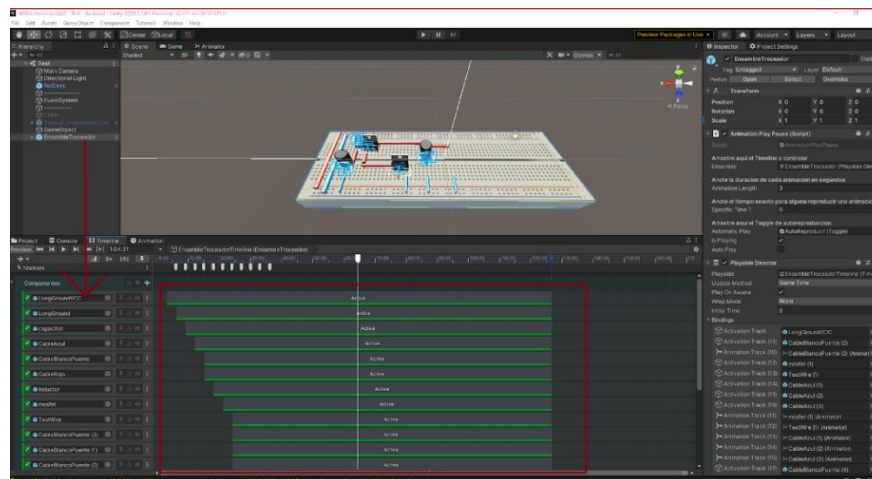


Figura 21. Pestaña "Activation Track" en Timeline.

A partir de este punto se puede proceder a animar los objetos a activar en la secuencia, de una manera individual para que cumplan con las expectativas del creador de la secuencia y animación.

2.5.2 Ventana de animación

Esta ventana brinda herramientas para crear clips de animación. Puedes animar en 3D, 2D, materiales, variables, componentes, etc. El clip de animación está disponible para algún objeto en específico.

La ventana de animación se encuentra en la pestaña Window → Animation → Animation (véase figura 22).

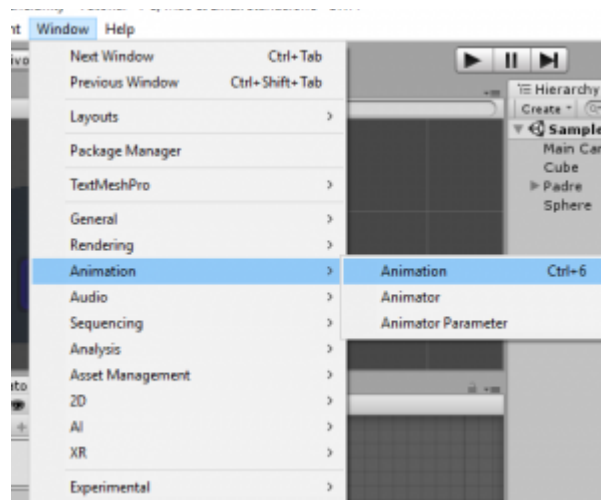


Figura. 22. Ingresar a la ventana de animación de Unity.

En la ventana de animación de Unity, puedes crear y editar diversos clips de animación como se ve en la figura 23. Para unir esos clips, usa la ventana de Timeline.

Cada clip de animación se guarda en la carpeta en la que se encuentra la ventana Project, al momento de crear la animación.

El clip de animación es una propiedad del Game Object que tengas seleccionado en la jerarquía. Ese clip solo estará disponible para el game object seleccionado.

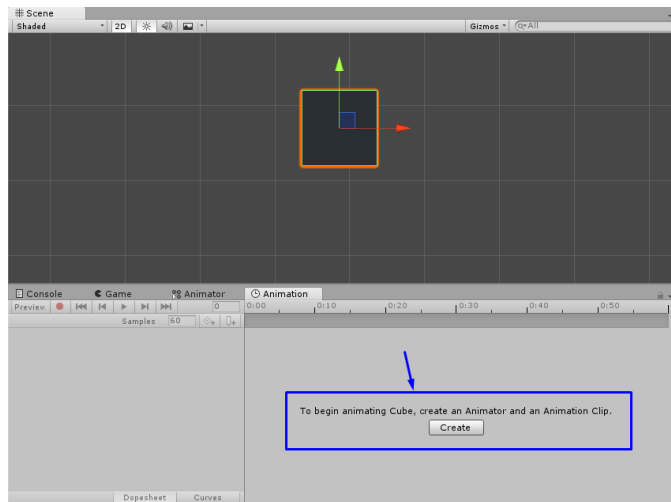


Figura. 23. Creando un nuevo animator y clip de animacion.

2.5.3 Creando un nuevo clip.

Para esto solo presiona el botón de crear nuevo clip. Puedes elegir la carpeta de destino y el nombre de tu clip de animación (figura 24).

Se recomienda crear una carpeta llamada Animaciones, solo para tener un orden dentro de los archivos de tu proyecto de Unity.

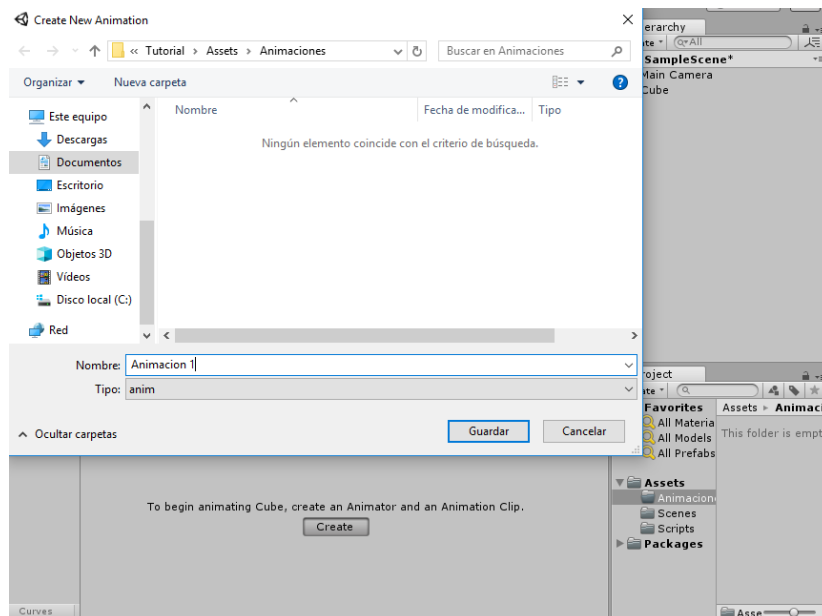


Figura. 24. Elegir la carpeta de destino y el nombre de tu clip de animación.

2.5.4 Herramientas de la ventana de animación

Al crear te aparecerá un clip de animación vacío. Como se muestra en la figura 25.



Figura. 25. Clip de animación vacío

2.5.5 Herramientas de la ventana de animación de unity

Preview, esta herramienta solo indica si quieres ver una vista previa de la animación (figura 26). Se activa sola al momento de grabar la animación.

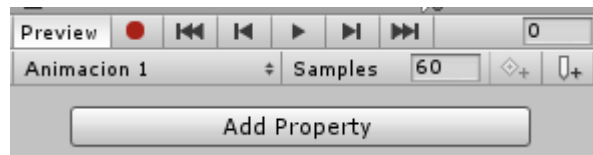


Figura. 26. Boton "Preview" de la ventana de animacion.

El botón de record, cuando se activa este botón puedes comenzar a realizar la grabación de tu animación. Ten en cuenta que todo lo que le hagas a tu objeto y a la línea de tiempo se va a guardar.

Recuerda siempre desactivar este botón una vez que termines de hacer tu clip de animación. Ya que si no lo desactivas puedes modificar la animación que ya tenías.

Botones de play, pause, etc. Con estos botones puedes visualizar tu animación.

El cuadro de la derecha indica la frame a la que se encuentra la animación en ese momento como se ve en la figura 27. Puedes teclear alguna frame para ir a esa posición.

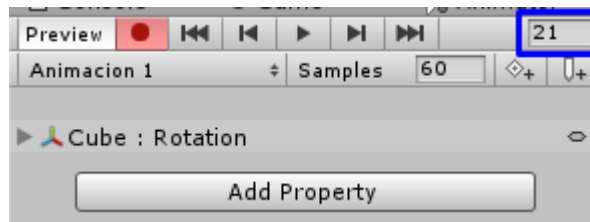


Figura. 27. Visualización del frame en cada parte de la animación.

Puedes crear múltiples animaciones para el mismo objeto, solo busca la opción donde está el nombre de tu animación y presiona crear nueva (figura 28).

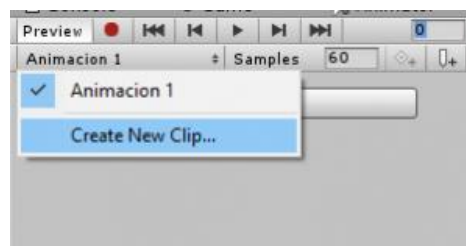


Figura. 28. Crear múltiples animaciones para el mismo objeto.

Samples, indica cuantas frames son por un segundo.

Añadir keyframe, esta opción puedes añadir una nueva keyframe en la línea de tiempo de la animación, observe figura 29.

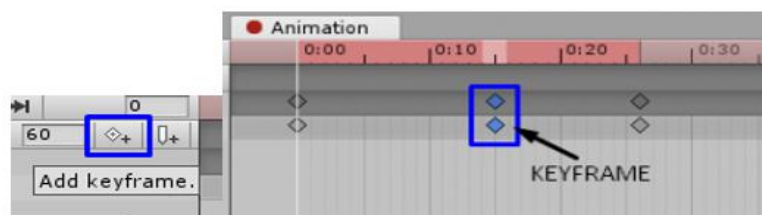


Figura. 29. Añadir una nueva keyframe en la línea de tiempo de la animación.

Añadir un evento (figura 30). Con esta opción puedes agregar un método de un script.

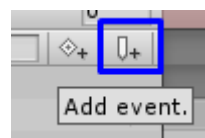


Figura. 30. Añadir un evento.

En la parte inferior derecha, puedes elegir la visualización de la animación como se ve en la figura 31. Si por línea de tiempo o por curvas.

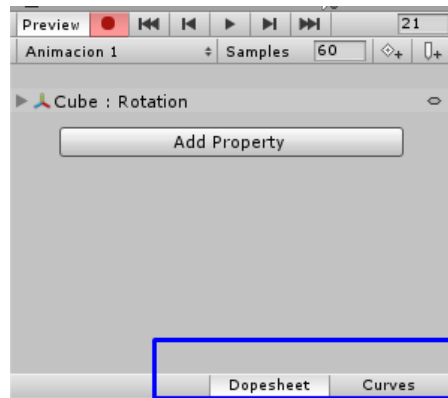


Figura. 31. Tipo de visualización de la animación

2.6 CREACIÓN DE UIs

2.6.1 Diseño básico

En esta sección nosotros miraremos cómo se pueden posicionar los elementos UI relativos al Canvas y cada uno. Si usted quiere probarlo usted mismo mientras lee, usted puede crear una Imagen utilizando el menú GameObject -> UI -> Image.

2.6.2 La Rect tool

Cada elemento UI es representado como un rectángulo por propósitos de diseño. Este rectángulo puede ser manipulado en la Vista de Escena utilizando la Rect Tool en la barra de herramientas (figura 32). La Rect Tool es utilizado para ambas las características 2D de Unity y UI, de hecho, puede ser incluso utilizado para objetos 3D también.



Figura. 32. Botones de la pestaña de herramientas con la Rect Tool seleccionada.

La Rect Tool puede ser utilizado para mover, cambiar el tamaño y rotar los elementos UI. Una vez usted haya seleccionado un elemento UI, usted lo puede

mover al hacer clic en cualquier parte dentro del rectángulo y arrastrándolo. Usted puede cambiar el tamaño al hacer clic en los bordes y esquinas y arrastrando. El elemento puede rotar al mover el cursor un poco lejos de las esquinas hasta que el cursor del mouse se ve como un símbolo de rotación. Usted puede hacer clic y arrastrar ya sea en cualquier dirección para rotar.

Tal como cualquier otra herramienta, la Rect Tool utiliza el modo de pivote actual y espacio, configurado en la barra de herramientas. Cuando trabaje con UI es buena idea mantener estos configurados a Pivot y Local como se ve en la figura 33.

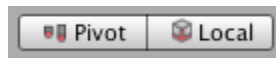


Figura. 33. Botones de la barra de herramientas configuradas a Pivot y Local.

2.6.3 Rect transform

El Rect Transform es un nuevo componente transform que es utilizado para todos los elementos UI en vez de un componente Transform regular véase figura 34.

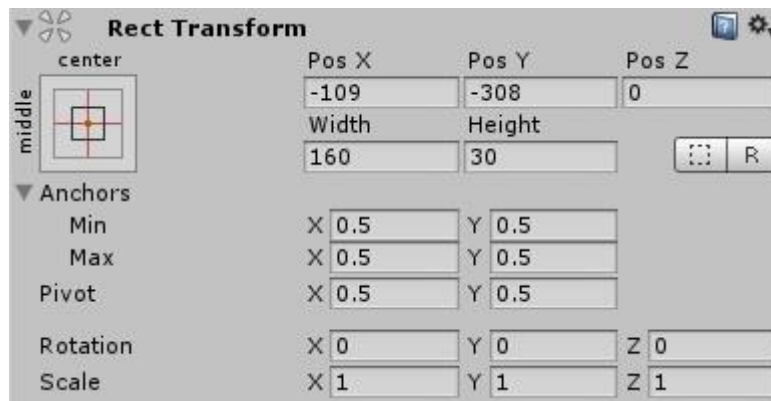


Figura. 34. Componente Rect Transform, característico de los elementos UI

Los Rect Transforms tienen posición, rotación, y escala justo como cualquier Transform regular, pero éste también tiene un ancho y altura, utilizado para especificar las dimensiones del rectángulo.

2.6.4 Cambiar el Tamaño Versus Escalando

Cuando la Rect Tool es utilizada para cambiar el tamaño de un objeto, normalmente para Sprites en el sistema 2D y para objetos 3D éste va a cambiar la *escala* local del objeto. Sin embargo, cuando es utilizado en un objeto con un Rect Transform en él, éste más bien va a cambiar el ancho y la altura, manteniendo la escala local sin cambiar. Este cambio de tamaño no va a afectar el tamaño del fondo, borde en las imágenes cortadas , y así

2.6.5 Pivot (Pivote)

Las modificaciones de rotación, tamaño y escala ocurren alrededor del pivote por lo que esa posición del pivote afecta el resultado de la rotación, el cambio de tamaño y la escala. Cuando el botón Pivot de la barra de herramientas es configurado a modo Pivote, el pivot de un Rect Transform puede ser movido en la vista de escena, observe figura 35.

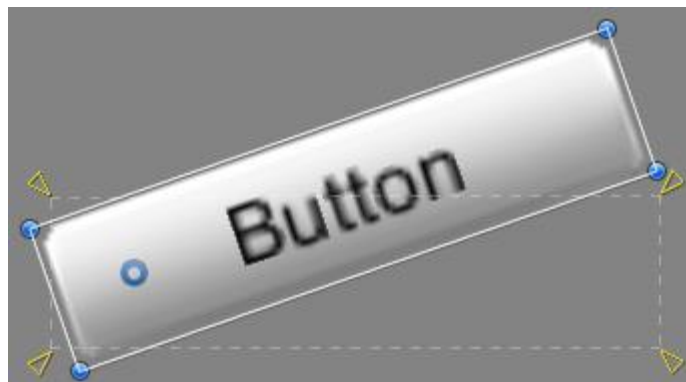


Figura. 35. Pivote

2.6.6 Anclajes

Los Rect Transforms incluyen un concepto de diseño llamado **anchors**. Los Anchors son mostrados como cuatro manijas pequeñas con figura de triángulos en la Vista de Escena y la información de anclaje es también mostrada en el Inspector.

Si el padre de un Rect Transform también es un Rect Transform, el Rect Transform hijo puede ser anclado al Rect Transform padre en varias maneras. Por ejemplo, el hijo puede ser anclado al centro del padre, o a una de las esquinas.

Elemento UI anclado al centro del padre. El elemento se mantiene en un desfase fijo al centro como se observa en la figura 36.

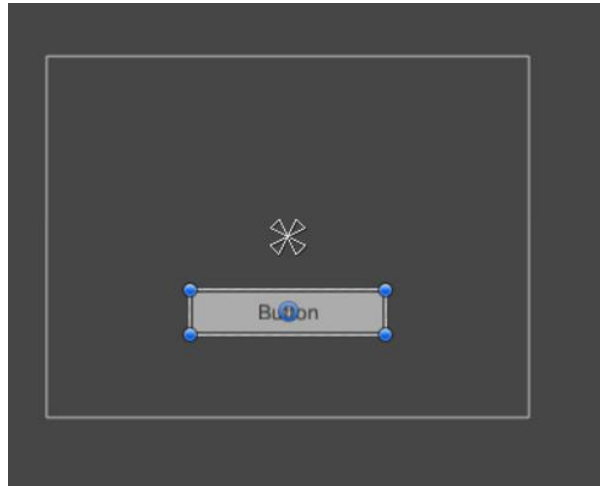


Figura. 36. Los anclajes de UI

El elemento UI anclado a la esquina inferior derecha del padre. El elemento contiene un desfase fijo a la esquina inferior derecha (figura 37).

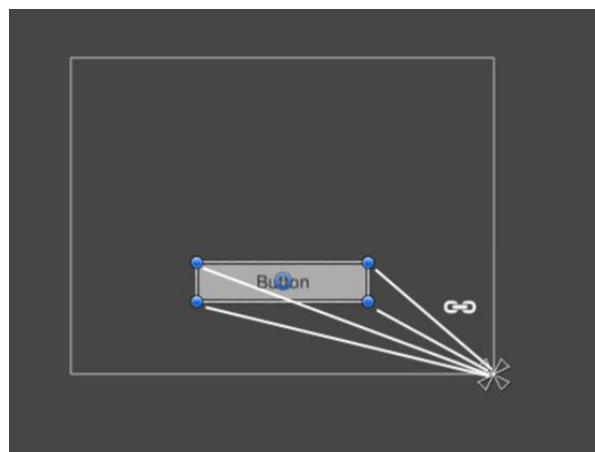


Figura. 37. Elemento UI anclado a la esquina

El anclaje también le permite al hijo estirarse junto con el ancho y la altura del padre. Cada esquina del rectángulo tiene un desfase fijo a su correspondiente ancla, i.e. la esquina superior izquierda del rectángulo tiene un desfase fijo al ancla superior izquierda, etc. De esta manera las esquinas diferentes del rectángulo pueden ser ancladas a diferentes puntos en el rectángulo padre.

Elemento UI con las esquinas izquierdas ancladas a la esquina inferior del padre y esquinas derechas ancladas a la parte inferior derecha. Las esquinas de los elementos se mantienen en un desfase fijo a sus anclas respectivas.

Las posiciones de las anclas están definidas en fracciones (o porcentajes) del ancho y la altura del rectángulo padre. 0.0 (0%) corresponde al lado izquierda o de abajo, 0.5 (50%) a la mitad, y 1.0 (100%) a la derecha o lado superior. Pero las anclas no están limitadas a los lados y la mitad; éstas pueden ser ancladas a cualquier punto dentro de un rectángulo padre.

Una característica útil des manijas de las anclas es que estas automáticamente snap a las anclas de los rectángulos hermosa para permitirle un posicionamiento preciso.

2.6.7 Presets de la Ancla

En el Inspector, el botón **Anchor Preset** puede ser encontrado en la esquina superior izquierda del componente Rect Transform. Al hacer clic en el botón trae el desplegable Anchor Presets, ver figura 38. De aquí usted puede rápidamente seleccionar de alguno de las opciones de anclajes más comunes. Usted puede anclarse al elemento Ui a los lados o la mitad del padre, o estirar juntos con el tamaño del padre. El anclaje horizontal y vertical es independiente.

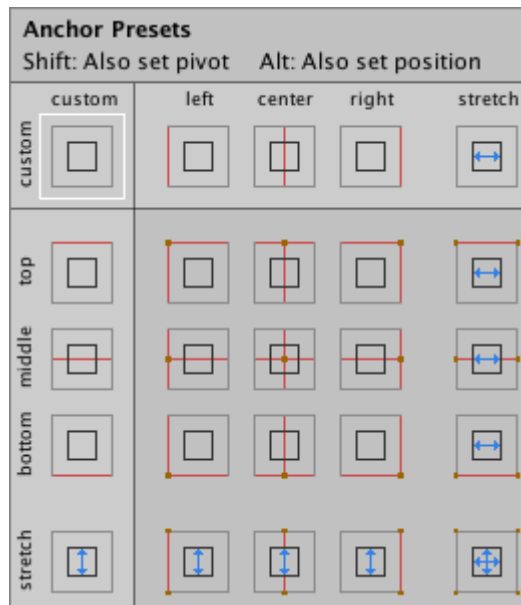


Figura. 38. Tipos de anclajes más comunes

Los botones Anchor Presets muestran la opción preset actualmente seleccionada si no hay uno ya. Si las anclas de cualquiera de los ejes horizontales o verticales están configurados a diferentes posiciones que los otros presets, la opción custom (personalizado) es mostrado).

2.6.8 Los campos de Anchor (ancla) y position (posición) en el inspector

Usted puede hacer clic en la expansión de las flechas para relevar los campos del número de anclaje si no están ya visibles. **Anchor Min** corresponde a la manija de anclaje inferior izquierda en la Vista de Escena, y **Anchor Max** corresponde a la manija superior derecha como se ve en la figura 39.

Los campos de posición del rectángulo son mostrados diferentemente dependiendo si las anclas están juntas (lo cual produce una altura y anchura fija) o separadas (cosa que causa que el rectángulo se estire junto con el rectángulo padre).

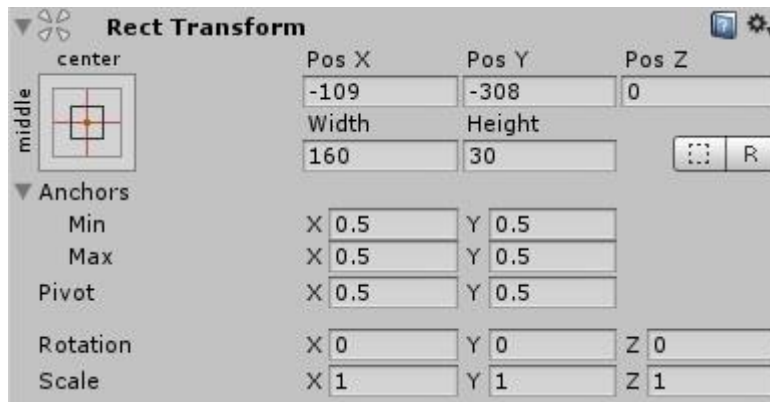


Figura. 39. Componente Rect Transform UI

Cuando todas las manijas de anclaje están juntas los campos mostrados son Pos X, Pos Y, Width (anchura) y Height (altura). Los valores de Pos X y Pos Y indican la posición del pivote relativo a las anclas.

Cuando las anclas son separadas los campos pueden cambiar parcialmente o completamente a la Izquierda, Derecha y al fondo. Estos campos definen el relleno dentro del rectángulo definido por sus anclas. Los campos Left (izquierda) y Right (derecha) son utilizados si las anclas son separadas horizontalmente y los campos Top (Superior) y Bottom (Fondo) son utilizados si son separadas verticalmente.

Tenga en cuenta que cambiando los valores en los campos de anchor o pivot va a normalmente contra-ajustar los valores de posicionamiento con el fin de hacer que el rectángulo se mantenga en lugar. En los casos que esto no es deseado, el **Raw Mode** puede ser activado utilizando un botón pequeño en el Inspector. Esto causa que el valor anchor y pivot ser capaz de cambiar sin que otro valor cambie como resultado. Esto va a causar que el rectángulo sea visualmente movido o cambie de tamaño, ya que su posición y tamaño es dependiente en los valores de anchor y pivot.

2.7 INTERACCIÓN UI Y ANIMACIONES

2.7.1 Componentes de Interacción

Esta sección cubre los componentes en el sistema UI que maneja las interacciones, tal como los eventos del mouse o táctiles y la interacción utilizando un teclado o controlador.

Los componentes de interacción no son visibles por ellos mismo, y deben ser combinadas con uno o más elementos visuales con el fin de que funcionen correctamente.

2.7.2 Funcionalidad Común

La mayoría de componentes de interacción tienen algunas cosas en común. Éstas son seleccionables lo cual significa que tienen una funcionalidad integrada compartida para visualizar transiciones entre estados (normal, highlighted, pressed, disabled), y para navegar a otros seleccionables utilizando el teclado o el controlador.

2.7.3 Button (Botón)

Un botón (figura 40), tiene un UnityEvent OnClick para definir lo que hará cuando sea oprimido. El control del Button responde a un clic del usuario y es utilizado para iniciar o confirmar una acción. Ejemplos familiares incluyen los botones Submit y Cancel utilizados en formularios web. Refiérase a la sección de scripting para ver la funcionalidad más detallada utilizada en este proyecto.



Figura. 40. Imagen de un Boton en Unity3D

Las propiedades de un botón las podemos encontrar cuando hacemos clic en el botón y en la ventana del inspector podemos observar dichas propiedades como se ve en la figura 41.

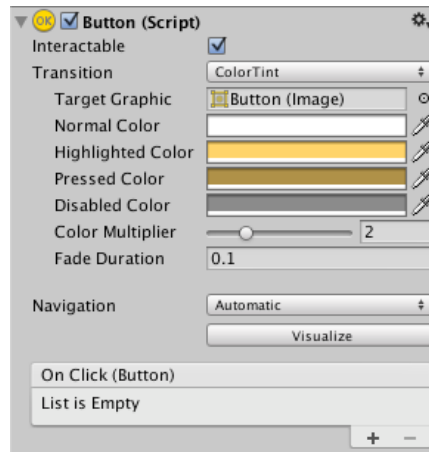


Figura. 41. Propiedades del botón en unity

La siguiente tabla (Propiedades del Boton) muestra las distintas propiedades que posee un Boton en el motor Unity y su función.

Propiedades del Botón

Propiedad:	Función:
Interactable	¿Será que este componente va a aceptar input?
Transition	Propiedades que determinan la manera que el control responde visualmente a las acciones del usuario.
Navigation	Propiedades que determinan la secuencia de controles.

A continuación, podemos observar la tabla Eventos del Botón, en la cual podemos ver la propiedad de este evento y su función.

Eventos del Botón

Propiedad:	Función:
On Click	Un UnityEvent que es invocado cuando un usuario hace clic en el botón y lo suelta.

Detalles

El botón está diseñado para iniciar una acción cuando el usuario hace clic y lo suelta. Si el mouse se mueve del control del botón antes de que el clic se haya soltado, la acción no toma lugar.

El botón tiene un solo evento llamado On Click que responde cuando el usuario completa un clic. Un caso de un uso típico incluye:

- Confirmando una decisión (eg, comenzando gameplay o guardando un juego)
- Moviéndose a un sub-menú en un GUI
- Cancelando una acción en progreso (eg, descargando una nueva escena)

2.7.4 Toggle

Un toggle tiene una casilla de verificación **Is On** que determina si el toggle está actualmente apagado véase figura 42. Este valor es cambiado cuando el usuario hace clic en el toggle, y la casilla se selecciona, y, si la casilla es seleccionada de nuevo esta se apaga. Éste también tiene un UnityEvent **OnValueChanged** para definir lo que hará cuando el valor cambie.

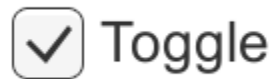


Figura. 42. Imagem de un Toggle unity 3D

El control Toggle es una casilla de verificación que le permite al usuario cambiar una opción a prendido o apagado.

Detalles

El control del Toggle le permite al usuario cambiar una opción a prendida o apagada. Usted también puede combinar varios toggles a un Toggle Group en casos dónde solo uno del conjunto debería estar prendido a la vez.

El Toggle tiene un solo evento llamado On Value Changed que responde cuando el usuario cambia el valor actual. El nuevo valor es pasado a la función de eventos como un parámetro boolean. Casos de uso típicos para Toggles incluyen:

- Cambiar una opción a prendida o apagada (eg, reproduciendo música durante un juego).
- Dejando que el usuario confirme que ha leído un aviso legal.
- Escogiendo uno de un conjunto de opciones (eg, un día de la semana) cuando se utiliza en un Toggle Group.

Las propiedades del Toggle también se visualizan en la ventana inspector cuando hemos seleccionado nuestro Ui (figura 43) .

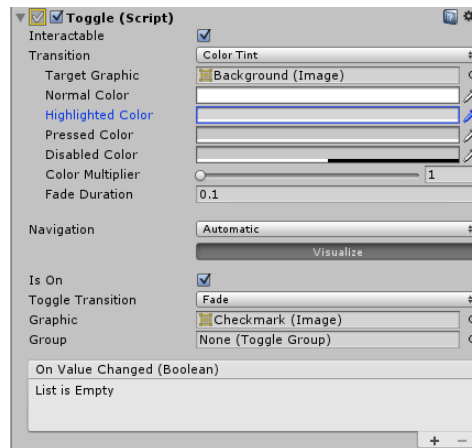


Figura. 43. Propiedades del Toggle en unity

La siguiente tabla (Propiedades del Toggle) muestra las distintas propiedades que posee un Toggle en el motor Unity y su función.

Propiedades del Toggle.

Propiedad:	Función:
Interactable	¿Será que este componente va a aceptar input?
Transition	Propiedades que determinan la manera que el control responde visualmente a las acciones del usuario.
Navigation	Propiedades que determinan la secuencia de los controles.
Is On	¿Es el toggle cambiado desde el principio?
Toggle Transition	La manera que el toggle reaccione gráficamente cuando su valor ha cambiado. Las opciones son None (ie, la marca de verificación simplemente aparece o desaparece) y Fade (ie, la marca de verificación se descansa adentro o afuera).

Propiedad:	Función:
Graphic	La imagen utilizada para la marca de verificación.
Group	El Toggle Group (si hay uno) el cual este Toggle pertenece.

A continuación, podemos observar la tabla Eventos del Toggle, en la cual podemos ver la propiedad de este evento y su función.

Eventos del Toggle.

Propiedad:	Función:
On Value Changed	Un UnityEvent que es invocado cuando el Toggle hace clic. El evento puede enviar el estado actual como un argumento dinámico tipo bool.

2.7.5 Slider (Deslizador)

Un Slider (Deslizador) tiene un número **Value** decimal el cual el usuario puede arrastrar entre un valor mínimo y máximo como se ve en la figura 44. Éste puede ser verticalmente u horizontalmente. Éste también tiene un UnityEvent **OnValueChanged** para definir lo que hará cuando el valor ha cambiado.



Figura. 44. Imagen de un Slider Unity 3D

2.7.6 Animación

La animación le permite a cada transición entre los estados del controler ser completamente animados utilizando el sistema de animación de Unity. Este es el modo más poderoso de transición debido a la cantidad de propiedades que pueden ser animadas simultáneamente.

Para utilizar el modo de transición de animación, un componente Animator necesita ser adjunto al elemento del controlador. Esto puede ser hecho de manera automáticamente al hacer clic en “Auto Generate Animation” como se muestra en la figura 45. Esto también genera un Animator Controller con estados ya configurados, los cuales necesitan ser guardados.

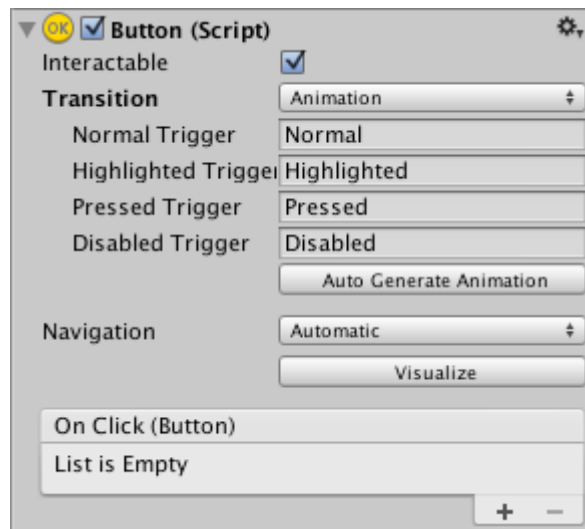


Figura. 45. Componente Animation adjunto a Botón

El nuevo Animator Controller está listo para su uso inmediatamente. A diferencia de la mayoría de los Animator Controllers, este controller también almacena las animaciones para las transiciones del controlador (véase figura 46) y estas pueden ser personalizadas, si se desea.

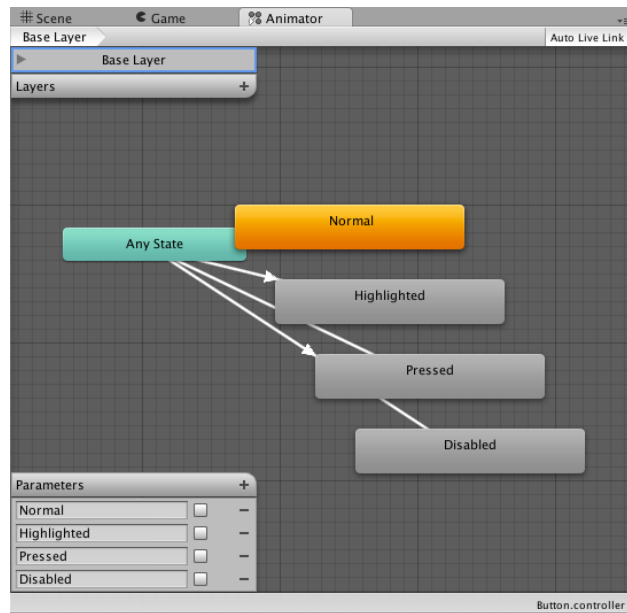


Figura. 46. Ventana de animación

Por ejemplo, si un elemento botón con un Animator Controller adjunto es seleccionado, la animación para cada uno de los estados del botón puede ser editados al abrir la ventana de Animation (Window>Animation).

Hay un menú emergente Animation Clip para seleccionar el clip deseado. Escoja de “Normal”, “Highlighted”, “Pressed” y “Disabled” (observe figura 47).

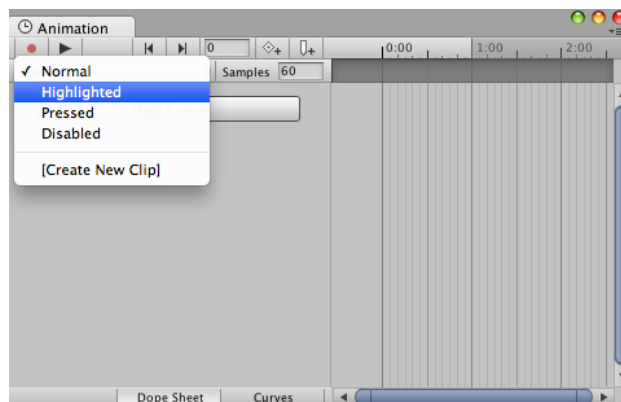


Figura. 47. Menú Animation Clip

El Estado Normal es configurado por los valores en el elemento del botón en sí y puede ser dejado vacío. En todos los otros estados, la configuración más común es

un solo keyframe al principio de la línea de tiempo. La animación de transición entre estados será manejada por el Animator.

Como un ejemplo, el ancho del botón en el estado Highlighted puede ser cambiado al seleccionar el estado Highlighted del menú emergente Animation Clip y con la cabeza a reproducción al principio de la línea de tiempo:

- Seleccione el botón de grabar
- Cambie el ancho del Botón en el inspector
- Sálgase del modo de grabación

Cambie el modo de reproducción para ver cómo el botón crece cuando es resaltado.

Cualquier número de propiedades pueden tener sus parámetros configurados en este keyframe

2.8 SCRIPTING

2.8.1 Creando y usando scripts

El comportamiento de los GameObjects es controlado por los Components que están adjuntos. Aunque los Componentes integrados de Unity pueden ser muy versátiles, usted va a ver que va a necesitar ir más allá de lo que estos pueden proporcionar para implementar las características propias de su juego. Unity le permite a usted crear su propio Componente utilizando scripts. Estos le permiten a usted provocar (activar/desactivar) eventos de su juego, modificar propiedades del Componente en el tiempo y responder a la entrada del usuario de la forma que usted quiera.

Unity admite el lenguaje de programación C # de forma nativa. C # (pronunciado C-sharp) es un lenguaje estándar de la industria similar a Java o C ++.

Además de esto, se pueden usar muchos otros lenguajes .NET con Unity si pueden compilar una DLL compatible; Aprender el arte de programar y el uso de estos lenguajes en particular, va más allá del alcance de esta introducción. Sin embargo, hay muchos libros, tutoriales y otros recursos para aprender cómo a programar con Unity.

2.8.2 Creando Scripts

A diferencia de la mayoría de los demás activos, las secuencias de comandos se crean normalmente directamente en Unity. Puede crear una nueva secuencia de comandos desde el menú Crear en la parte superior izquierda del panel Proyecto o seleccionando Activos > Crear > Secuencia de comandos C # en el menú principal. El nuevo script será creado en la carpeta que usted haya seleccionado en el Panel del Proyecto. El nuevo nombre del archivo script será seleccionado, pidiéndole a usted que ingrese otro nuevo nombre, como se observa en la figura 48.

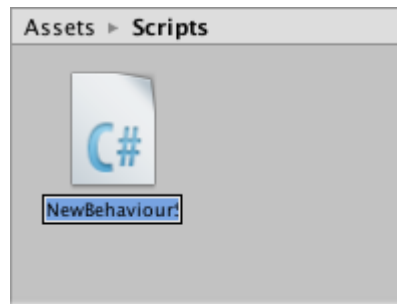


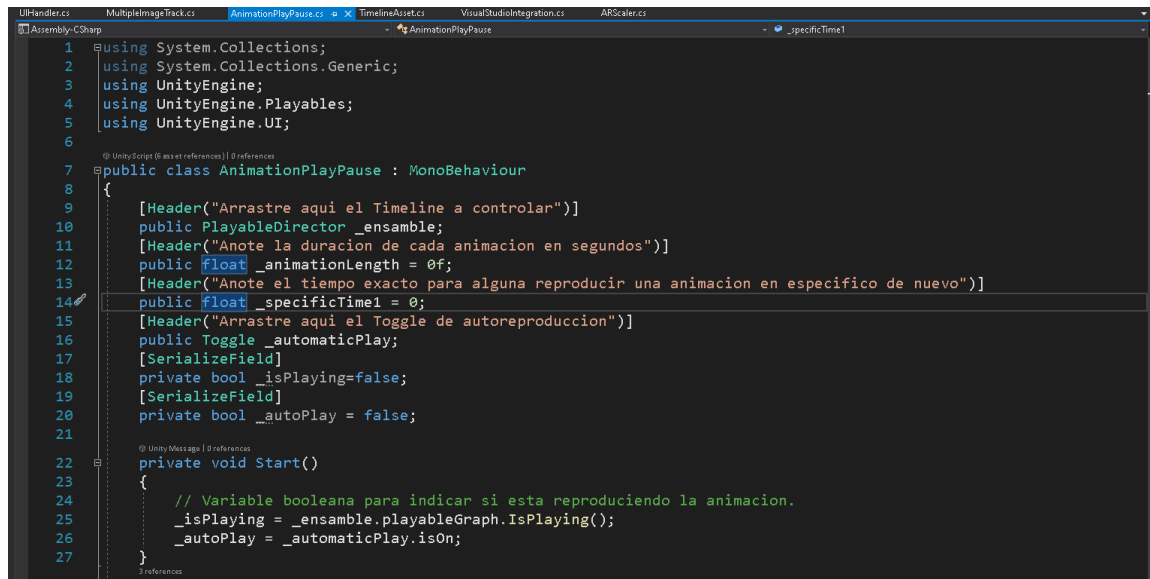
Figura. 48. Imagen de un Script al ser creado

Es una buena idea ingresar el nombre del nuevo script en este punto en vez de editarlo después. El nombre que usted ingrese será utilizado para crear el texto inicial dentro del archivo, como es descrito abajo.

2.8.3 Anatomía de un archivo Script.

Cuando hace doble clic en un activo de secuencia de comandos en Unity, se abrirá en un editor de texto. De forma predeterminada, Unity usará Visual Studio, pero puede seleccionar cualquier editor que desee del panel Herramientas externas en las preferencias de Unity (vaya a Unity > Preferencias).

Los contenidos editados del archivo script para el control de animación se verán como la figura 49:



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.Playables;
5 using UnityEngine.UI;
6
7 public class AnimationPlayPause : MonoBehaviour
8 {
9     [Header("Arrastre aqui el Timeline a controlar")]
10    public PlayableDirector _ensamble;
11    [Header("Anoté la duracion de cada animacion en segundos")]
12    public float _animationLength = 0f;
13    [Header("Anoté el tiempo exacto para alguna reproducir una animacion en especifico de nuevo")]
14    public float _specificTime1 = 0;
15    [Header("Arrastre aqui el Toggle de autoreproduccion")]
16    public Toggle _automaticPlay;
17    [SerializeField]
18    private bool _isPlaying=false;
19    [SerializeField]
20    private bool _autoPlay = false;
21
22    private void Start()
23    {
24        // Variable booleana para indicar si esta reproduciendo la animacion.
25        _isPlaying = _ensamble.playableGraph.IsPlaying();
26        _autoPlay = _automaticPlay.isOn;
27    }
28 }
```

Figura. 49. Ejemplo de script utilizado en el proyecto

Un script hace sus conexiones con el funcionamiento interno de Unity al implementar una clase que deriva desde la clase integrada llamada MonoBehaviour. Usted puede pensar una clase como un tipo de plano (blueprint) para crear un nuevo tipo de Component que puede ser adjuntado a GameObjects. Cada vez que usted adjunte un componente script a un GameObject, éste crea una nueva instancia del objeto definido por el plano. El nombre de la clase es tomado desde el nombre que usted proporciona cuando el archivo se crea. El nombre de la clase y el archivo se debió suministrar cuando el archivo fue creado. El nombre de la clase y el archivo debe ser el mismo para permitirle al componente script en ser adjuntado al GameObject.

La primera cosa para tener en cuenta son dos funciones definidas dentro de la clase. La función Update es el lugar para colocar el código que se encargará de la actualización por frame para el GameObject (en este ejemplo se omite el Update, ya que el Script reacciona a eventos OnClick de los botones). Este puede incluir movimiento, acciones de trigger y responder al input del usuario, básicamente cualquier cosa que necesite ser manejado en el tiempo durante el Tiempo de Ejecución. Para que la función Update haga su trabajo, a veces es útil configurar

variables, leer preferencias y hacer conexiones con otros GameObjects antes de que cualquier acción del juego tome lugar. La función Start va a ser llamada por Unity antes de que el gameplay comience (ie, antes de que la función Update sea llamada por la primera vez) y es un lugar ideal para hacer cualquier inicialización.

Recomendación para programadores con experiencia: Usted puede sorprenderse que la inicialización de un objeto no es hecho utilizando una función constructora. Esto se debe a que la construcción de los objetos es manejada en el editor y no toma lugar al comienzo del gameplay como usted podría suponer. Si usted intenta definir un constructor para un componente script, éste va a interferir con la operación normal de Unity y puede causar problemas mayores con el proyecto.

2.8.4 Controlando un GameObject

Como es dicho arriba, un script solo define un plano para un Component por lo que nada de su código será activado hasta que una instancia del script es adjuntada al GameObject. Usted puede adjuntar un script arrastrando el script asset a un GameObject en el panel de la jerarquía o al inspector del GameObject el cual está actualmente seleccionado. También hay un sub-menú de Scripts en el menú del Componente que va a contener todos los scripts disponibles en el proyecto, incluyendo aquellos que usted ha creado usted mismo. La instancia de script se ve mucho como cualquier otro Componente en el Inspector (ver figura 50):

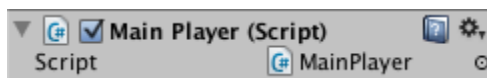


Figura. 50. Script en la ventana Inspector

Una vez adjuntado, el script va a comenzar a funcionar cuando usted presione Play y corra el juego. Usted puede revisar esto al agregar el siguiente código en la función Start como se ve en la figura 51.

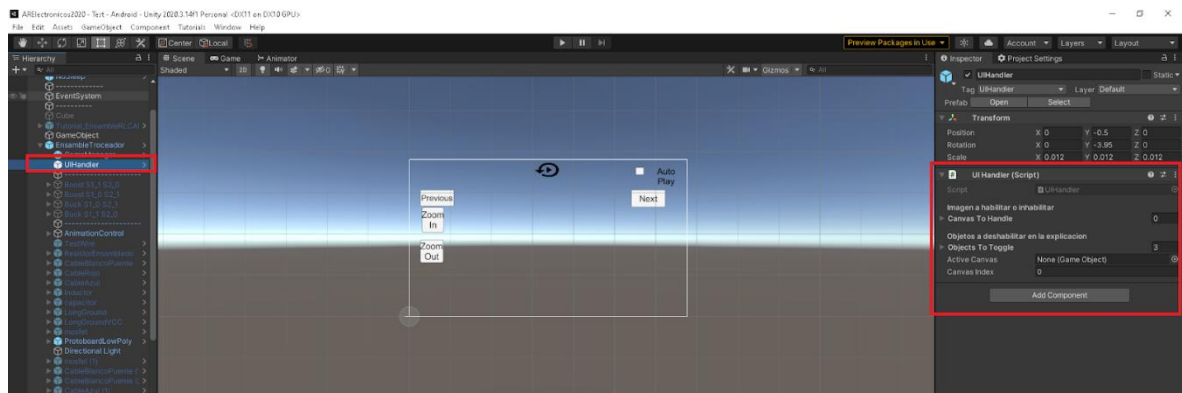


Figura. 51. Ejemplo de Script instanciado en un GameObject

Debug.Log es un comando simple que simplemente imprime un mensaje en la salida de la consola de Unity. Si presiona Reproducir ahora, debería ver el mensaje en la parte inferior de la ventana principal del editor de Unity y en la ventana Consola (menú: Ventana> General> Consola).

2.8.5 Scripts usados en este proyecto.

- AnimationPlayPause: se encarga de pausar, rebobinar y reanudar la reproducción del TimeLine principal de la animación en cada escena.
- ARScaler: Se encarga de redimensionar el tamaño de un modelo 3D instanciado en la escena de Realidad Aumentada.
- GameManager: Se encarga de las funciones principales del manejo de escenas en el proyecto.
- ImageHandler: Se encarga de hacer los cambios de imágenes a mostrar en las ayudas visuales dentro de las escenas en Realidad Aumentada.
- NoSleep: Se encarga de evitar que el celular desactive la pantalla mientras se ejecuta la aplicación en Realidad Aumentada.
- Resistor: Se usa para cambiar el código de colores presentado por cada resistencia en escena.

- ToggleButtons: Se usa para activar y desactivar botones por medio de eventos disparados por el TimeLine principal y la interacción del usuario con los botones.
- UIHandler: Este script se encarga de activar y desactivar Imágenes en pantalla, así como menús interactivos en pantalla mediante los event disparados por el TimeLine y la interacción del usuario con los botones.

2.8.6 Ejemplo de uso frecuente de los scripts en el proyecto.

El uso más frecuente que se le da a los Scripts en este proyecto es la interacción con el TimeLine en cada escena de Realidad Aumentada.

Estos se usan para activar / desactivar botones, imágenes; así como también para cambiar el estado del secuenciador del TimeLine (rebobinar, pausar, reproducir, reproducción automática, etc.)

A continuación, ilustraremos la forma más empleada de los scripts, la cual es con los “Signal Emitters” del TimeLine (observe las figuras encerradas en rectángulos rojos en la figura 52); los cuales se activan en determinados tiempos establecidos por los emisores dentro de la secuencia del TimeLine.

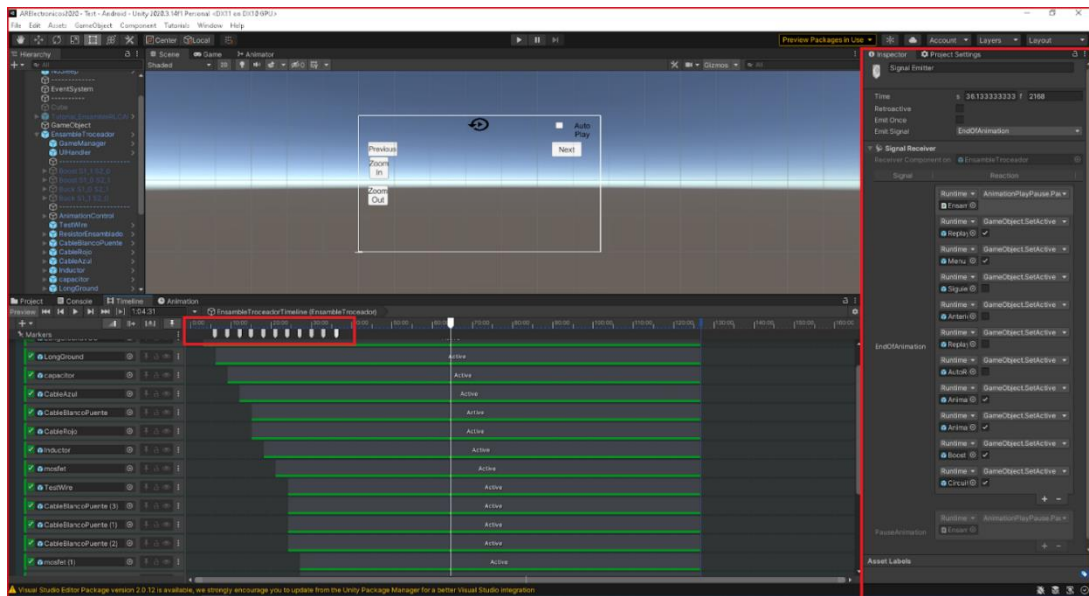


Figura. 52. Signal Emitters

El script AnimationPlayPause es el que más interactúa con la secuencia del TimeLine, este se encarga de parar la reproducción, rebobinar, etc. por medio de las interacciones del usuario con los botones presentados en pantalla.

2.9 BUILD

Esta sección se puede omitir si se instala el motor de Unity 3D desde la aplicación "Unity Hub", seleccionando el módulo correspondiente a Android y asegurándose de seleccionar los submódulos "Android SDK & NDK Tools" y "Open JDK". Lo anterior se debe a que el instalador descargara y configurara las versiones más óptimas para la versión de Unity 3D que se está instalando. Pudiendo omitir los pasos 2.9.1 y 2.9.2 de las siguientes secciones.

2.9.1 Configurando su ambiente de desarrollo Android

Necesitará tener arreglado su ambiente de desarrollo Android antes de poder probar sus juegos en el dispositivo. Esto implica descargar e instalar el SDK de Android que contenga cada una de las diferentes versiones y plataformas de Android que serán trabajadas, y agregando su dispositivo físico al sistema (esto se hace de

forma un poco diferente dependiendo de si el equipo tiene Windows o Mac). Este proceso de configuración es explicado en la página web para desarrolladores de Android, y puede que exista información adicional proporcionada por el fabricante de tu dispositivo. Dado que es un proceso complejo, hemos proporcionado un esquema básico de las tareas que deben ser completadas antes de que puedas ejecutar código en tu dispositivo Android o en el emulador de Android.

2.9.2 Agregar la ruta del SDK de Android a Unity

La primera vez que construyes un proyecto para Android (o si Unity posteriormente falla en localizar el SDK), se te pedirá ubicar el directorio en donde se halla instalado el SDK de Android (debes seleccionar el directorio raíz donde está instalado el SDK). La ubicación del SDK de Android también puede ser modificada en el editor seleccionando Unity > Preferences del menú y luego dando clic en External Tools sobre el cuadro de diálogo preferences.

2.9.3 Hacer que el dispositivo sea reconocido por el sistema

Esto podría ser complicado, en especial en sistemas Windows en donde los drivers tienden a ser un problema. También, tu dispositivo puede venir con información adicional o drivers específicos procedentes del fabricante.

Para Windows: Si el dispositivo Android es reconocido automáticamente por el sistema, es bastante probable que necesites actualizar los drives con los que vinieron con el Android SDK. Esto se hace a través del Windows Device Manager (Administrador de Dispositivos de Windows). —>Si el dispositivo no es reconocido automáticamente, usa los drivers del SDK de Android, o cualquier driver específico proporcionado por el fabricante.

Nota: No olvides habilitar la opción “USB Debugging” en tu dispositivo, en Settings -> Developer options. A partir de Android Jelly Bean 4.2 las opciones de desarrollador están escondidas por omisión. Para habilitarlas, haz tap varias veces

en Settings -> About Phone -> Build Versión . Luego serás capaz de acceder a Settings -> Developer options.

2.9.4 Construcción

Por último, damos clic en la pestaña File > Build Settings > Android, a continuación, se deben agregar las escenas que se desean construir en la ventana de “Scenes in Build” una vez seleccionadas las escenas hacemos clic en el botón “Build”, a continuación, se desplegara una ventana para seleccionar una ruta donde guardar nuestra aplicación, seleccionamos donde deseamos guardar y hacemos clic en guardar como se ve en la figura 53.

Ahora inicia el proceso de construcción, el cual puede demorar según las especificaciones de su pc.

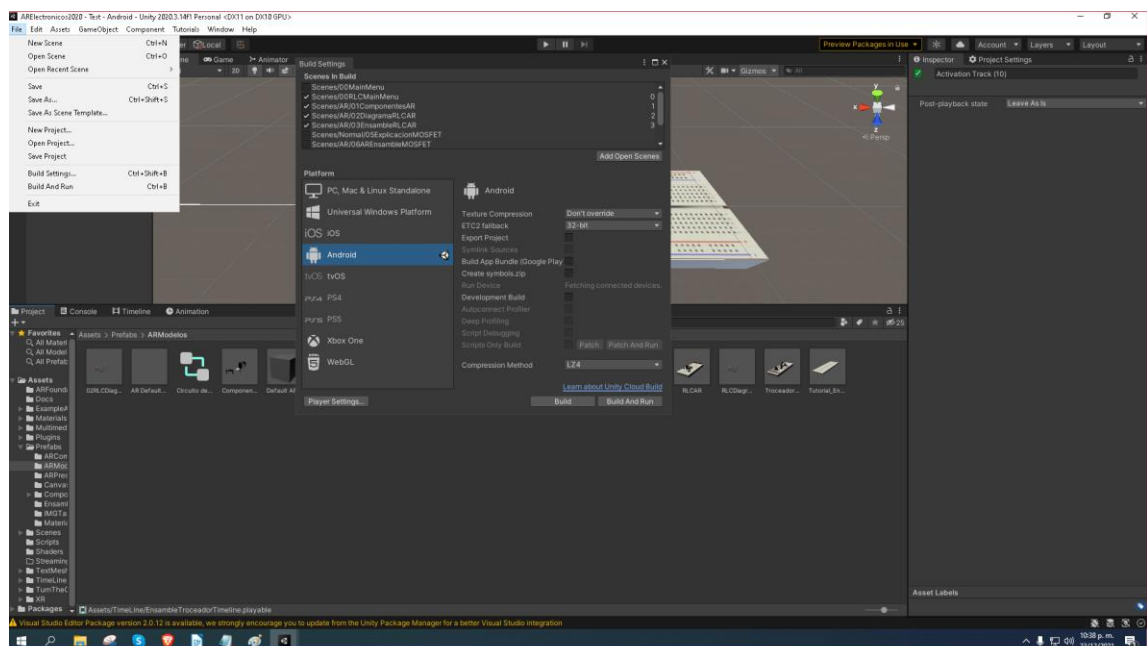


Figura. 53. Escenas agregadas a la Build de una APK

2.10 PRUEBAS

Durante todas las iteraciones del proyecto se hicieron pruebas usando dos dispositivos móviles con Android, uno de ellos un celular Huawei Y9S y el otro un Samsung Galaxy S8.

Hacer pruebas en dispositivos físicos permite detectar problemas de desempeño, así como corregir errores que se puedan suscitar en tiempo de ejecución; lo anterior deriva en un producto final de mayor calidad.

Un ejemplo claro del uso de varios dispositivos de prueba fue la corrección de los anclajes de las Interfaces de Usuario para que se adaptaran de manera dinámica a los tamaños de pantalla de varios teléfonos celulares.

2.11 RESULTADOS

El resultado obtenido del proceso descrito en el presente documento nos da una aplicación de Realidad Aumentada de fines educativos con una robustez aceptable y lo suficientemente pulida como para ser lanzada en canales oficiales de distribución de aplicaciones Android; lo anterior con la finalidad de hacer más fácil su distribución al usuario final, por ejemplo, maestros, estudiantes y cualquier persona con el interés de aprender más al respecto de la práctica con ensambles electrónicos.

CAPÍTULO 3

PROCESO METODOLÓGICO PARA EL DESARROLLO DE LA APP AR- ELECTRONICS

3.1 DESCRIPCIÓN DEL DESARROLLO Y DISEÑO DE LA APP AR- ELECTRONICS

La enseñanza generalmente incluye clases presenciales y de laboratorio. Estas sesiones incluyen clases magistrales y actividades donde el estudiante realiza ejercicios de aplicación y completa las actividades de laboratorio. Los materiales proporcionados en las actividades del laboratorio de electrónica son manuales, que incluyen desarrollos teóricos y prácticos y ejemplos de aplicación. El responsable del laboratorio proporciona el material necesario para su realización; sin embargo, el material no siempre es suficiente para todos los alumnos, por lo que herramientas de apoyo como la que se presenta en este artículo siempre son de gran ayuda. En el caso de electrónica de potencia, el alumno deberá tener conocimientos previos de circuitos eléctricos y de análisis, simulación e implementación de circuitos analógicos. Además, deben conocer álgebra lineal, derivación e integración numérica.

La taxonomía de Bloom ha sido ampliamente aceptada en la comunidad educativa indicando la profundidad de comprensión a partir del proceso de conocimiento, aplicaciones, análisis, síntesis y evaluación [34]. Para adquirir la profundidad de la comprensión, los educadores están motivados a utilizar diferentes técnicas de enseñanza. Las experiencias prácticas promueven significativamente el aprendizaje en todos los niveles de la educación científica cuando están diseñadas y guiadas adecuadamente por educadores calificados [35]. Las experiencias prácticas de laboratorio son fundamentales para el proceso de aprendizaje en todas las áreas de estudio. Las investigaciones han demostrado que los estudiantes que enfrentan

estrategias bien planificadas en experiencias de laboratorio obtienen habilidades de resolución de problemas y pensamiento crítico. Las herramientas de AR son bastante eficientes para ayudar a los estudiantes a comprender y motivarlos en el proceso de capacitación en el laboratorio [36].

El curso de electrónica de potencia implementado para estudiantes de ingeniería se resume en la Figura 54. Los temas principales son circuitos RLC, semiconductores, rectificadores, convertidores CC-CC e inversores. Se seleccionaron dos temas principales para implementar la herramienta AR: uno se abordó al inicio del curso, para motivar a los estudiantes con la nueva herramienta, el otro tema AR se abordó al final del curso. Por lo tanto, podemos tener una visión general para ver la factibilidad de diseñar un curso completo con esta herramienta.

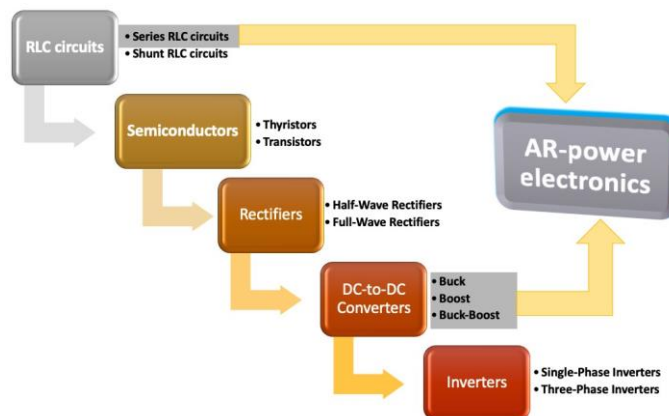


Figura. 54. Power electronic course

La metodología de diseño de AR-APP se incluye en la Figura 55. Para su diseño se requirió el uso de Blender [37] para la elaboración de los elementos 3D de los dispositivos electrónicos y la exportación de estos diseños a UNITY [38] para elaborar y coordinar las actividades en el script C#, junto con la programación en time-lapse, para construir la APK, de modo que los usuarios puedan brindar los insumos necesarios para guiarlos en las prácticas de laboratorio. Si bien Vuforia y ARkit tienen grandes capacidades para aplicaciones AR, se eligió ARCore porque mantiene mapas más grandes en términos de monitoreo del entorno de la vida real en comparación con otros SDK de realidad aumentada, como Vuforia, sin descuidar

la documentación y el soporte existente, tanto de Google y la propia comunidad. Además de lo anterior, ARCore es totalmente gratuito, y la configuración e integración de ARCore en UNITY es mucho más sencilla, por lo que se utilizó ARCore para realizar el seguimiento y reconocimiento de imágenes entre el mundo real y la realidad aumentada.

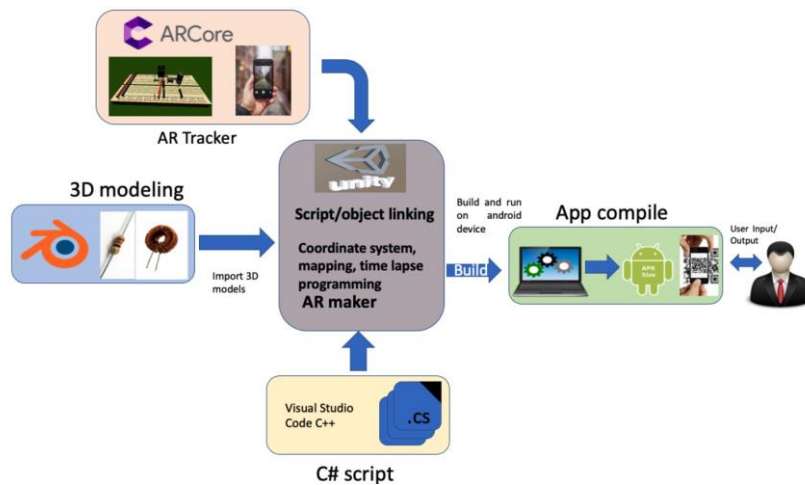


Figura. 55. Diseño de la aplicación AR—electrónica en UNITY.

3.2 DESCRIPCIÓN DE LA APP ELECTRÓNICA DISEÑADA Y DESARROLLADA

Para el diseño del proyecto se adoptó un modelo incremental, en el cual se realizó un análisis de requerimientos para el diseño de los modelos e interfaces; se realizó su codificación y pruebas hasta lograr los cambios propuestos y agregar las funcionalidades. Para la arquitectura lógica de la aplicación se consideraron los requisitos funcionales como la aplicación para dispositivos móviles, reconocimiento de códigos QR e interactividad con el usuario, así como requisitos no funcionales como la usabilidad y tiempos de respuesta aceptables. Las herramientas de desarrollo y bibliotecas utilizadas fueron ARCore [39] y UNITY. Usando diferentes API, ARCore permite que un teléfono celular detecte su entorno, comprenda el mundo e interactúe con la información programada en UNITY para crear simulaciones tridimensionales interactivas. ARCore utiliza tres capacidades clave

para integrar el contenido virtual con el mundo real visto a través de la cámara del teléfono, el seguimiento del movimiento, la comprensión del entorno y la estimación de la luz, lo que permite la detección del tamaño y la ubicación de los objetos y las condiciones de iluminación del entorno [39]. Para lograr la funcionalidad de los modelos 3D desarrollados en Blender, la creación de una escena 3D requería al menos tres componentes clave: modelos, materiales y luces, junto con la edición de objetos como mallas, curvas y superficies. Una vez desarrollado el objeto en 3D, se puede exportar en formato FBX para ser utilizado en UNITY 3D. Una vez exportados a UNITY, los objetos desarrollados deben ensamblarse usando la pestaña del inspector para proporcionar texturas, posiciones, tamaños y ensamblaje de objetos.

La figura 56 ilustra el diseño para llevar a cabo la interacción entre la aplicación y el usuario. Se crearon animaciones con líneas de tiempo usando el secuenciador UNITY 3D (Timeline) para activar objetos, botones, sliders y texto de tal manera que los usuarios controlen el seguimiento de su proceso de montaje, tanto en la APP como con los elementos reales en el Laboratorio. .

Los componentes electrónicos se hicieron a partir de formas “primitivas” incluidas en el motor UNITY 3D. En la Figura 57 se puede ver un objeto padre llamado “ARResistor”, el cual es un GameObject vacío donde se colocan como objetos hijos los elementos que componen la resistencia; por ejemplo, el GameObject que se selecciona en la jerarquía de la imagen se identifica con el nombre "ResistorBody", es decir, el cuerpo de la resistencia. El inspector que se muestra en la Figura 57 ilustra que el GameObject tiene un componente de "Transformación", que proporciona información sobre la posición, la rotación y la escala del GameObject, este componente tiene sus tres parámetros manipulados para dar la forma y la proporción y lograr la apariencia que proporciona al GameObject la forma de la resistencia. Los componentes "Mesh Filter" y "ResistorBody" se colocaron para

proporcionar las características típicas de las resistencias. Del mismo modo, el inductor, los cables de conexión y los condensadores se crearon de esta manera.

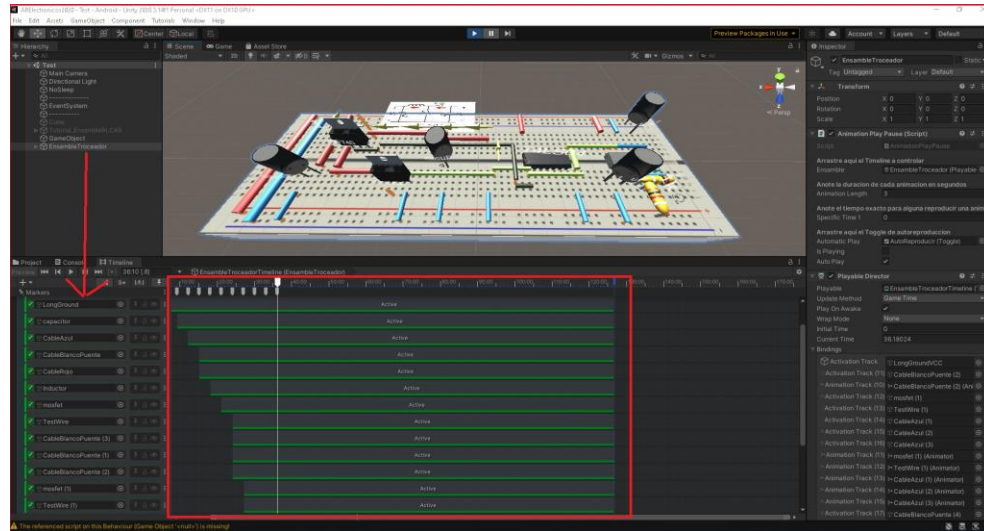


Figura. 56. Ventana de animaciones en Timeline

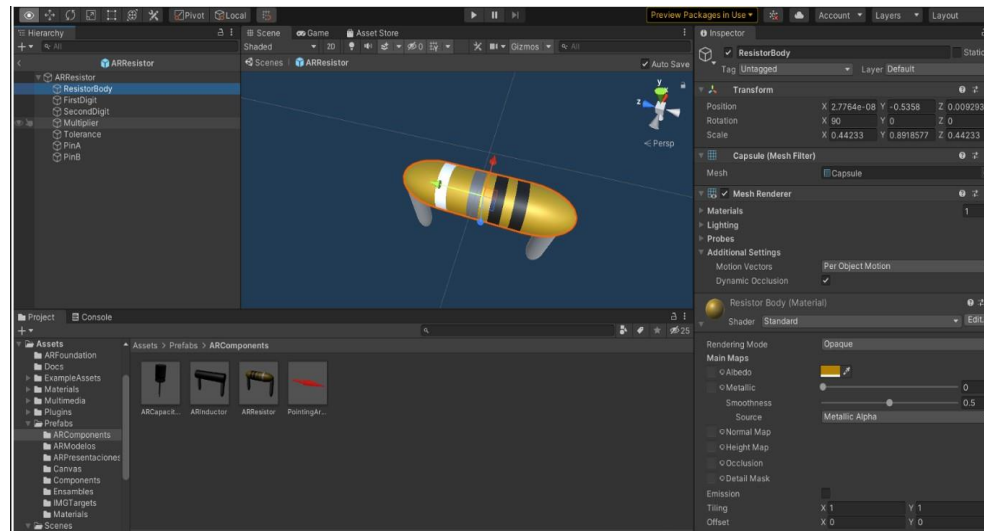


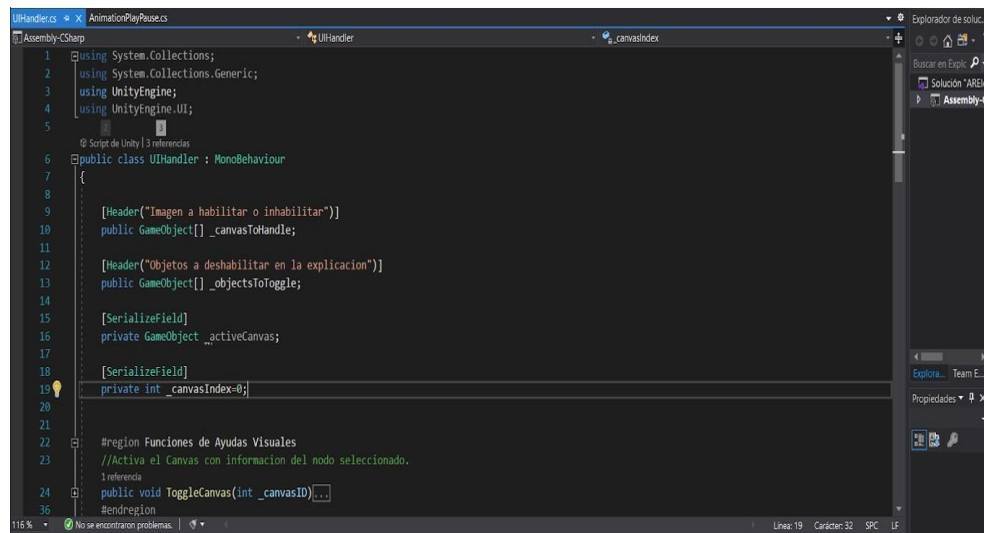
Figura. 57. AR—resistor GameObject.

Para controlar las escenas se diseñaron unos guiones. Por ejemplo, "AssembleTroceador" en este caso es el objeto "Padre" de un grupo de modelos 3D bajo su jerarquía. Contiene el script "AnimationPlayPause" que se encarga de pausar, rebobinar y reanudar la reproducción de la línea de tiempo principal de la animación en cada escena. Inmediatamente debajo de la jerarquía, conocida como

objeto hijo, se programa un script denominado “UIHandler” (Figura 58) en el cual se ubican los botones “Anterior” y “Siguiete” del lienzo con el nombre

Se agregan "AnimationControl". Este script se encarga de activar y desactivar las imágenes en pantalla, así como los eventos interactivos activados por menús en pantalla.

Una vez que tenemos los elementos con los que interactuar, es necesario realizar ajustes en la interfaz de usuario (UI) usando el menú UNITY GameObject y la herramienta Rect para mover, cambiar el tamaño y rotar los elementos de la UI de acuerdo con la interacción y dar funcionalidad a los elementos programados en UNITY. Luego se construye un APK para el tema de circuitos RLC configurado para realizar el montaje en laboratorio y el análisis matemático del circuito. De manera similar, para el convertidor Buck-Boost bidireccional, se construye otro APK para analizar individualmente los convertidores Buck y Boost y que funcione como un convertidor bidireccional.



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class UIHandler : MonoBehaviour
7 {
8     [Header("Imagen a habilitar o inhabilitar")]
9     public GameObject[] _canvasToHandle;
10
11     [Header("Objetos a deshabilitar en la explicacion")]
12     public GameObject[] _objectsToToggle;
13
14     [SerializeField]
15     private GameObject _activeCanvas;
16
17     [SerializeField]
18     private int _canvasIndex=0;
19
20
21
22 #region Funciones de Ayudas Visuales
23 //Activa el Canvas con informacion del nodo seleccionado.
24 public void ToggleCanvas(int _canvasID)
25 #endregion
26
```

Figura. 58. Script UIHandler para activar/desactivar la ayuda visual o los cambios de escena.

3.3 DESARROLLO DEL AR PARA CIRCUITOS RLC

Uno de los principales problemas al enseñar el análisis matemático de circuitos RLC es que algunos estudiantes no pueden obtener sus ecuaciones de espacio de estado. Por lo tanto, estas ecuaciones se obtuvieron de forma interactiva diseñándolas en AR, guiando al usuario a comprender su modelado. El circuito RLC implementado en este estudio considera una conexión en serie, como se muestra en la Figura 59, y las ecuaciones en el espacio de estado están dadas por la Ecuación (1).

$$\begin{bmatrix} \frac{dV_C(t)}{dt} \\ \frac{dI_L(t)}{dt} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} V_C(t) \\ I_L(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} \begin{bmatrix} 0 \\ v(t) \end{bmatrix} \quad (1)$$

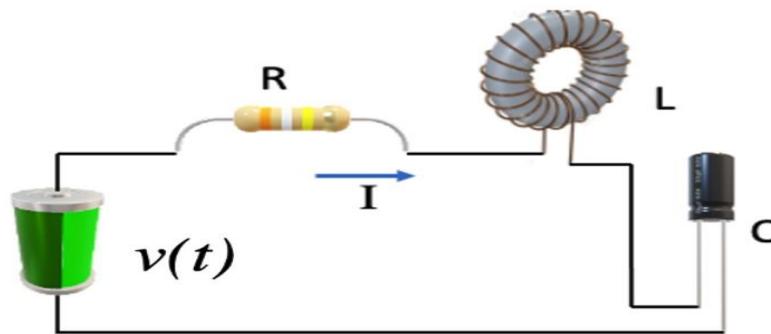


Figura. 59. Circuito en serie RLC

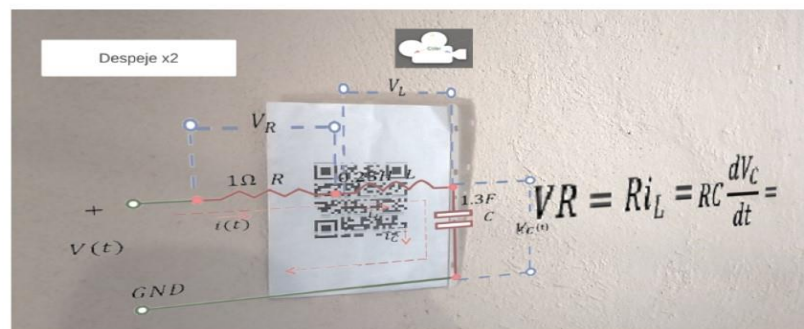


Figura. 60. AR - Análisis matemático interactivo del circuito RLC

El diseño del circuito AR para RLC se dividió en dos etapas: una explicación matemática interactiva para obtener su modelo matemático en espacios de estado

y representar el circuito RLC en dos ecuaciones diferenciales (Figura 60); el montaje del circuito RLC en un protoboard electrónico, mostrando paso a paso los dispositivos montados en el protoboard. Adicionalmente, se agregó el comportamiento de las señales de voltaje y corriente, visualizadas en un osciloscopio (Figura 61), lo que le muestra al estudiante cómo colocar las puntas del osciloscopio dentro del circuito RLC para obtener las respuestas de las formas de onda. La figura 61 ilustra la circulación de energía a través de cada uno de los dispositivos con flechas, así como sus valores de magnitud.

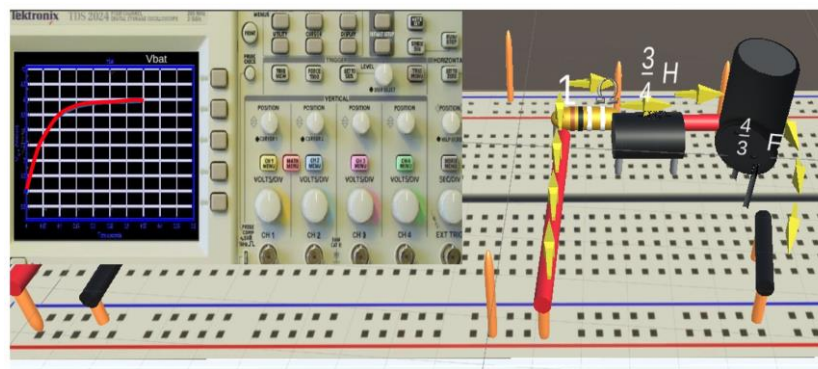


Figura. 61. GUI del circuito AR-RLC para ensamblar dispositivos en una protoboard electrónica.

3.4 DESARROLLO DE AR PARA EL CONVERTIDOR DC-DC BUCK-BOOST

La Figura 62 muestra el diagrama eléctrico del convertidor Buck-Boost bidireccional. Este convertidor tiene un inductor, un capacitor, dos interruptores de potencia y dos fuentes de voltaje; esto garantiza un flujo de energía bidireccional, lo que lo hace adecuado en aplicaciones en vehículos eléctricos y energías renovables. Por lo tanto, esto hace que los estudiantes muestren un interés particular en este tipo de convertidores. El convertidor Buck-Boost bidireccional tiene dos fuentes de voltaje en los extremos, V_1 y V_2 ; los interruptores S_1 y S_2 funcionan de forma

complementaria; por lo tanto, tiene cuatro modos de operación según el nivel de voltaje de las fuentes y el estado de los interruptores (Figura 63).

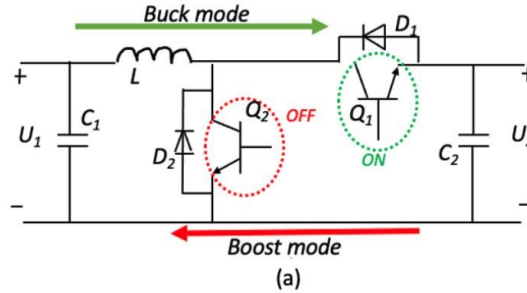


Figura. 62. Diagrama eléctrico del convertidor Buck-Boost bidireccional

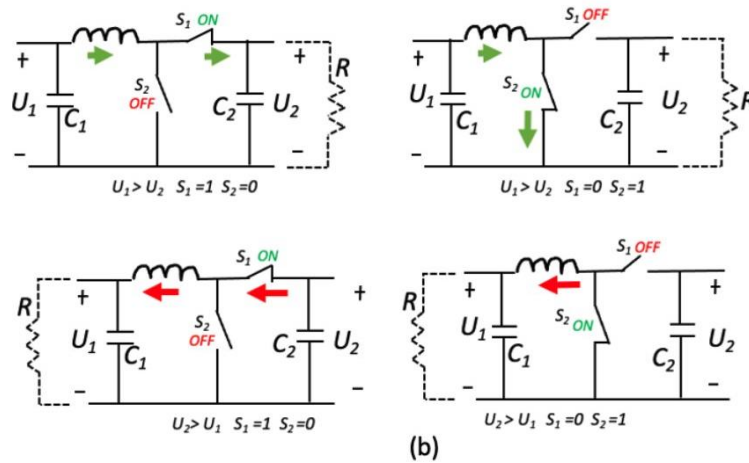
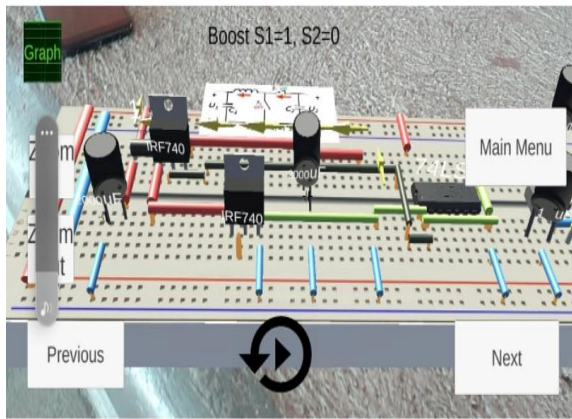
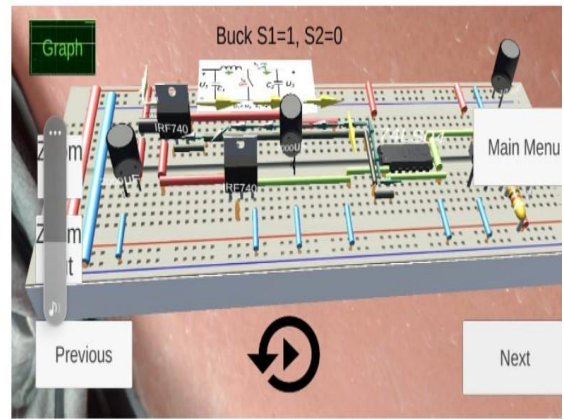


Figura. 63. Topología de convertidor DC-DC bidireccional.

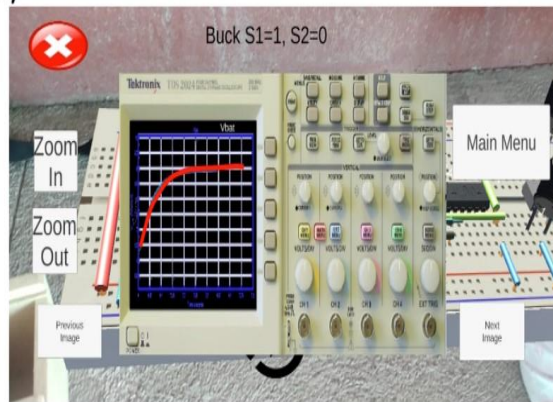
El diseño del convertidor bidireccional en AR se divide en dos etapas, una para el convertidor Buck (Figura 64a) y otra para el convertidor Boost (Figura 64b). El flujo de energía que circula de una batería a otra se muestra con animaciones para aclarar su funcionamiento. Además, las formas de onda del voltaje a través de los capacitores 1 y 2 se muestran en la Figura 64c.



(a)



(b)



(c)

Figura. 64. AR GUI para ensamblar el convertidor Buck/Boost. (a) Guía de conexión para Buck; (b) Ayuda de montaje para Boost; (c) Usar un osciloscopio para medir el voltaje de las baterías.

CAPÍTULO 4.

PROCESO EXPERIMENTAL

4.1 PARTICIPANTES

La Tabla 1 muestra los participantes que utilizaron la herramienta creada. Si bien esta herramienta fue diseñada para estudiantes de ingeniería, también fue utilizada por estudiantes del Centro de Enseñanza Automotriz CD GUZMAN para fomentar desde temprana edad el interés y la vocación por los temas de ingeniería. Para los quince participantes (de 15 a 19 años), esta era la primera vez que realizaban actividades de ingeniería y no tenían ningún conocimiento sobre electrónica, por lo que el montaje del circuito RLC se realizó físicamente y con el apoyo de la aplicación

Tabla 1. Participantes usando la aplicación AR-electronics.

Actividad	Estudiantes vocacionales	Estudiantes universitarios	Trabajadores de la industria
Experimentación física RLC	X	X	X
Análisis Matemático RLC		X	X
Experimentación Física Buck-Boost		X	X

Participaron 17 universitarios de segundo año entre 20 y 25 años del Instituto Tecnológico de Ciudad Guzmán, para quienes su primera exposición a la electrónica fue circuitos eléctricos y electrónica básica, así como cursos de matemáticas. Por lo tanto, estos participantes utilizaron la aplicación AR para apoyar

el montaje del circuito RLC, el análisis de las ecuaciones en el espacio de estado y el montaje del convertidor bidireccional Buck-Boost durante los cursos implementados. La Figura 65 muestra algunos estudiantes utilizando la aplicación de ARelectronics, a partir de lo cual se podrían establecer una serie de beneficios, como se ilustra en la Tabla 2, donde se mencionan las características, servicios y beneficios de la aplicación desarrollada.

Una de las principales actividades de un investigador es mantenerse en contacto con la industria y poder realizar investigaciones conjuntas u ofrecer servicios, incluidos cursos de formación para su personal. Por lo tanto, se implementó un curso de electrónica de potencia con la ayuda de AR, el cual se impartió a 6 trabajadores entre 30 y 50 años en una empresa [42]. Su especialidad era en áreas afines como ingenieros de planta, técnicos, supervisores y capacitación. Cabe mencionar que [42] es una empresa dedicada al diseño electrónico con aplicaciones en la industria automotriz, industrial, médica y tecnológica con colaboración en más de 30 países; por lo tanto, la herramienta AR utilizada en los cursos de electrónica de potencia podría tener un impacto internacional.



Figura. 65. Usando la aplicación AR-electronics.

Tabla 2. Lista de características y servicios que ofrece la aplicación AR-electronics.

N	Hallazgos del uso de la aplicación AR-Electronics
1	Las estudiantes muestran motivación en la clase.
2	Estudiantes con participación activa y aprendizaje a su propio ritmo
3	Reducir el tiempo para realizar actividades en el laboratorio.
4	Desarrollo de la independencia para realizar tareas.
5	Mejor comprensión para el trabajo en equipo y el aprendizaje colaborativo
6	La aplicación brinda asistencia con la operación en el sitio durante las actividades en el laboratorio.
7	Apoyar el análisis matemático de los circuitos RLC
8	Los estudiantes aprenden más rápido y más fácilmente.

4.2 PERCEPCIONES Y EXPERIENCIAS DE LAS TECNOLOGÍAS

El objetivo de este estudio fue examinar el efecto de la herramienta AR en términos de mejora del conocimiento y carga cognitiva de los estudiantes. La evaluación general de la aplicación de AR en cursos de electrónica se presenta en la Tabla 3 para examinar la percepción de los formadores sobre los aspectos generales, siete preguntas enfocadas en el uso de la aplicación AR, desde su instalación, y de la parte interactiva. Se les animó a responder si estaban totalmente de acuerdo, de acuerdo, en desacuerdo o totalmente en desacuerdo. La Figura 66 ilustra que la mayoría de los estudiantes estuvo totalmente de acuerdo en que AR puede ayudarlos a tener una mejor comprensión de los temas; solo tres de los encuestados no estuvieron de acuerdo, especialmente en la parte de la aplicación en la industria,

y en sus comentarios mencionaron que esta herramienta podría distraerlos de sus actividades laborales.

Tabla 3. Evaluación general para el uso de la aplicación AR-electronics.

N	Pregunta para evaluación general
Q1	¿Fue fácil la instalación de la aplicación?
Q2	¿Fue la aplicación útil e interactiva?
Q3	¿La aplicación AR me ayudó a mantener mi atención durante el curso?
Q4	¿Me gustó estudiar este curso?
Q5	¿Crees que la realidad aumentada te ayudó en tu formación?
Q6	¿Es la realidad aumentada más eficiente para aprender que una página web?
Q7	¿Estoy dispuesto a recomendar esta herramienta de aprendizaje AR/3D a mis amigos?

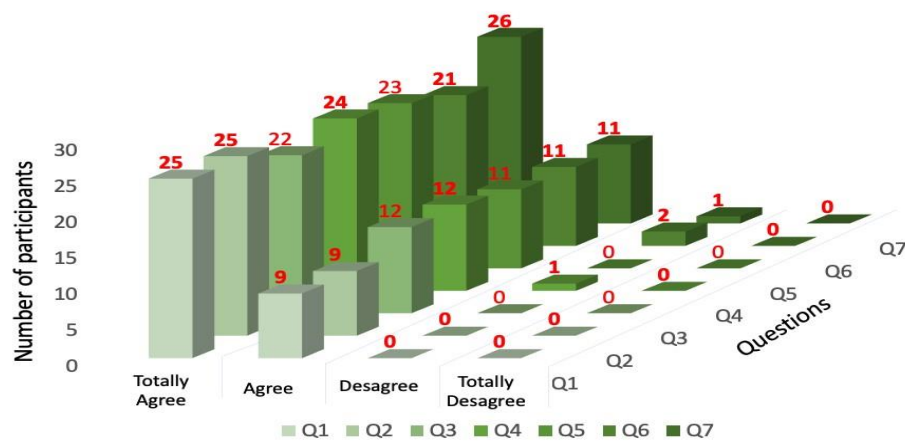


Figura. 66. Evaluación general de la aplicación AR-electronics.

La Tabla 4 muestra cuatro preguntas sobre el uso de los circuitos AR-RLC; las preguntas se enfocaron en la parte de interactividad matemática para entender las ecuaciones del espacio de estados y la parte del circuito de ensamble, así como las mediciones realizadas sobre la resistencia, inductancia y capacitancia para medir el voltaje y la corriente, como lo harían con un osciloscopio. En esta encuesta, se les pidió que calificaran la aplicación AR del 1 al 5, donde la calificación más alta fue 5. La Figura 67 muestra los resultados de la encuesta, donde la aceptación se puede observar principalmente en la parte de comentarios. La percepción tanto de la parte matemática como del montaje de los elementos fue buena. Algunos con conocimientos de electrónica afirmaron que la aplicación AR puede ser un poco tediosa, lo que sugiere una interacción más rápida en el montaje en el protoboard. Sin embargo, esto podría ser contraproducente para los principiantes.

Tabla 4. Encuesta sobre el uso de los circuitos AR-RLC.

N	Preguntas sobre el uso de los circuitos AR-RLC
Q1	¿Es claro el análisis matemático del circuito RLC en AR y me ayudó a entenderlo?
Q2	¿Es una presentación de matemáticas en AR mejor que una presentación de Power point?
Q3	¿El uso de la aplicación AR-3D RLC me ayudó a comprender mejor el ensamblaje realista de los circuitos RLC?
Q4	¿La APP AR-RLC facilita la comprensión teórico-práctica?

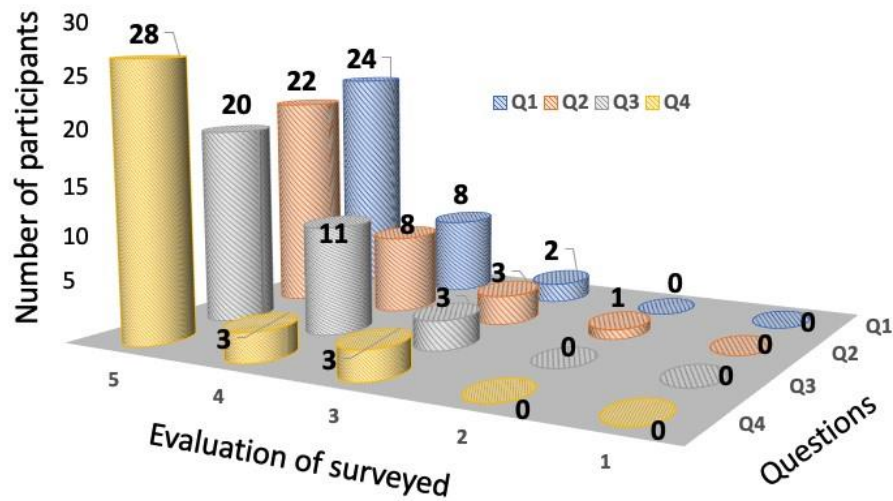


Figura. 67. Encuesta de circuitos AR-RLC.

En el caso de los convertidores DC-DC bidireccionales, la Tabla 5 muestra cuatro preguntas sobre la mejora en la comprensión al usar la aplicación AR. Indicaron menos confusión sobre el comportamiento del convertidor Buck o Boost, así como una mejor apreciación cuando el ensamblaje AR se muestra en una placa electrónica. De manera similar, se les pidió que calificaran la aplicación AR del 1 al 5, donde la calificación más alta fue 5 (Figura 68).

Tabla 5. Encuesta para el uso de los convertidores AR DC-DC.

N	Preguntas sobre los convertidores CC-CC
Q1	¿Entiendo mejor cómo fluye la energía en un convertidor Buck-Boost bidireccional?
Q2	¿Entiendo mejor la interacción entre los interruptores?
Q3	Con animaciones AR, ¿entiendo cuando funcionan como Buck o Boost?

Q4 ¿El montaje AR-3D del circuito Buck-Boost en la protoboard me ayuda a hacerlo más fácilmente físicamente?

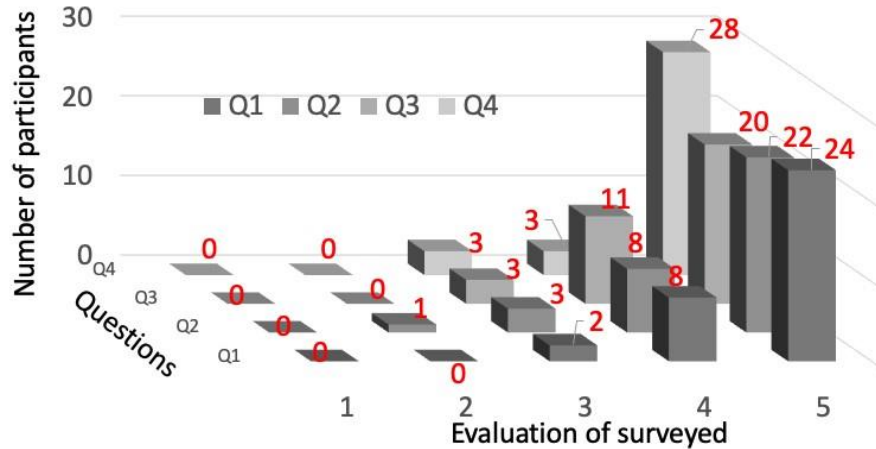


Figura. 68. Encuesta AR-DC-DC

4.3 REALIDAD MIXTA APLICADA AL CONTROL NO LINEAL DE LA AUTONOMÍA DEL TRACTOR

Para realizar la integración de un entorno de realidad virtual, se debe tener la integración de oculus para Unity o el SDK oculus en Unity, a través de esta API y en conjunto con el hardware y software que nos provee oculus podemos llevar al usuario a experimentar de lo que trata la realidad virtual. Una vez que hemos logrado tener en Unity esta integración se debe preparar el entorno virtual que será visible para el usuario, el modelado 3D con Blender será el principal protagonista pues nos permite exportar estos modelos en formato .fbx el cual es compatible con Unity, organizar de la manera correcta estos modelos para proporcionar una experiencia más agradable en términos de visualización, agregar las texturas adecuadas para que estos objetos tomen vida en el entorno de realidad virtual.

Una vez que tenemos el entorno y al usuario o jugador dentro de él, debemos recurrir al compilador visual studio un ide que nos ayuda a integrar a Unity nuestro

desarrollo mediante scripts de C#, estos scripts son otro tipo de componente, los agregamos al objeto a través del Inspector. Por dentro, estos componentes solo son trozos de código que crearemos para que hagan tareas varias o cambien el comportamiento del objeto según lo deseemos o de acuerdo a las necesidades de nuestro proyecto.

El siguiente paso es crear una interfaz gráfica que otorgue al usuario parámetros de visualización básicos, la integración de UIs es una opción que tenemos disponible en Unity y es el modo en que el jugador puede interactuar con nuestro videojuego, como, por ejemplo, el menú de inicio o los botones para poder mover los personajes o todo tipo de indicadores que necesitamos estén presentes en nuestro juego.

Enseguida, tenemos que preparar nuestro entorno de manera que nos permita logearnos como se ve en la figura 69, para poder enviar nuestros datos en tiempo real a nuestra consola en Firebase, crear una UI que nos permite hacer este login y desarrollar los scripts que permitan esta función. Existen numerosos ejemplos en repositorios que nos facilitan toda esta tarea, y solo sería suficiente con adecuarlos a nuestras necesidades.

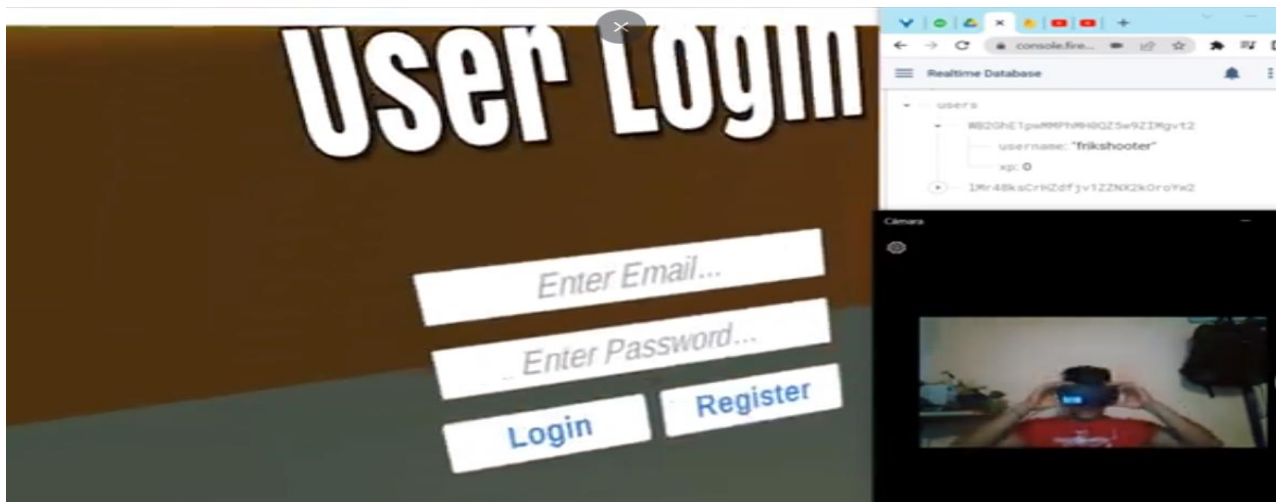


Figura. 69. Interfaz para login en Firebase

El primer paso es asegurarse que de tener SDK de Firebase Unity más reciente lo podemos descargar del siguiente enlace:

<https://firebase.google.com/docs/unity/setup?hl=es-419>

Después hay que cumplir con los siguientes requisitos:

Instala Unity 2018.4 o una versión más reciente. Las versiones anteriores pueden ser compatibles, pero no de forma activa. La compatibilidad con Unity 2018.4 se considera obsoleta y ya no se admitirá de forma activa después de la próxima actualización importante.

Para continuar, nuestro proyecto de Unity debe tener estos requisitos véase figura 70:

- iOS: Debe segmentarse a iOS 10 o una versión más reciente.
- Android: Debe segmentarse al nivel de API 19 (KitKat) o una versión más reciente.

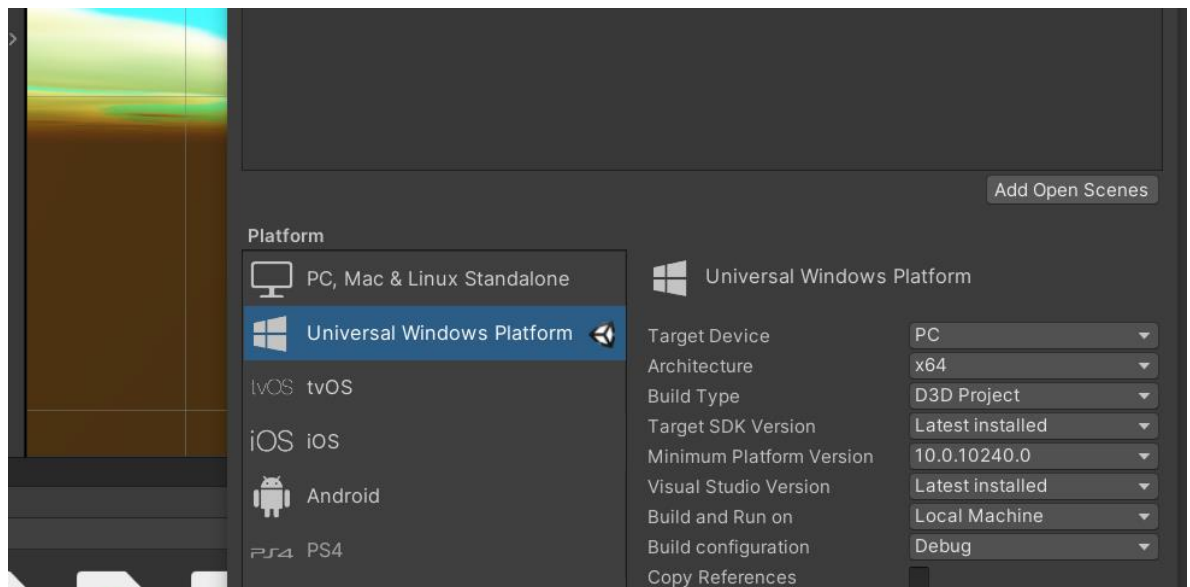


Figura. 70. Módulos requeridos en Unity - Firebase

Enseguida hay que agregar un proyecto en Firebase como se muestra en la figura 71, se debe crear un nuevo proyecto y asignarle un nombre. Una vez que revisamos y aceptamos las condiciones de Firebase, después, hacemos clic en Crear proyecto.

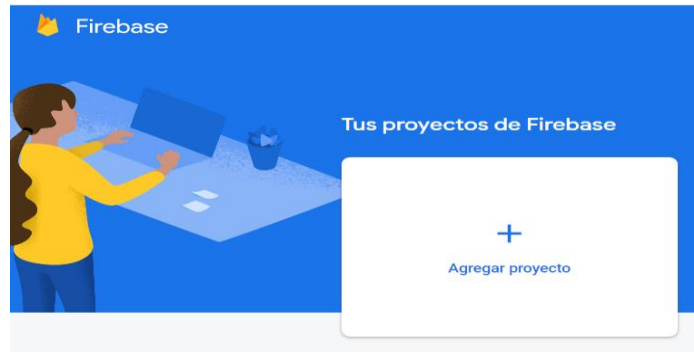


Figura. 71. Vista para agregar proyecto en Firebase

Es importante como siguiente paso agregar registrar la app como se muestra en la figura 72 (proyecto de Unity) con Firebase, de nuevo hacemos clic en Firebase console, en el centro de la página de descripción general del proyecto, haz clic en el ícono de Unity para iniciar el flujo de trabajo de configuración.

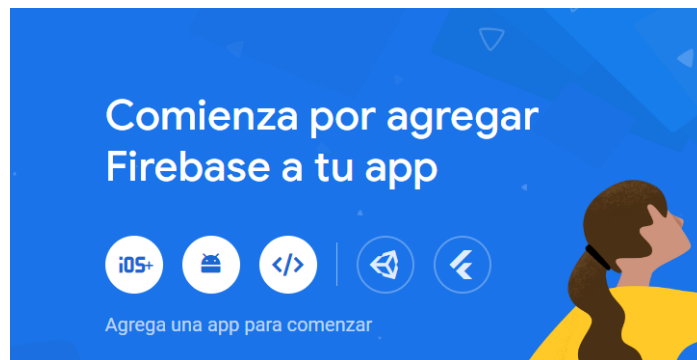


Figura. 72. Ventana para agregar APP en Firebase

Después fue necesario agregar el ID del proyecto de Unity, lo encontramos en Unity > File > Build and settings > Player settings > Product name, aquí encontramos el ID de nuestro proyecto de Unity y hacemos clic en Registrar APP (ver figura 73).

The image shows a screenshot of the 'Register app' form in the Firebase console. The form is divided into two sections: one for Apple and one for Android. The Apple section includes a text input for 'ID del paquete de Apple' with the value 'com.company.appname' and another for 'Sobrenombre de la app (opcional)' with the value 'Mi app para Apple'. The Android section has a checked checkbox for 'Registrar como app para Android', a text input for 'Nombre del paquete de Android' with the value 'com.wedwwd', and another for 'Sobrenombre de la app (opcional)' with the value 'Mi app para Android'. A blue 'Registrar app' button is located below the form. At the bottom, there is a vertical progress indicator with four steps: 1. 'Registrar app' (completed), 2. 'Descargar archivo de configuración', 3. 'Agregar el SDK de Firebase', and 4. 'Próximos pasos'.

Figura. 73. Registro de APP para Android

Usando el flujo de trabajo de configuración de Firebase console obtenemos los archivos de configuración de Firebase específicos de las plataformas.

Por ejemplo, para nuestro caso, Android: Haz clic en `Download google-services.json.`, abrimos la ventana `Project` del proyecto de Unity y transferimos los archivos de configuración a la carpeta `Assets` como se muestra en la figura 74, volvemos a Firebase console y, en el flujo de trabajo de configuración, damos clic en `Siguiente`.

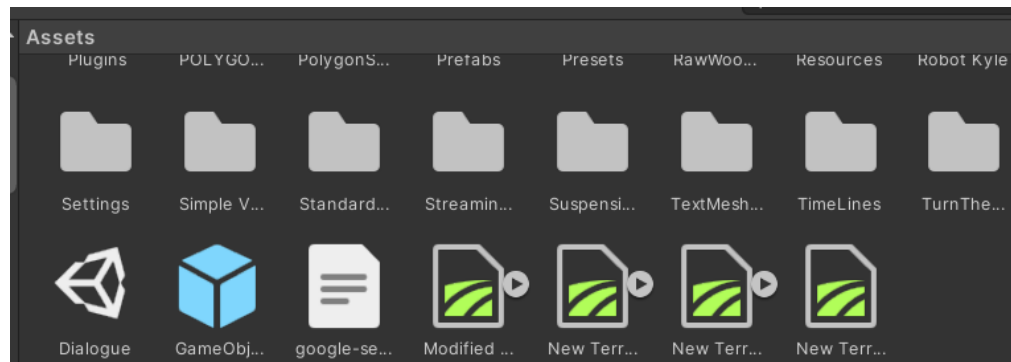


Figura. 74. Vista de la carpeta Assets en Unity

Agregar los SDK de Firebase Unity

Para los usuarios nuevos del SDK de Unity, se recomienda el siguiente flujo de trabajo de configuración en el SDK de Unity.

En Firebase console, haz clic en Descargar el SDK de Firebase Unity y descomprímelo en el lugar que prefieras.

- Puedes volver a descargar el SDK de Firebase Unity cuando quieras.
- El SDK de Firebase Unity no es específico para cada plataforma.

Abre tu proyecto de Unity, ve a Elementos > Importar paquete > Paquete personalizado.

- En el SDK descomprimido, selecciona los productos de Firebase admitidos que deseas usar en la app.
- En la ventana Import Unity Package, haz clic en Import.
- Vuelve a Firebase console y, en el flujo de trabajo de configuración, haz clic en Siguiente.
- Confirma los requisitos de la versión de los Servicios de Google Play
- El SDK de Firebase Unity en Android requiere los Servicios de Google Play actualizados para poder usarlo.

Se debe agregar el siguiente código al inicio de la aplicación (ver figura 75). Puedes buscar los Servicios de Google Play y, de forma opcional, actualizarlos a la versión que requiera el SDK de Firebase Unity. De esta manera, es posible llamar a otros métodos en el SDK.

```

Firebase.FirebaseApp.CheckAndFixDependenciesAsync().ContinueWith(task => {
    var dependencyStatus = task.Result;
    if (dependencyStatus == Firebase.DependencyStatus.Available) {
        // Create and hold a reference to your FirebaseApp,
        // where app is a Firebase.FirebaseApp property of your application class.
        app = Firebase.FirebaseApp.DefaultInstance;

        // Set a flag here to indicate whether Firebase is ready to use by your app.
    } else {
        UnityEngine.Debug.LogError(System.String.Format(
            "Could not resolve all Firebase dependencies: {0}", dependencyStatus));
        // Firebase Unity SDK is not safe to use here.
    }
});

```

Figura. 75. Sección de código encontrado en Firebase

Estos son los productos de Firebase admitidos, para el caso de nuestro proyecto seleccionamos Realtime Database en la documentación de referencia hay más información sobre las bibliotecas de Firebase Unity.

El SDK de Firebase Unity admite los siguientes productos de Firebase en iOS y Android, se muestran la tabla de la figura 76:

Producto de Firebase	Paquete de Unity
AdMob	Se distribuye por separado en el complemento de AdMob para Unity
Analytics	FirebaseAnalytics.unitypackage
Authentication	FirebaseAuth.unitypackage
Cloud Firestore	Firestore.unitypackage
Cloud Functions	FirebaseFunctions.unitypackage
Cloud Messaging	FirebaseMessaging.unitypackage (recomendado) FirebaseAnalytics.unitypackage
Cloud Storage	FirebaseStorage.unitypackage
Crashlytics	FirebaseCrashlytics.unitypackage (recomendado) FirebaseAnalytics.unitypackage
Dynamic Links	FirebaseDynamicLinks.unitypackage (recomendado) FirebaseAnalytics.unitypackage
Realtime Database	FirebaseDatabase.unitypackage
Remote Config	FirebaseRemoteConfig.unitypackage (recomendado) FirebaseAnalytics.unitypackage

Figura. 76. Tabla de productos Firebase - Unity

Funcionalidad de Realidad Mixta

El siguiente script de C# llamado `FirestoreManager` (figura 77.), encontraremos todos los elementos de la UI, entrada de textos, avisos, botones y lo necesario para realizar el login como se ve en la figura 78.

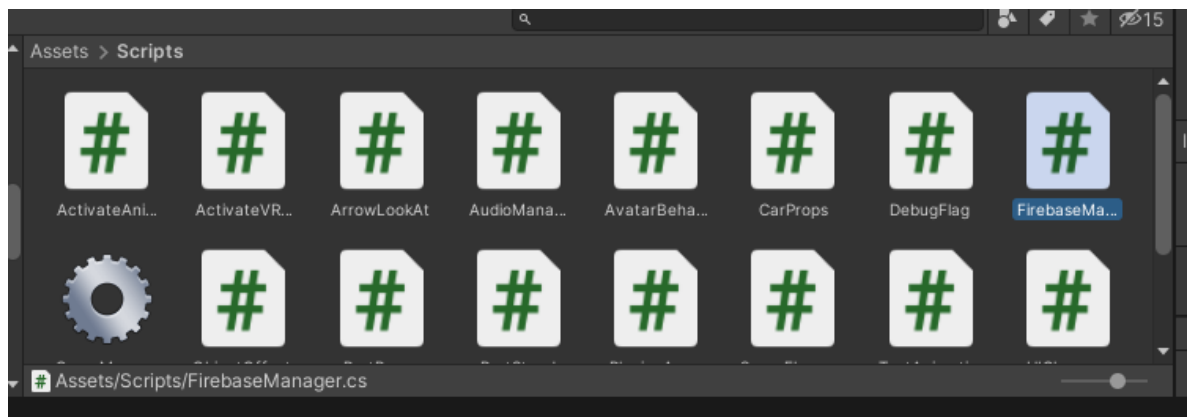


Figura. 77. Ubicación del script FirestoreManager en Assets>Scripts

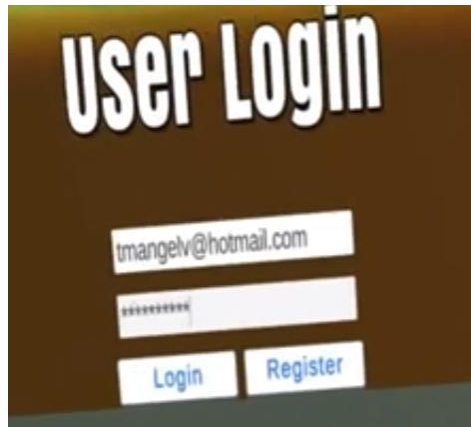


Figura. 78. UI para realizar login Unity - Firebase

Firebase Variables, esta sección de script es donde colocamos la url de la base de datos que se encuentra en el archivo JSON como se muestra en la figura 78, es una variable publica con la opción de modificarse en el editor, el resto son tipos de datos pertenecientes a las librerías del Google Firebase.

```
//Firebase variables  
[Header("Firebase")]  
public string _uri = "";  
public DependancyStatus dependancyStatus
```

Figura. 79. Sección para introducir url de la base de datos

Login Variables, esta sección nos muestra los objetos de la UI del tipo Text Mesh Pro que son requeridos para el inicio de sesión e interacción con el usuario como se observa en la figura 80. Aquí en el editor arrastramos el objeto o UI que será mostrado en el inicio de sesión y registro de variables

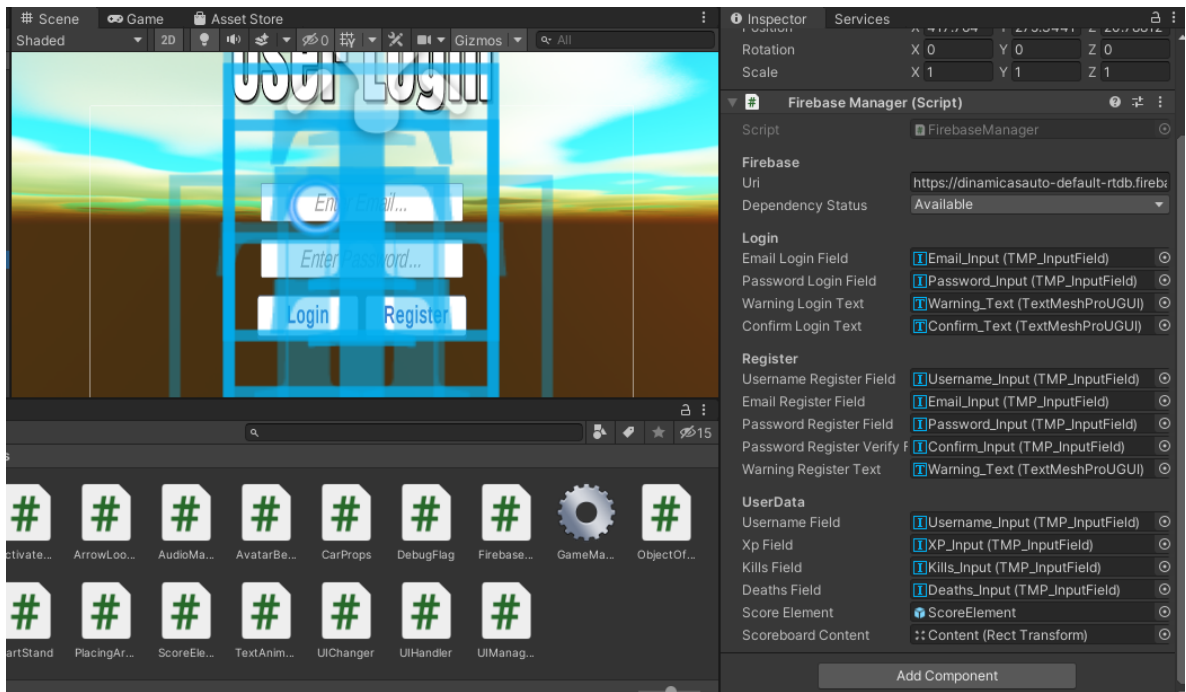


Figura. 80. Ejemplo de UI Text Mesh Pro

Void Awake, en este método revisa que todas las dependencias necesarias estén presentes en el sistema de Firebase, es decir, la aplicación

Private void initialize firebase, en este método, si todo está correcto se inicializa Firebase

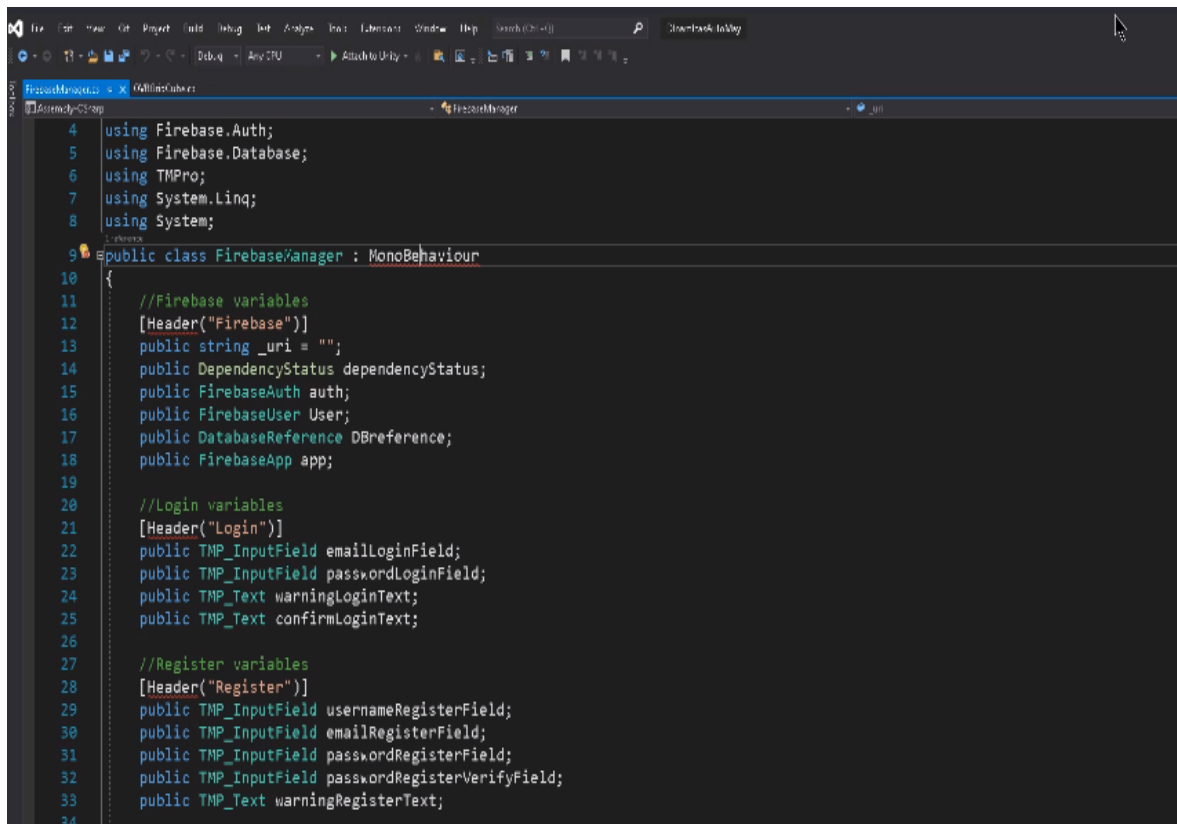
Llama a la rutina de inicio de sesión pasando el correo electrónico y la contraseña

Función para el botón de registro `public void RegisterButton()` Llama a la rutina de registro pasando el correo electrónico, la contraseña y el nombre de usuario

Función para el botón de cerrar sesión `public void SignOutButton()`

Función para el botón de guardar `public void SaveDataButton()`

Estos métodos llaman a la corrutina que da una tarea a cada función de los diferentes botones y toman los datos del texto en la interfaz, ver script de la figura 81.



```
4 using Firebase.Auth;
5 using Firebase.Database;
6 using TMPPro;
7 using System.Linq;
8 using System;
9
10 public class FirebaseManager : MonoBehaviour
11 {
12     //Firebase variables
13     [Header("Firebase")]
14     public string _uri = "";
15     public DependencyStatus dependencyStatus;
16     public FirebaseAuth auth;
17     public FirebaseUser User;
18     public DatabaseReference DBreference;
19     public FirebaseApp app;
20
21     //Login variables
22     [Header("Login")]
23     public TMP_InputField emailLoginField;
24     public TMP_InputField passwordLoginField;
25     public TMP_Text warningLoginText;
26     public TMP_Text confirmLoginText;
27
28     //Register variables
29     [Header("Register")]
30     public TMP_InputField usernameRegisterField;
31     public TMP_InputField emailRegisterField;
32     public TMP_InputField passwordRegisterField;
33     public TMP_InputField passwordRegisterVerifyField;
34     public TMP_Text warningRegisterText;
```

Figura. 81. Sección de script FirebaseManager

El script SteeringWheel que se muestra en la figura 82, implementa la interfaz SimpleInputDraggable lo que nos permite realizar el giro del volante con las manos virtuales, el mouse o cualquier dispositivo HID a su vez se actualiza el eje horizontal de la entrada de datos del usuario. También lleva a cabo el efecto de retracción del volante una vez que se suelta este, se reestablece la rotación a la posición inicial de cero, realiza la función de rotar la imagen del volante, este script acota el ángulo hasta el cual deseamos que gire el volante y la velocidad de giro del mismo.

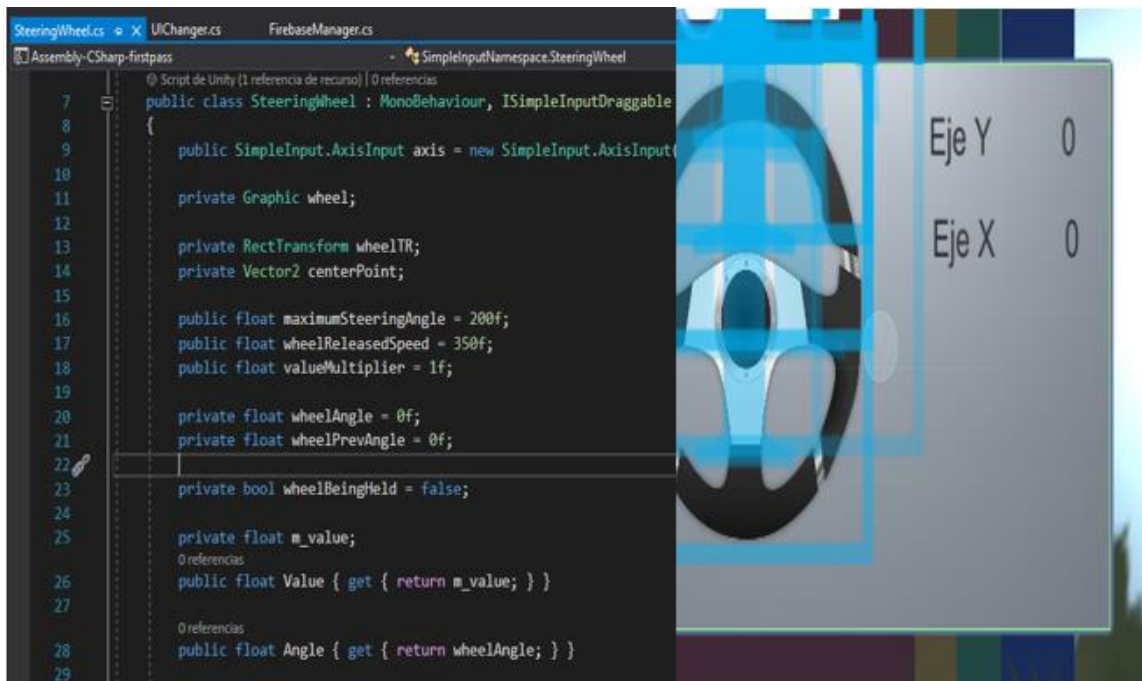


Figura. 82. Sección de script SteeringWheel

El Script de C# UIChanger que se muestra en la figura 83, es el que se encarga de mostrar los datos de actualización de la posición del volante en las UI del tipo texto que indiquemos, además, toma el valor mostrado de estos textos y lo multiplica por 100 para obtener un rango de valores del -100 hasta el 100, dejando así de lado el remanejo de variables del tipo float en los posteriores programas, este dato lo guarda en la variable SaveSteeringWheel luego de buscar el objeto con la etiqueta "FiraBaseManager", solicitar el componente FirebaseManager es decir el script de Firebase, el cual contiene la variable dicha anteriormente, que es enviada a la consola de Realtime Database. Este proceso o método es llamado una vez cada frame, esto significa que la velocidad de actualización dependerá del hardware en cuestión, entre más fps mayor será la latencia de actualización de nuestra variable

```

6 public class UIChanger : MonoBehaviour
7 {
8     // Textos en la UI para representar Rotacion en Y y X.
9     public Text _xText;
10    public Text _yText;
11
12
13    private float inputHorizontal;
14    private float inputVertical;
15
16    private string horizontalAxis = "Horizontal";
17    private string verticalAxis = "Vertical";
18
19    // Update is called once per frame
20    void Update()
21    {
22        inputHorizontal = SimpleInput.GetAxis(horizontalAxis);
23        inputVertical = SimpleInput.GetAxis(verticalAxis);
24
25        _xText.text = inputHorizontal.ToString();
26        _yText.text = inputVertical.ToString();
27        GameObject.FindGameObjectWithTag("FirebaseManager").
28            GetComponent<FirebaseManager>().
29            SaveSteeringWheel((int)(inputHorizontal*100));
30    }
31 }

```

Figura. 83. Script UIChanger

`public void SaveSteeringWheel(int _steering)` este método que se observa en la figura 84, se ejecuta al girar el volante una vez que Firebase a autentificado nuestros datos de usuario

```

1 referencia
public void SaveSteeringWheel(int _steering)
{
    StartCoroutine(UpdateUsernameAuth(usernameField.text));
    StartCoroutine(UpdateUsernameDatabase(usernameField.text));

    StartCoroutine(UpdateXp(_steering));
}

```

Figura. 84. Método publico SaveSteeringWheel

En la corrutina `UpdateXp(int _xp)` esta ingresa el dato a la base de datos actual y con nuestro user id y dentro de este actualiza el valor entero del campo xp.

Después el script de C# `UIChanger`, aquí encontramos los objetos públicos del tipo texto donde se muestran los valores actuales tipo float, enseguida, en el void `Update`, se toman los valores de la entrada horizontal y vertical del giro o posición

del volante, para este caso se utiliza solamente el eje x acotándolo con un valor float de -1 a 1, después actualiza el valor de x e Y en la interfaz del volante para ser visible al usuario. Después busca al objeto con la etiqueta “Firebasemanager” y accede al script `FirebaseManager` (ver figura 85) para llamar al método `SaveSteeringWheel` y le pasa el valor adquirido

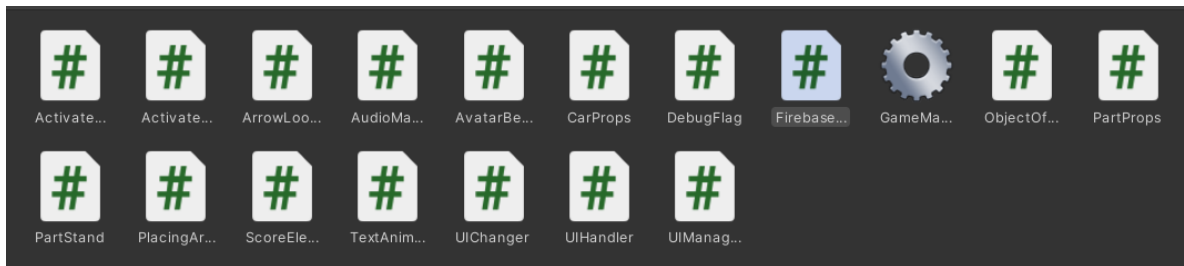


Figura. 85. Scripts

A grandes rasgos, esta es parte de la configuración e integración de Unity – Firebase en realtime database que nos permite enviar un dato del tipo entero en tiempo real a una base de datos donde podemos utilizarlo para distintos fines y aplicaciones, para este caso, se trata de un ejemplo práctico de la posición de un volante partiendo desde la posición 0, donde el dato que conseguimos enviar va desde el -100 con el volante completamente a la izquierda hasta 100 con el volante completamente a la derecha, se envía el dato el cual es recibido por nuestra base de datos preparada en Firebase.

En esta integración el usuario es desplegado en medio de un garaje donde puede moverse libremente dentro de los límites pre definidos, el usuario podrá encontrarse con una pantalla o UI en la cual deberá colocar un correo electrónico y contraseña que le dará acceso a la base de datos realtime database y le permitirá enviar los valores que se han establecido en rango de valores predefinidos. Para poder enviar este dato, el usuario, luego de loggearse, se encontrará con una pantalla (UI) que muestra el volante con los valores (figura 86) que este puede modificar y enviar, el usuario puede señalar con su mano el volante y sosteniendo el gatillo del controlador puede manipular el volante hacia la izquierda o a la derecha tanto como el margen de los parámetros lo permita.

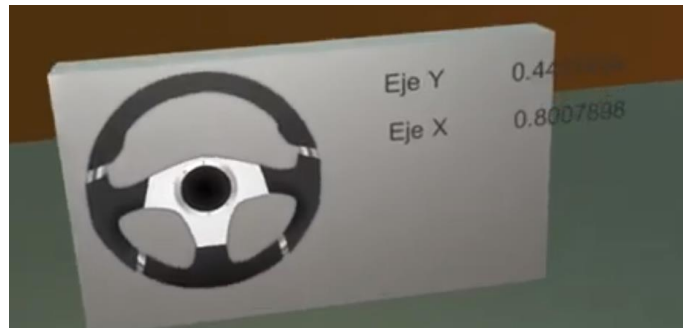


Figura. 86. UI con volante y valores

El volante es dotado de un efecto de retracción que se asemeja mucho a la reacción que se tiene con un volante de un automóvil en la vida real, al mismo tiempo que estamos manipulando el volante el valor está siendo enviado a Firebase en el apartado realtime, donde podemos comparar que efectivamente el dato se envía casi en tiempo real como se muestra en la figura 87, este dato puede ser tomado de Firebase por un controlador que a su vez manipule un volante real o de la indicación de giro a las ruedas de un automóvil, logrando de esta forma la realidad mixta como se puede ver en las figuras 88 y 89 .

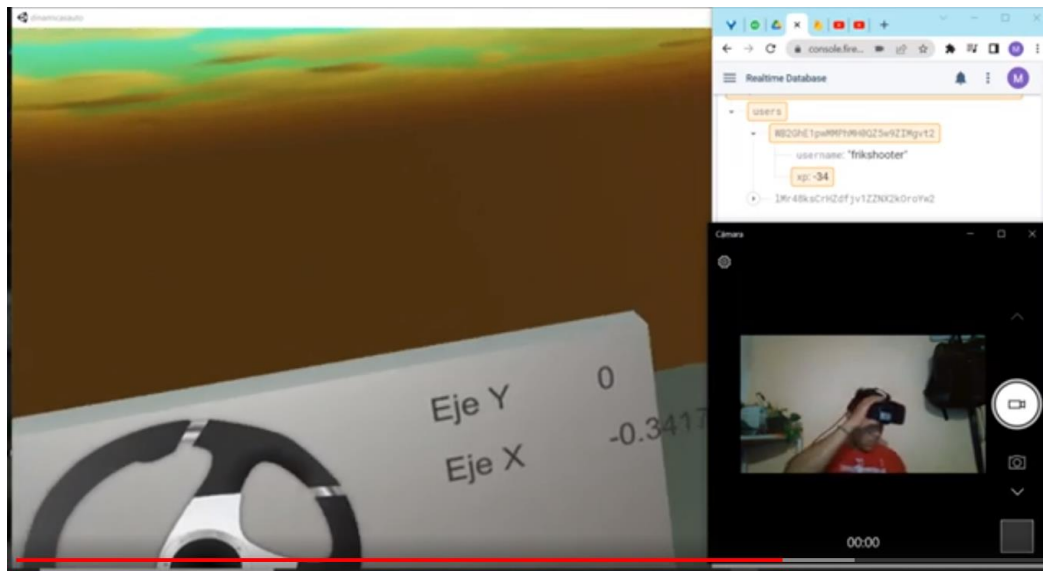


Figura. 87. Ejemplo de envío de datos Unity - Firebase



Figura. 88. Pruebas de plataforma de tractor y Realidad Mixta



Figura. 89. Envío de datos a través de Firebase a plataforma de tractor autónoma

CAPÍTULO 5.

RESULTADOS

5.1 RESULTADOS Y DISCUSIONES

Durante el curso, los estudiantes universitarios se dividieron en dos subgrupos de 13 estudiantes: el primer grupo se denominó Grupo-Control (GC) siguiendo la enseñanza tradicional, mientras que el Grupo-Experimental (EG-AR) utilizó la aplicación AR en las asignaturas de circuitos RLC y convertidores DC-DC. En estas dos materias se aplicó una prueba previa, cuya evaluación promedio se muestra en la Figura 90 y la Tabla 6. Al finalizar el curso, se implementó una actividad de laboratorio para reforzar sus conocimientos y evaluar el uso de la aplicación AR. Los resultados de la evaluación post-test se ilustran en la figura 88. La Tabla 6 ilustra el análisis estadístico del método ANOMA aplicado durante el pre-test y el test final para establecer diferencias notables entre los conocimientos obtenidos de ambos grupos. El método ANOMA indicó que no hubo diferencia significativa durante el pretest ($p = 0,532 > 0,05$), indicando que se habían adquirido conocimientos muy similares entre los grupos GC y EG-AR. En el post-test, ANOMA proporcionó un valor ($p = 0,006 < 0,05$), indicando una diferencia en el aprendizaje alcanzado.

Tabla 6. Alumnos participantes.

	Pre - Test			Post - Test		
	Promedio	F	<i>p</i>	Promedio	F	<i>p</i>
TT grupo	6.77	5.1263	0.033	7.35	8.839	0.0066
Grupo de aplicaciones AR- Electronics	6.9			8.7		

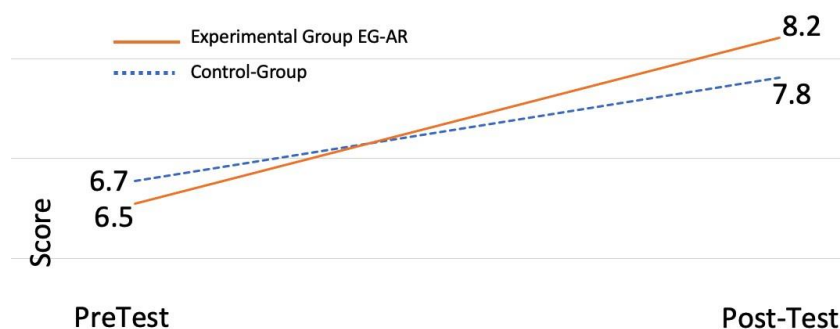


Figura. 90. Medición de la mejora del conocimiento.

En la tabla 7 se muestra la rúbrica de evaluación utilizada para evaluar a los estudiantes en el tema de circuitos RLC, dando mayor importancia al montaje físico en el laboratorio y al análisis matemático. La tabla 8 describe la rúbrica de evaluación utilizada para evaluar el conocimiento de los estudiantes sobre el tema de los convertidores CC-CC, y se le dio mayor importancia al montaje físico de Buck, Boost, Buck-Boost y el control PWM.

Tabla 7. Rúbrica de evaluación para el análisis de RLC.

N	Instrucciones para el aprendizaje	Parámetro de evaluación
1	Identificar las diferencias de una resistencia, una bobina y un condensador.	5%
2	Explicar el comportamiento de un circuito RLC conectado en serie.	10%
3	Usa el protoboard correctamente	10%
4	Montaje físico del circuito RLC en el laboratorio	35%
5	Análisis matemático	20%
6	Mediciones	10%
7	Conclusiones que demuestran el conocimiento de los circuitos RLC	10%

Table 8. Rúbrica de evaluación para el análisis Buck-Boost.

N	Instrucciones para el aprendizaje	Parámetro de evaluación
---	-----------------------------------	-------------------------

1	Explicar el comportamiento del modo Buck	5%
2	Explicar el comportamiento del modo Boost	5%
3	Montaje físico del circuito Buck en el laboratorio	25%
4	Montaje físico del circuito Boost en el laboratorio	25%
5	Montaje físico del controlador PWM	20%
6	Mediciones	10%
7	Conclusiones sobre el convertidor Buck-Boost bidireccional	10%

Los resultados presentados nos permitieron darnos cuenta que la motivación de los estudiantes mejoró con la herramienta desarrollada; manifestaron interés en seguir aprendiendo con este tipo de herramienta moderna, sugiriendo la factibilidad de extender la herramienta a todos los temas de un curso de electrónica de potencia. El aumento de las capacidades mostradas por los estudiantes que utilizaron la herramienta permite prever que los profesionales que se gradúen serán altamente calificados, tal como lo requiere el país y la sociedad. Por tanto, esto da, a los autores, el incentivo para seguir mejorando y ampliando la herramienta.

5.2 CONTRIBUCIONES

De este trabajo se publicó el artículo “ON THE USE OF AUGMENTED REALITY TO REINFORCE THE LEARNING OF POWER ELECTRONICS FOR BEGINNERS “ en la revista MDPI electronics, con la colaboración de sus respectivos autores, del cual se emitió el certificado de aceptación que se observa en la figura 91.

Publicación de Artículo “ON THE USE OF AUGMENTED REALITY TO REINFORCE THE LEARNING OF POWER ELECTRONICS FOR BEGINNERS “ en revista MDPI Electronics, con cuartil Q2, con factor de impacto JCR (Journal Citation Reports) 2.69.



Figura. 91. Certificado de aceptación de artículo.

Se desarrollaron herramientas TICs como lo fueron las aplicaciones de nombres: Circuito RLC y Convertidores Buck – Boost (figura 92) que ayudan a las materias estrechamente relacionadas a la electrónica de potencia para los distintos niveles de estudio.

Las herramientas desarrolladas se implementan en cursos impartidos a la industria, así como en diferentes niveles de estudio.



Figura. 92. Aplicación RLC y Buck - Boost,

5.3 CONCLUSIONES

En este estudio, se utilizó tecnología AR para enseñar electrónica de potencia. Se desarrolló un entorno de realidad aumentada para circuitos RLC y convertidores Buck-Boost para explorar los impactos del uso de esta tecnología. Los resultados experimentales indicaron un efecto positivo en los estudiantes en comparación con el enfoque de enseñanza tradicional; por lo tanto, la posibilidad de diseñar todo el curso con herramientas AR en el futuro es apropiada. El estudio tuvo como objetivo medir las habilidades obtenidas y la motivación de los usuarios. Por tanto, los resultados indicaron un mejor rendimiento cognitivo con un 40% más de retención por las actividades realizadas, por lo que es factible seguir diseñando herramientas para todo el curso de electrónica. Además, la exploración realizada con la herramienta AR para capacitar a los trabajadores de la industria indicó su aceptación. Se dieron comentarios alentadores para la solución de problemas de estas herramientas.

Agradecimientos: Los autores agradecen profundamente a Flextronics Manufacturing Mex SA de CV.

ANEXOS

ANEXO A. REGISTRO DE CAMBIOS

Fecha	Cambios detallados
11 de Enero del 2021	Instruir a crear repositorio en Git
11 de Enero del 2021	Crear UI (y logica agregar peso)
11 de Enero del 2021	Revision de los cambios realizados
12 de Enero del 2021	Analizar el gameplay ya creado
12 de Enero del 2021	Agregar funcionalidad al gameplay (interaccion UI)
14 de Enero del 2021	Agregar animaciones a la UI al agregar peso
19 de Enero del 2021	Revision de los cambios realizados
20 y 21 de Enero del 2021	UI senalar el punto a colocar pieza tomada
26 y 27 de Enero del 2021	AVATAR dar indicaciones en cada evento
28 y 29 de Enero del 2021	AVATAR Interaccion fisica con el jugador y el chasis mediante movimientos.
31 de Enero del 2021	Se afinaron elementos del UI, se corrigio problemas de colisiones con ambas manos.
3 de Febrero del 2021	Reproducir una sola vez el sonido por tipo de pieza
5, 6 y 7 de Febrero del 2021	Agregar dos plataformas: Carroceria, asientos, volante, tablero, motor.
13 de Febrero del 2021	Se hizo un modelo sencillo del circuito temporizador con eLLM555.
13 de Febrero del 2021	Se hicieron resistencias personalizables con el codigo de colores por medio del inspector de Unity 3D.
13 de Febrero del 2021	Se implemento el funcionamiento basico de la UI y se hicieron pruebas. Sistema funciona de manera basica.
14 de Febrero del 2021	Se hicieron ajustes al modelo 3D del circuito del LM555 y se agrego animacion de Led Parpadeando.
15 de Febrero del 2021	Revision de los cambios realizados explicar componentes de manera individual en UI. caracteristicas Escena inicial con explicacion -Ensamblaje con animacion explicativa al presionar un boton. pieza por pieza. -3 ventanas -- explicacion de los componentes
18 de Febrero del 2021	Se hizo primera iteracion de la funcionalidad del cambio de UI de la explicacion
21 de Febrero del 2021	Se cambio el modelo por el circuito con MOSFET Se hicieron las animaciones para los componentes para la explicacion. Se agregaron descripciones a la escena explicacion MOSFET y se programo la funcionalidad de la misma.
21 de Febrero del 2021	Se implemento la escena en realidad aumentada, y se hicieron pruebas y ajustes necesarios
22 de Febrero del 2021	Se cambio el modelo por el circuito con MOSFET en PROTOBOARD y se hicieron pruebas en celular.
24 de Febrero del 2021	Revision de los cambios realizados
25 de Febrero del 2021	Se creo la animacion del ensamble en proto board.
26 de Febrero del 2021	Se agrego interfaz de control de la animacion a la Escena "Ensamble MOSFET".
27 de Febrero del 2021	Se ajustaron detalles. Se hicieron botones en forma de cubos. Se deshabilitan los botones fisicos al activar las ayudas visuales.
1 de Marzo del 2021	Revision de los cambios realizados
2 de Marzo del 2021	Revision de los cambios realizados
6 de Marzo del 2021	Planeacion del siguiente circuito RLC.
9 de Marzo del 2021	Circuito RLC 1R0.1/4H Henrio bobina.4/3 F de faradio. Explicacion
9 de Marzo del 2021	Elaboracion de la animacion.
9 de Marzo del 2021	Revision de los cambios realizados
14 de Marzo del 2021	PARALELO actualizar el proyecto a unity 2020 con la otra plantilla para el tracking una vez hechos los cambios OPCIONAL SP32 micro controlador interface con unity usando UDUino.
	denotar pinout de los componentes MOSFET y circuitos de 3 pines en adelante en escena explicacion AGREGAR!! control de luz en la escena explicacion y deshabilitar sombras en protoboard. Y otro para cambiar el fondo de color de la camara. Se anota el pinout del MOSFET y se agrego el slider para control de iluminacion. Se agregaron las imagenes correspondientes a los puntos de prueba Se hicieron ajustes a las escenas para AR (RLC) y tambien se hicieron pruebas en celular. Se agrego escena con menu principal.
15 de Marzo del 2021	Se hicieron correcciones en las imagenes de los diagramas en la UI
20 de Marzo del 2021	Cambios anotados en los siguientes renglones
22 de Marzo del 2021	Agregar diagramas de Kirchoff, metodo de solucion de bloques y de estado.Mostrar imagenes al presionar los Scopes.
22 de Marzo del 2021	Pruebas y elaboracion de material didactico
25 de Marzo del 2021	Cambiar colores de fondo de la camara.
24 de Marzo del 2021	Cambios anotados en los siguientes renglones
26 de Marzo del 2021	Revision de los cambios realizados
6 de Abril del 2021	Revision de los cambios realizados -IMPORTANTE ajustar modelo 3d, posicion
7 de Abril del 2021	Animar las flechas para que se activen en secuencia. Se reacomodo el boton de "Ver Diagrama"
9 de Abril del 2021	Cambios anotados en los siguientes renglones "ESTO VA EN UNA ESCENA INDEPENDIENTE mostrar componentes con codigo QR descripcion R, L y C y su simbolo electronico junto a el. mostrar codigo de colores de resistencias.
12 de Abril del 2021	
13 de Abril del 2021	Revision de los cambios realizados

	aislar imagen con animacion otro img target en otra escena desplegar formulas al tocar componentes dentro del diagrama mostrado agregando descripcion 1.1 en el capacitor 1.2 para el inductor 1.3 para resistencia dentro de 1.3, 1.4 y 1.5 DEJAR PROCEDIMIENTO EN EL ESPACIO 3D
14 de Abril del 2021	!!!!TEXT MESH PRO SUBINDEX AND SUPER INDEX!!!!
15 de Abril del 2021	Revision de los cambios realizados
14 de Abril del 2021	Revision de los cambios realizados
15 de Abril del 2021	Revision de los cambios realizados
19 de Abril del 2021	Cambios y ajustes a la escena Diagrama RLC AR
24 de Abril del 2021	Cambios a realizar en siguientes renglones
25 de Abril del 2021	
	Contrastar la ecuacion al ser agragada para resaltar. -agregar animacion de cambio de color para la ecuacion relacionada al boton que se acaba de presionar, con el fin de resaltarla de las demas. -Pruebas.
25 de Abril del 2021	
25 de Abril del 2021	Compresion y subida de videos con el procedimiento
28 de Abril del 2021	Revision de los cambios realizados
28 de Abril del 2021	poner animacion de diagrama
29 de Abril del 2021	Revision de los cambios realizados
30 de Abril del 2021	Se continuo con la animacion de las formulas
5 de Mayo del 2021	Se continuo con la animacion de las formulas y hablamos de lo hecho en FireBase Database poner una manita indicadora de donde hacer clic al inicio de la aplicacion.
10 de Mayo del 2021	Se continuo con las animaciones en la escena "Test"
12 de Mayo del 2021	Se continuo con la animacion de las formulas.
17 de Mayo del 2021	Se termino la logica de las animaciones de la escena "Test"
23 de Mayo del 2021	Se comenzo la animacion del despeje de Matrices
24 de Mayo del 2021	Se termino la animacion del despeje de Matrices y se realizo una build de prueba
25 de Mayo del 2021	Cambios ligeros a la animacion y realizacion de build de prueba
26 de Mayo del 2021	Builds de prueba y ajustes minimos
27 de Mayo del 2021	Builds de prueba y ajustes
29 de Mayo del 2021	Revision de los cambios realizados
30 de Mayo del 2021	Puebra del volante virtual y generacion de APK de prueba.
31 de Mayo del 2021	Revision de los cambios realizados
1 de Junio del 2021	Hacer QR con las imagenes de diapositivas
12 de Junio del 2021	Revision de los cambios realizados
14 de Junio del 2021	Escena "Ensamble RLCAR" agregar animacion de un multimetro midiendo en cada componente (resistencia, bobina y capacitor) mostrando imagenes como las del scope para mostrar cada una con un boton. 2 imagenes en cada medicion, simulacion y osciloscopio. (Pendientes imagenes a entregar por parte de Miguel)
14 de Junio del 2021	Revision de los cambios realizados
22 de Junio del 2021	Compilacion de dos APKs, para presentar (EnsambleRLC y Diapositivas)
26 de Junio del 2021	Se agregaron imagenes para las mediciones al Ensamble RLC AR.
26 de Junio del 2021	Importacion y exportacion de paquetes al proyecto de VR (Database y Volante)
27 de Junio del 2021	Mandar dato del volante al RealtimeDatabase
28 de Junio del 2021	Hacer escena de prueba con la UI y el oculus VR
29 de Junio del 2021	Se agrego un script al "CarBody" del final del ensamble para que cambie de escena a la VRDatabase
30 de Junio del 2021	Se agregaron imagenes actualizadas para las mediciones al Ensamble RLC AR y compilado de APK
1 de Julio del 2021	Pruebas de Oculus y cambios a realizar
4 de Julio del 2021	Pruebas de Oculus y busqueda de error en el proyecto
6 de Julio del 2021	Reversion del proyecto a Unity 2019 y reinsercion del modulo de Realtime Database
6 y 7 de Julio del 2021	modificar animacion de bobina para que mida corriente.
7 de Julio del 2021	Verificacion de integridad del proyecto Dinamicas Auto toggle para cambiar de formula a imagen en iL, vR y vC y agregar a la animacion. Y agregar al FINAL
7 de Julio del 2021	Checar carpeta Unity3D/Downloads y compilacion de APK
9 de Julio del 2021	Se cambio el repositorio a Gitlab y se detecto problema de bloqueo de puertos del ISP "Totalplay" agregar script para buscar camara de Eventos en los objetos AR al activarse cada uno. O alternativa de activar y desactivar escenas en la misma escena ya cargada. En la escena de DIAPOSITIVAS.
13 de Julio del 2021	
14 de Julio del 2021	SE SOLUCIONO cambiando todos los objetos 3D por un solo menu.
	Escalar objetos para evitar el bug de la cuadrícula de cubos generado por OVRPlayer. Se llevara todo a una sola escena ya previamente funcional.
12 de Julio del 2021	Verificar bugs de las manos (copiar y adaptar el avatar de "Modificado 3") desactivar canvas activos al hacer el toggle MANTENIMIENTO
	BUILD DE AR COMPONENTES, DIAGRAMA Y ENSAMBLE DE RLC SOLAMENTE CON MENU CADA ESCENA DE QR INDIVIDUAL 1,2 Y 3
20 de Julio del 2021	
28 de Julio del 2021	Revision de cambios en RLC AR listados arriba

3 de Agosto del 2021	Modificado 6 en VR, revisar script que quita el rigidbody y lo intenta acceder otro script, darle un delay.
3 de Agosto del 2021	Se modifico las propiedades de la presentacion AR y se hizo la build correspondiente
4 y 5 de Agosto del 2021	Se modifico las propiedades de la presentacion AR y se hizo la build correspondiente con Canvas 2D. Tambien se efectuo la build con el Menu usando QRs solamente
6 de Agosto del 2021	Dinamicas Auto: Se cambiaron lineas de codigo en el script "PlacingArea". El rigidbody se elimina con un retraso para intentar evitar el bug.
18 de Agosto del 2021	Se hicieron builds de prueba del modelo 3D ya proporcionado del Troceador
19 de Agosto del 2021	Empezar la animacion del Troceador.
20 de Agosto del 2021	Revision de los cambios realizados
22 de Agosto del 2021	Animacion del troceador sin control de reproduccion de la misma, se hizo la build de prueba
23 de Agosto del 2021	Se agrego la UI del control de secuencia del Troceador y se hicieron builds de prueba
24 de Agosto del 2021	Se hicieron ajustes a los botones de la UI del troceador, en botones siguiente y anterior.
25 de Agosto del 2021	Revision de los cambios realizados
29 de Agosto del 2021	Quitar escena RLC Ensamblado
30 de Agosto del 2021	agregar animacion de flechas de flujo de corriente del troceador en ambas direcciones y que se muestre los diagramas al activarse o las formulas S1 y S2, y se hicieron las builds y se probó en cel
31 de Agosto del 2021	Se agregaron las imagenes de los diagramas a las animaciones de los modos Buck y Boost
1 de Septiembre del 2021	Revision de los cambios realizados
1 de Septiembre del 2021	"-cambiar color de las flechas. Poner leyenda de los switches Buck boost en la parte superior central y agrandarlos y cambiar letras a color negro. Botones de anterior, quitar la palabra "Animacion" -Se cambio el nombre de la aplicacion y se agrego un icono sencillo, se hizo build de prueba
2 de Septiembre del 2021	Poner imagen en el protoboard con los diagramas de la animacion activa con flechas. Agrandar flechas. -Poner nombre a las graficas y descripcion, poner nombre nombre del archivo. Se hizo build
8 de Septiembre del 2021	Revision de los cambios realizados
20 de Septiembre del 2021	Revision de los cambios realizados
21 de Septiembre del 2021	modificar el tamaño de los modelos con una UI, hacer zoom con los dedos seria lo ideal o usar un slider
22 de Septiembre del 2021	Se agrego el control de zoom al troceador.
23 de Septiembre del 2021	Revision de los cambios realizados
26 de Septiembre del 2021	Sentido de la corriente en el RLC con flechas. al final de 03EnsambleRLC
28 de Septiembre del 2021	LM555 en el circuito buck boost, agregar al final de la animacion.
29 de Septiembre del 2021	Revision de los cambios realizados
1 de Octubre del 2021	
5 y 6 de Octubre del 2021	Cambiar UI a ingles
18 de Octubre del 2021	Elaboracion de Documento de proyecto
19 de Octubre del 2021	Elaboracion de Documento de proyecto
20 de Octubre del 2021	Revision de los cambios realizados
21 de Octubre del 2021	Elaboracion de Documento de proyecto
26 de Octubre del 2021	Revision de los cambios realizados
26 de Octubre del 2021	Activar todas las flechas del flujo de corriente al mismo tiempo. en Buck boost, segun corresponda. Quitar fondo blanco del diagrama de circuito RLC (se deshabilito el Background) Anotar etiquetas con valor de componentes en los ensambles de los circuitos (RLC listo) Reducir el diametro de los cables en el Ensamble RLC (Se redujeron a 0,77 unidades en ejes X y eje Z Ocultar resistencia de lado izquierdo en Buck Boost y poner cable
27 de Octubre del 2021	Se reemplazo el diagrama de Buck Boost plasmado sobre el protoboard
27 de Octubre del 2021	Se cambiaron los tiempos de parpadeo de las flechas indicadoras de flujo de corriente para ambos circuitos Se hicieron dos builds (una para buck boost y otra para RLC)
29 de Octubre del 2021	Se cambio la UI a ingles del EnsambleRLC. Se ajusto la UI de Troceador. Se cambiaron las imagenes del Buck-Boost. Se cambiaron las imagenes de mediciones de graficas con transparencia y se agrego una imagen de osciloscopio. Se volvieron a agregar las animaciones de las mediciones al EnsambleRLC.
31 de Octubre del 2021	
2 de Noviembre del 2021	Terminar de hacer los cambios anteriores, cambiar la UI. Cambiar color de flechas. Color Dorado
3 de Noviembre del 2021	Se agrego animacion donde se retira el cable para medir corriente, se cambiaron flechas al color dorado, se reescalo diagrama RLC, se hicieron cambios en la UI (varias escenas), Se agrego animacion fill rate a las graficas de Buck Boost.
4 de Noviembre del 2021	Se agrego cuadrícula a las imagenes del osciloscopio y se intento rotar el modelo
5 de Noviembre del 2021	Se agregaron los nombres a los componentes del troceador, se modificaron las posiciones de las graficas en la UI del Osciloscopio, se cambio de lugar el circuito de referencia Buck-Boost sobre el protoboard. Se agrego la compuerta not al modelo
9 de Noviembre del 2021	Terminar de implementar el circuito NOT
9 de Noviembre del 2021	Creacion de la build, pruebas, asi como capturas de pantalla y grabacion del funcionamiento
12 de Noviembre del 2021	Se agrego una conmutacion entre la activacion de las flechas de la compuerta NOT. Se agrego el Zoom a los componentes RLC y se removio el controlador de color de fondo de camara.
16 de Noviembre del 2021	
26 de Noviembre del 2021	Build de prueba de VR Database
1 de Diciembre del 2021	
19 de Diciembre del 2021	Revision de los cambios realizados
22,23 y 26 de Diciembre del 2021	AR explicar metodologia de trabajo y el proceso completo de la creacion de la app de Realidad aumentada para principios de Enero
30 de Diciembre del 2021	
2 de Enero del 2022	Se continuo con el documento acerca de la creacion de la Realidad Aumentada
3 de Enero del 2022	Ensamble Troceador, se agrego descripcion mas detallada al documento. -timeline y mencionar bindings en imagen.-mencionar los signal emitters usados. -describir el objeto UIHandler y su funcion. -describir los Canvas utilizados en el objeto. Animation Control. explicar Script del Image Handler.
14 de Enero del 2022	explicar Script del Toggle Buttons dentro de las ayudas visuales."
14 de Enero del 2022	Se extrajo el codigo SHA-1 para usarse en el realtime database con la nueva keystore
16 de Enero del 2022	Se hicieron varios cambios al proyecto Dinamicas Auto, se creo un nuevo repositorio y se hizo un Fork
5 de Mayo del 2022	VR funcionando

BIBLIOGRAFÍA

References

1. Mylonas, G., Amaxilatis, D., Pocero, L., Markelis, I., Hofstaetter, J., and Koulouris, P., "An educational IoT labkit and tools for energy awareness in European schools," in *International Journal of Child-Computer Interaction*, 269 2nd vol. 20, pp. 43-53, 2019, 10.1016/j.ijcci.2019.03.003
2. Marquez J., Villanueva J., Solarte Z., Garcia A., "IoT in education: Integration of objects with virtual academic communities," in *New Advances in Information Systems and Technologies*. Springer, 2nd, vol. 444 pp. 201-212, 2722016. 10.1007.978-3-319-31232-3.19
3. Jimenez Betancourt, R. O., Gonzalez Lopez, J. M., Barocio Espejo, E., Concha Sanchez, A., Villalvazo Laureano, E., Sandoval Perez, S., and Contreras Aguilar, L., "IoT-based electricity bill for domestic applications," in *Sensors*, 2nd vol. 20 no. 21, pp. 6178, 2020, 10.3390/s20216178
4. Carlos-Mancilla, M. A., Luque-Vega, L. F., Guerrero-Osuna, H. A., Ornelas-Vargas, G., Aguilar-Molina, Y., and González-Jiménez, L. E., "Educational Mechatronics and Internet of Things: A Case Study on Dynamic 278 Systems Using MEIoT Weather Station," in *Sensors*, 2nd vol. 21 no. 1, pp. 181, 2021, 10.3390/s21010181
5. Simuscuka, A. A., Markande, T. M., Muntean, G. M., "Real-virtual world device synchronization in a cloud-enabled social virtual reality IoT network," in *IEEE Access*, 2nd vol. 7, pp. 106588 - 106599, 2019, 10.1109/ACCESS.2019.2933014
6. Morelot, S., Garrigou, A., Dedieu, J., and N Kaoua, B., "Virtual reality for fire safety training: Influence of immersion and sense of presence on conceptual and

procedural acquisition,” in *Computers and Education*, 2nd vol. 166, pp. 104145, 2021, 10.1016/j.compedu.2021.104145

7. Gonzalez Lopez, J. M., Jimenez Betancourt, R. O., Ramirez Arredondo, J. M., Villalvazo Laureano, E., and 286 Rodriguez Haro, F., “Incorporating virtual reality into the teaching and training of Grid-Tie photovoltaic power plants design,” in *Applied Sciences*, 2nd vol. 9, no. 21 pp. 4480, 2019, 10.3390/app9214480

8. Santamaria-Bonfil, G., Ibanez, M. B., Perez-Ramirez, M., Arroyo-Figueroa, G., and Martinez-Alvarez, F., “Learning analytics for student modeling in virtual reality training systems: Lineworkers case,” in *Computers and Education*, 2nd vol. 151, pp. 103871, 2020, 10.1016/j.compedu.2020.103871

9. Wolfartsberger, J. “Analyzing the potential of Virtual Reality for engineering design review,” in *Automation 292 in Construction*, vol. 104, pp. 27-37, 2019

10. Radianti, J., Majchrzak, T. A., Fromm, J., and Wohlgenannt, I., “A systematic review of immersive virtual 294 reality applications for higher education: Design elements, lessons learned, and research agenda,” in *Computers and Education*, 2nd vol. 147, pp. 103778, 2020, 10.1016/j.compedu.2019.103778

11. di Lanzo, J. A., Valentine, A., Sohel, F., Yapp, A. Y., Muparadzi, K. C., and Abdelmalek, M. “A review of the uses of virtual reality in engineering education,” in *Computer Applications in Engineering Education*, vol. 28, 298 no. 3, pp. 748-763, 2020

12. Huang, W., and Roscoe, R. D. “Head-mounted display-based virtual reality systems in engineering education: A review of recent research,” in *Computer Applications in Engineering Education*, vol. 29, no. 5, pp. 1420-1435, 2021

13. Siatras, V., Athanasopoulou, L., Alexopoulos, K., Stavropoulos, P., and Mourtzis, D. “Applying the Teaching 303 Factory Paradigm and Augmented Reality Technology for Operator Training in Assembly Operations, 304 Available at SSRN 3859475, 2021.

14. Wei, X., Weng, D., Liu, Y., and Wang, Y., "Teaching based on augmented reality for a technical creative design 306 course," in *Computers and Education*, 2nd vol. 81, pp. 221-234, 2015, 10.1016/j.compedu.2014.10.017
15. Soltani, P., and Morice, A. H., "Augmented reality tools for sports education and training," in *Computers and Education*, 2nd vol. 155, pp. 103923, 2015, 10.1016/j.compedu.2020.103923
16. Ibanez, M. B., Portillo, A. U., Cabada, R. Z., and Barron, M. L., "Impact of augmented reality technology on academic achievement and motivation of students from public and private Mexican schools," in *Computers and Education*, 2nd vol. 145, pp. 103734, 2015, 10.1016/j.compedu.2019.103734
17. Rossano, V., Lanzilotti, R., Cazzolla, A., and Roselli, T., "Augmented reality to support geometry learning," 313 in *IEEE Access*, 2nd vol. 8, pp. 19679344, 2020, 10.1109/ACCESS.2020.3000990
18. Li, W., Nee, A. Y. C., and Ong, S. K., "A state-of-the-art review of augmented reality in engineering analysis 315 and simulation," in *Multimodal Technologies and Interaction*, vol. 1, no. 3, pp 17, 2017.
19. Berkemeier, L., Zobel, B., Werning, S., Ickerott, I., and Thomas, O. , "Engineering of augmented reality-based 317 information systems," in *Business and Information Systems Engineering*, vol. 61, no. 1, pp 67-89, 2019.
20. Fraga-Lamas, P., Fernandez-Carames, T. M., Blanco-Novoa, O., and Vilar-Montesinos, M. A., "A review on industrial augmented reality systems for the industry 4.0," in *IEEE Access*, vol. 6, pp. 113358-13375, 2018.
21. Mourtzis, D., Synodinos, G., Angelopoulos, J., and Panopoulos, N. "An augmented reality application for robotic cell customization," in *Procedia CIRP*, vol. 90, pp. 654-659, 2020.
22. Hayden, E., Wang, K., Wu, C., and Cao, S. "Augmented Reality Procedure Assistance System for Operator Training and Simulation," in *Proceedings of the*

Human Factors and Ergonomics Society Annual Meeting, vol. 64, 324 No. 1, pp. 1175-1180, Sage CA: Los Angeles, CA: SAGE Publications, 2020.

23. Bosch, T., Van Rhijn, G., Krause, F., Könemann, R., Wilschut, E. S., and de Looze, M. "Spatial augmented reality: a tool for operator guidance and training evaluated in five industrial case studies," in Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments, pp. 1-7, 2020.

24. Lucas, P., Vaca, D., Dominguez, F., and Ochoa, X. , "Virtual circuits: An augmented reality circuit simulator for engineering students," in 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), 330 2nd pp. 380-384, 2018, 10.1109/ICALT.2018.00097

25. Selek, M., and Kıymaz, Y. E., " Implementation of the augmented reality to electronic practice," in Computer Applications in Engineering Education, 2nd vol. 28, no. 2, pp. 420-434, 2020, 10.1002/cae.22204

26. Avilés, F. R., and Cruz, C. A., "Mobile augmented reality on electric circuits," in IEEE Computing conference, pp. 660-667, London, UK, 2017

27. Álvarez-Marín, A., Velázquez-Iturbide, J. Á., and Castillo-Vergara, M., "Technology Acceptance of an 336 Interactive Augmented Reality App on Resistive Circuits for Engineering Students," in Electronics, vol. 10 337 no. 11 pp. 1286, 2021

28. Baran, B., Yecan, E., Kaptan, B.,and Pasayigit, "Using augmented reality to teach fifth grade students about electrical circuits," in Education and Information Technologies, 2nd vol. 25, no. 2, pp. 1371-1385, 2020,10.1007/s10639-019-100019

29. Ozuag, M., Canturk, I., and Ozyilmaz, L., " Augmented Reality-Based Simulation of Some Basic Electrical 342 Circuits Which Requires Oscilloscope for Analysis without Hardware," in Journal of Circuits, Systems and 343 Computers, 2nd vol. 29, no. 6, pp. 2050096, 2020, 10.1142/S0218126620500966

30. Singh, G., Mantri, A., Sharma, O., Dutta, R., and Kaur, "Evaluating the impact of the augmented reality learning environment on electronics laboratory skills of engineering students," in *Computer Applications in Engineering Education*, vol. 27, no. 6, pp. 1361-1375, 2019
31. Singh, G., Mantri, A., Sharma, O., and Kaur, R., "Virtual reality learning environment for enhancing 348 electronics engineering laboratory experience," in *Computer Applications in Engineering Education*, vol. 29, no.1, pp. 229-243, 2021.
32. Aviles-Cruz, C., and Villegas-Cortez, "A smartphone-based augmented reality system for university students for learning digital electronics," in *Computer Applications in Engineering Education*, 2nd vol. 27, no. 3, pp. 352 615-630, 2019, 10.1002/cae.22102
33. Stark, E., Kucera, E., Haffner, O., Drahos, P., and Leskovsky, R., " Using Augmented Reality and Internet of Things for Control and Monitoring of Mechatronic Devices," in *Electronics*, vol. 9, no. 8, pp. 1272, 2020,10.3390/electronics9081272
34. Crompton, H., Burke, D., and Lin, Y. C. , " Mobile learning and student cognition: A systematic review of PK-12 research using Bloom's Taxonomy," in *British Journal of Educational Technology*, vol. 50, no. 2, pp.684-701, 2019
35. Kizilaslan, A., Zorluoglu, S. L., and Sozbilir, M., " Improve learning with hands-on classroom activities: science instruction for students with visual impairments," in *European Journal of Special Needs Education*, vol. 361 36, no. 3, pp. 371-392, 2021
36. Yechkalo, Y., Tkachuk, V., Hrunтова, T., Brovko, D., and Tron, V., " IAugmented reality in training engineering students: Teaching methods," in *CEUR Workshop Proceedings*, (<http://ceur-ws.org/>).
37. Blender, "The Blender Foundation", <https://www.blender.org/>
38. Unity, "UnityLabs", <https://unity.com>
39. ARCore, "Google", <https://developers.google.com/ar>

40. Rashid, M. H., " Power electronics handbooks," in Elsevier Butterworth-Heinemann, 4th, 2017,
41. Venkit, S. M., and Athira, P. C. Solar powered ZCS bidirectional buck-boost converter used in battery energy storage 369 systems. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT) (pp.1-5). IEEE.
42. Flextronics, https://www.emis.com/php/companyprofile/MX/Flextronics_Manufacturing_Mex_SA_de_CV_es_1223303.html
43. <https://docs.unity3d.com/es/530/Manual/>
44. <https://gamedevtraum.com/es/desarrollo-de-videojuegos/tutoriales-y-soluciones-unity/serie-manejo-general-del-motor-unity/instalar-motor-unity-3d-desde-unity-hub-instalar-modulos/>