



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Mecatrónica

Sistema de entrenamiento y capacitación en el área eléctrica de subestaciones por medio de realidad virtual

TESIS

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias en Mecatrónica

Presenta:

Ing. Jorge Armando De Haro Luna

Dirigido por:

Dr. Luis Alberto Morales Hernández

Centro Universitario
San Juan del Río, QRO
México.

14 de noviembre de 2019



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Mecatrónica

Sistema de entrenamiento y capacitación en el área eléctrica de subestaciones por medio de realidad virtual

TESIS

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias en Mecatrónica

Presenta:

Ing. Jorge Armando De Haro Luna

Dirigido por:

Dr. Luis Alberto Morales Hernández

SINODALES

Dr. Luis Alberto Morales Hernández
Presidente

Firma

Dr. Irving Armando Cruz Albarrán
Secretario

Firma

Dr. Roque Alfredo Osornio Ríos
Vocal

Firma

Dr. Juan Primo Benítez Rangel
Suplente 1

Firma

Dr. Aurelio Domínguez González
Suplente 2

Firma

Dr. Manuel Toledano Ayala
Director de la Facultad de Ingeniería

Dra. Ma. Guadalupe Flavia Loarca Piña
Directora de Investigación y Posgrado

Centro Universitario
San Juan del Río, QRO
México.

14 de noviembre de 2019

Dirección General de Bibliotecas UAQ

© 2019 - Ing. Jorge Armando De Haro Luna

All rights reserved.

Dirección General de Bibliotecas UAQ

Dirección General de Bibliotecas UAQ

Esta tesis esta dedicada a mi familia y amigos, por todo el apoyo y comprensión, motivándome a esforzarme siempre.

Agradecimientos

De la manera más atenta agradezco a mis sinodales, sobre todo al Dr. Luis Alberto Morales Hernández y el Dr. Roque Alfredo Osornio Ríos por su guía y perseverancia hacia este trabajo, además de agradecer de antemano a la Universidad Autónoma de Querétaro por facilitar el acceso del laboratorio de visión y su equipo, así como a CONACYT por proporcionar una beca para cursar el posgrado (No. de beca: 631459).

Dirección General de Bibliotecas UAQ

Abstrac

With the rise of new technologies, always looking for ways in which you can expand the forms of teaching, now with virtual reality opens a path for both training and skill training thus promoting kinesthetic learning, capturing the attention of the users as it experiences it first hand. The Oculus rift system has glasses and movement controls that will allow interaction in an immersive environment. In this way, a simulator was developed using the graphic engine “Unreal Engine” recreating an electrical substation for skill training and a classroom for training, programming a sequence for the user to carry it out and respond to a survey to validate the simulator.

Dirección General de Bibliotecas UNO

Resumen

Con el Auge de las nuevas tecnologías, siempre se buscan las maneras en las cuales se puede expandir las formas de enseñanza, ahora con la realidad virtual se abre un camino tanto para la capacitación y el entrenamiento fomentando así un aprendizaje kinestésico, capturando la atención del usuario dado que lo experimenta de primera mano. El sistema Oculus rift cuenta con gafas y controles de movimiento que permitirá la interacción en un ambiente inmersivo, de esta manera se desarrolló un simulador utilizando el motor gráfico “Unreal Engine” recreando una subestación eléctrica para el entrenamiento y un aula para la capacitación, programando una secuencia para que el usuario la realice y responda una encuesta para validar el simulador.

Dirección General de Bibliotecas UNO

Índice

Dedicatoria	I
Agradecimientos	II
Abstract	III
Resumen	IV
1. Introducción	1
1.1. Antecedentes	1
1.2. Descripción del problema	12
1.3. Justificación	12
1.4. Objetivos	12
1.4.1. Objetivo general	12
1.4.2. Objetivos particulares	12
1.5. Hipótesis	13
2. Fundamentación teórica	14
2.1. Realidad virtual	14
2.1.1. Tipos de realidad virtual	14
2.1.2. Tipos de hardware para realidad virtual	15
2.1.3. Técnicas de realidad virtual	19
2.2. Áreas de capacitación eléctrica	21
3. Metodología	22
3.1. Caso de estudio	22
3.2. Diseño de elementos 3D	23
3.3. Motor gráfico	24
3.4. Sistema virtual	24
3.5. Demostración	25
4. Resultados	28
4.1. Caso de estudio	28
4.2. Diseño de elementos 3D	28
4.3. Motor gráfico	28
4.4. Sistema virtual	34
4.4.1. Motor gráfico, uso de plantilla VR	34
4.4.2. Importar modelos 3D - Comprobación de Características	35
4.4.3. Situar modelos en el espacio virtual	35
4.4.4. Asignación como actor al modelo 3D	37
4.4.5. Otorgar a los modelos características VR	40
4.4.6. Programación de la secuencia	41
4.4.7. Programación de interfaces de usuario	48
4.5. Demostración	56
4.6. Certificación del simulador	62
5. Conclusiones	67

Índice de Figuras

1.	Laboratorio virtual del Instituto Stevens de Tecnología (Potkonjak <i>et al.</i> , 2016).	3
2.	OpenSimulator (Potkonjak <i>et al.</i> , 2016).	3
3.	Menú Equipo VEMA (Valdez <i>et al.</i> , 2014).	4
4.	Vista general de la interfaz de usuario de Wirefusion (Valdez <i>et al.</i> , 2014).	5
5.	Vista previa de la funcionalidad de intersección dinámica (Hamrol <i>et al.</i> , 2013).	5
6.	Proyección tridimensional en tiempo real de estructuras anatómicas internas en el campo quirúrgico con HoloSurgical (Bernardo, 2017).	6
7.	Personal simulado dentro del edificio virtual (Kasurinen, 2017).	6
8.	Escenarios desarrollados: (izquierda) Vista de laboratorio y (derecha) Vista de celosía (Quishpe-Armas <i>et al.</i> , 2015).	7
9.	Montaje del túnel central del Jeep renegado: simulación virtual vs. trabajo real. (Caputo <i>et al.</i> , 2017).	7
10.	Participante explorando el ambiente no inmersivo (izquierda) y el ambiente inmersivo (derecha) (Paesa <i>et al.</i> , 2017).	8
11.	Visualización de la zona muerta para el esquema de control 2 (izquierda) y los esquemas de control 3 y 4 (derecha). La caja representa el tamaño aproximado de la zona muerta (Martel & Muldner, 2017).	9
12.	Diagrama de arquitectura del sistema (Beattie <i>et al.</i> , 2015).	9
13.	Captura de pantalla de la simulación utilizada en este estudio: La imagen superior es una captura de pantalla de la versión PC y la inferior es una captura de pantalla de la RV, que muestra la tecnología de visualización estereoscópica utilizada. (Makranskya <i>et al.</i> , 2017).	10
14.	Estructura de análisis agrupada por variables asociadas al accidente. (Castillo-Rosa <i>et al.</i> , 2016).	11
15.	Imagen del dispositivo Samsung Gear VR (Samsung, 2018).	15
16.	Imagen del dispositivo Oculus rift con el sistema de controles llamados Touch (Oculus, 2018).	16
17.	Imagen del dispositivo Leap Motion (Buckwald & Holz, 2017).	18
18.	Imagen muestra del uso del GloveOne con integración del HTC Vive (NeuroDigitalTechnologies, 2017).	20
19.	Organigrama de Comisión Federal de Electricidad (CFE)	21
20.	Diagrama a bloques de la metodología general (elaboración propia).	22
21.	Diagrama a bloques del diseño de elementos 3D (elaboración propia).	23
22.	Diagrama a bloques del motor gráfico (elaboración propia).	25
23.	Diagrama a bloques del sistema virtual (elaboración propia).	26
24.	Diagrama a bloques de la demostración (elaboración propia).	27
25.	Diseño de los modelos 3D de la subestación en el software de diseño Autodesk 3ds max (elaboración propia).	28
26.	Menú de importación del motor gráfico “Unreal Engine” (elaboración propia).	30
27.	Herramienta de “Unreal Engine” para las mallas estáticas (elaboración propia).	31
28.	Modelo 3D de un casco de protección sin material ni malla de colisión (elaboración propia).	31
29.	Menú cascada del botón de los tipos de colisión (elaboración propia).	31
30.	Modelo 3D de un casco de protección sin material con malla de colisión (elaboración propia).	32
31.	Mapa de Luz del modelo 3D del casco (elaboración propia).	32
32.	Herramienta de creación de materiales (elaboración propia).	32
33.	Diseño final de los modelos 3D del aula y su importación al motor gráfico “Unreal Engine” (elaboración propia).	33

34.	Menú de “Unreal Engine” de selección de plantillas para nuevos proyectos (elaboración propia).	34
35.	Blueprints incluidos en la plantilla VR de “Unreal Engine” (elaboración propia).	34
36.	Versión alpha del ambiente de la Subestación (elaboración propia).	36
37.	Versión final del ambiente de la Subestación (elaboración propia).	36
38.	Versión final del ambiente del Aula (elaboración propia).	36
39.	Editor de blueprint de “Unreal Engine” (elaboración propia).	37
40.	Editor de blueprints sección Event Graph (elaboración propia).	38
41.	Código para habilitar al actor como objeto de interacción de realidad virtual (elaboración propia).	39
42.	Menú para la creación de varios tipos “assets” o blueprints (elaboración propia).	39
43.	Menú para la creación de clases de blueprints (elaboración propia).	40
44.	Editor de blueprint para añadir la interface de realidad virtual (elaboración propia).	40
45.	Editor de blueprint para asignar un Static Mesh (elaboración propia).	41
46.	Menú para añadir nodos al blueprint (elaboración propia).	41
47.	Caja disparador de eventos “Box Trigger” (elaboración propia).	42
48.	Localización del Level Blueprint (elaboración propia).	42
49.	Box Trigger seleccionado para hacer referencia en el Level Blueprint (elaboración propia).	42
50.	Menú para la creación de referencias a un actor o añadir tipos de eventos (elaboración propia).	43
51.	Nodo de entrada para la detección de traslape para detonar eventos (elaboración propia).	43
52.	Preparación del ambiente para el paso 1 de la prueba (elaboración propia).	44
53.	Icono del blueprint de mapa del nivel (elaboración propia).	44
54.	Colocación del trigger en su lugar para disparar el nivel 2 (elaboración propia).	44
55.	Nodos de Programación para el “Box Trigger” y carga del siguiente nivel (elaboración propia).	45
56.	Colocación del trigger para el nivel 3 y con las boquillas de baja tensión cortocircuitadas (elaboración propia).	45
57.	Colocación del trigger para el nivel 4 y con las boquillas de alta tensión cortocircuitadas (elaboración propia).	46
58.	Colocación del trigger para el nivel 5 y conexión de la terminal roja del megger a H1 (elaboración propia).	46
59.	Colocación del trigger para el nivel 6 y conexión de la terminal negra del megger a X0 (elaboración propia).	47
60.	Colocación del puente auxiliar de la boquilla X0 a la tierra física (elaboración propia).	47
61.	Ejemplo de un Menú con interacción mediante puntero de ratón (elaboración propia).	48
62.	Añadiendo componentes y clases al blueprint del actor que contendrá el widget (elaboración propia).	49
63.	Creación del widget que usará el actor previamente creado (elaboración propia).	49
64.	Editor de diseño para las interfaces de usuario (elaboración propia).	50
65.	Ejemplo en el editor de diseño para las interfaces de usuario, programación de los botones (elaboración propia).	50
66.	Panel de herramientas usadas para la diseño del Widget (elaboración propia).	51
67.	Panel de detalles de la herramienta de texto (elaboración propia).	51
68.	Panel de detalles de la herramienta de Borde (elaboración propia).	52
69.	Diseño final del Widget para la interfaz de usuario (elaboración propia).	52
70.	Interfaz de programación en blueprint para widgets (elaboración propia).	53
71.	Nodos de programación para los botones de Entrenamiento y Salir (elaboración propia).	53
72.	Nodos de programación para los botones de Adelante y Atrás (elaboración propia).	54

73.	Fragmento del código que genera y destruye el Widget del menú (elaboración propia).	54
74.	Fragmento del código que ancla el menú a la cámara y comprueba posiciones válidas (elaboración propia).	55
75.	Fragmento del código para interactuar con el widget Menú (elaboración propia).	55
76.	Usuario utilizando el software original con entrada teclado y ratón (elaboración propia). . .	56
77.	Usuario utilizando el simulador desarrollado con sistema de realidad virtual (elaboración propia).	56
78.	Vista que obtiene el usuario al usar el sistema de realidad virtual en la aula (elaboración propia).	57
79.	Usuario desplazándose hacia la zona de cálculo de transformadores trifásicos (elaboración propia).	57
80.	Usuario sosteniendo el casco con su respectiva infografía (elaboración propia).	58
81.	Usuario sosteniendo el transformador con su respectiva infografía (elaboración propia). . .	58
82.	Usuario utilizando el widget con el láser para pasar a la zona de entrenamiento (elaboración propia).	58
83.	Usuario usando el widget para pasar a la siguiente instrucción (elaboración propia).	59
84.	Vista que obtiene el usuario al usar el sistema de realidad virtual en la subestación (elaboración propia).	59
85.	Usuario realizando la primer conexión en la zona de entrenamiento (elaboración propia). . .	59
86.	Vista del usuario al completar la instrucción (elaboración propia).	60
87.	Vista del usuario al seguir las instrucciones (elaboración propia).	60
88.	Código para la obtención del tiempo para cada nivel (elaboración propia).	60
89.	Registro obtenido del tiempo en segundos al completar el primer nivel (elaboración propia). .	61
90.	Widget de instrucciones pasado con botones pequeños (elaboración propia).	66
91.	Widget de instrucciones corregido con botones más grandes (elaboración propia).	66

Índice de tablas

- 1. Recomendaciones al usar RV ((Oculus, 2018)). 27
- 2. Modelos 3D de herramienta y equipo (elaboración propia). 29
- 3. Modelos 3D del Aula y la Subestación (elaboración propia). 29
- 4. Porcentaje de mejora respecto al tiempo de los usuarios (elaboración propia). 65

Dirección General de Bibliotecas UAQ

1. Introducción

La presente tesis refiere sobre el uso de la realidad virtual como una herramienta de los métodos de enseñanza para captar la atención de las nuevas generaciones. Se ha manifestado que la generación del 2010 en adelante, denominada generación T ([González, 2019](#)), esta acostumbrada a experiencias rápidas y dinámicas, encontrando aburridos los métodos de enseñanza tradicionales. Por esta razón, la realidad virtual logra una inmersión con un ambiente realista, dándole la capacidad al estudiante de experimentar él mismo el tema desarrollado. La realidad virtual no es tecnología nueva, pero recientemente obtuvo un gran auge y ahora tiene un interés no solo comercial, si no académico, científico, social y cultural.

El sector energético es clave para el país en el crecimiento económico, las finanzas públicas, las cuentas internacionales, el desarrollo tecnológico, el ambiente, la política exterior y la seguridad nacional. Por tal motivo el sector eléctrico es fundamental ya que todas los sectores dependen de ella y ésta es la principal razón por la cual Comisión Federal de Electricidad (CFE) se debe mantener a la vanguardia para salvaguardar la generación y distribución de la energía en el país. Una empresa no es nada sin sus trabajadores y una empresa de calidad capacita y entrena a sus trabajadores, dándoles los cursos y las herramientas mas actualizadas para mejorar su productividad sin descuidar la seguridad. Para una empresa las capacitaciones fungen como inversiones, dado que conllevan altos costos además del respectivo tiempo dedicado.

Por lo tanto CFE se encuentra siempre abierta a la posibilidad de encontrar nuevas formas de capacitar a sus trabajadores, dado que las capacitaciones consistían solamente en tomar un curso de 40 horas que abarcan teoría y práctica. Aunque se debe resaltar que por el alto volumen de trabajadores en capacitación, no siempre era posible que todos practicasen con el equipo debido al tiempo. CFE consiente de este problema solicitó a la Subdirección de distribución un laboratorio virtual en el que los trabajadores pudieran completar para reforzar y complementar lo visto en el curso.

1.1. Antecedentes

La Iniciativa de Educación Inmersiva (iED) es una colaboración internacional sin fines de lucro de instituciones educativas, institutos de investigación, museos, consorcios y empresas. La Iniciativa se estableció en 2005 con la misión de definir y desarrollar estándares, mejores prácticas, plataformas tecnológicas, programas de formación y educación y comunidades de apoyo a mundos virtuales, realidad virtual, realidad aumentada y mixta, simulaciones, sistemas de aprendizaje y formación basados en juegos y entornos totalmente inmersivos como cuevas y cúpulas ([ImmersiveEducation, 2018](#)).

La Immersive Learning Research Network (iLRN) es una organización internacional de desarrolladores, educadores y profesionales de la investigación que colaboran para desarrollar el potencial científico, técnico y aplicado del aprendizaje inmersivo ([ImmersiveLearning, 2018](#)).

En la iLRN se utilizan múltiples técnicas que abarcan muchas disciplinas, incluyendo ciencias de la computación, experiencia del usuario y diseño de medios, ciencias del aprendizaje, arquitectura, desarrollo de juegos, inteligencia artificial, biología, medicina y las miles de áreas de contenido disciplinario y ocupacional donde el aprendizaje inmersivo y la capacitación pueden ser relevantes, con esto no se busca limitarse, si no definir operativamente a través de estos contextos interdisciplinarios las propiedades, los contextos, las condiciones, los procedimientos, los estados, las medidas y los resultados que convergen para crear de manera confiable una experiencia de aprendizaje inmersivo eficaz para los estudiantes [ImmersiveLearning \(2018\)](#).

Estas dos iniciativas redoblan esfuerzos en promocionar las ciencias, las tecnologías y la educación, de manera que se esté al alcance de cualquiera, pero como se sabe, la enseñanza a distancia enfrenta un

reto que es principalmente la ausencia de equipo especializado donde se pueda obtener las habilidades y la experiencia necesaria. Buscando una solución a este reto se proponen los laboratorios virtuales, donde los investigadores Potkonjak *et al.* (2016) hacen un revisado de los principales laboratorios virtuales enfocados a la ciencia, tecnología e ingeniería (CTI) y al final se enfoca en la robótica dado que engloba varias disciplinas. Dando así cuatro criterios necesarios que deben tener los laboratorios virtuales, los cuales son:

- (C1) Las interfaces de usuario de cada equipo deben ser idénticas a las de los dispositivos reales correspondientes.
- (C2) El comportamiento del sistema virtual (por ejemplo, sus variables de estado y control) debe ser equivalente al comportamiento del sistema en el paradigma físico.
- (C3) Se debe proporcionar una visualización que haga que los estudiantes se sientan como si estuvieran viendo algo realmente auténtico.
- (C4) Se debe crear un espacio de laboratorio en 3D que permita la comunicación y colaboración entre los estudiantes y con el supervisor del laboratorio (o experto en el campo).

En este artículo además se presentan las ventajas en el uso de los laboratorios virtuales:

- Ventajas del ahorro: los sistemas virtuales ofrecen a las escuelas y universidades una forma rentable de organizar el trabajo de laboratorio de alta calidad en las disciplinas de la CTI.
- Flexibilidad: diferentes experimentos virtuales (simulación) que involucran diferentes componentes (aparatos virtuales) que pueden ser fácilmente creados.
- Acceso múltiple: varios estudiantes pueden utilizar el mismo equipo virtual al mismo tiempo.
- Cambio en la configuración del sistema: es posible modificar parámetros que a menudo no pueden ser cambiados en un sistema real - en un ejemplo de robot un usuario puede cambiar los eslabones del robot, reemplazar los motores, etc.
- Resistencia al daño: en un ejemplo de robótica virtual, se permite la colisión con el entorno; también se permite la sobrecarga - en este caso el brazo del robot colapsaría y después de que el usuario reemplace los motores con algo más fuerte, el robot se levantaría de nuevo y continuaría trabajando. Así, el “daño” está permitido en el mundo virtual, abriendo así la posibilidad de aprender de los errores.
- Haciendo lo “invisible” en visible: la mayoría de los aparatos reales de laboratorio tienen una cubierta para protegerse del polvo, etc.; en la mayoría de los casos, la cubierta no se puede quitar, al menos no fácilmente. En un ejemplo robótico, esto permite la posibilidad de abrir el brazo y revelar los motores y las cajas de engranajes y observar y aprender sobre el rotor, el estator, los engranajes y otros componentes de transmisión (como correas y husillos).

Sin dejar de lado se presentan las desventajas en el uso de los laboratorios virtuales:

- Se deriva de las solicitudes impuestas a los recursos informáticos. El modelado dinámico y el modelado CAD 3D de objetos pueden ser bastante complejos y requerir altos cargos de procesamiento, especialmente si los objetos están integrados en un mundo virtual que soporte el concepto de ambiente.

- El sistema realmente no existe y, por lo tanto, no puede salir nada realmente malo. Este hecho a veces crea una actitud específica en el estudiante: falta de seriedad, responsabilidad y cuidado - los estudiantes pueden sentirse como si estuvieran jugando a un videojuego.
- En la fase final de la formación suele requerir equipo real.

A pesar de que varios laboratorios virtuales llegan a cumplir de dos a tres criterios, salta a la vista que tanto el software de código libre como el comercial, son propuestas que se siguen complementando y algunos ya han sido abandonados. Cabe destacar que todos estos laboratorios virtuales hacen uso de un teclado y ratón para interactuar dentro de estos programas o mundos virtuales con un avatar. A partir de la revisión los autores [Potkonjak et al. \(2016\)](#) le es posible identificar los aspectos más débiles de los laboratorios virtuales existentes consistiendo en los criterios C2 con la dinámica y C4 de los espacios virtuales. Por lo tanto, propone que es necesario un mayor esfuerzo de investigación y desarrollo para crear mundos virtuales avanzados basados en la dinámica de los objetos del mundo real actual pudiendo facilitar la selección e introducción de nuevos objetos. Para finalizar los autores hacen mención de las tendencias emergentes en el área de dispositivos alternativos de entrada/salida para mundos virtuales (retroalimentación háptica/fuerza, detección de movimiento, pantallas estereoscópicas) y el uso de cascos de realidad virtual como el Oculus Rift. Un buen ejemplo de la realización de los cuatro criterios definidos por los investigadores [Potkonjak et al. \(2016\)](#) se observa en la Figura 1. En la Figura 2 se presenta un ejemplo de como el entorno virtual ofrece la posibilidad de transferir conocimientos y aprender en grupo.



Figura 1: Laboratorio virtual del Instituto Stevens de Tecnología ([Potkonjak et al., 2016](#)).



Figura 2: OpenSimulator ([Potkonjak et al., 2016](#)).

Un estudio de los autores [Valdez *et al.* \(2014\)](#) presentan un escenario virtual de un laboratorio de instrumentación y medidas (Figura 3 y 4). Los estudiantes trabajan con representaciones de circuitos eléctricos enfocados en el aprendizaje de conceptos eléctricos y análisis de circuitos, así como en el estudio y desarrollo de procedimientos de acción. Su objetivo es ayudarles a aprender a pensar como ingenieros, mientras aprenden sobre la teoría de circuitos, también aprenden a aplicar las muchas habilidades de un ingeniero, como organizar la información para escribir informes técnicos y pudiendo competir entre ellos. Para esto, el montaje de operaciones se divide en dos sets: uno que consta de seis operaciones generales y otro que contiene cinco circuitos simples. La mayoría de las actividades consisten en simulaciones interactivas para construir nueve circuitos en CC (Corriente Continua) y cuatro circuitos de CA (Corriente Alterna). Su entorno VEMA proporciona una nueva y eficaz plataforma de aprendizaje que se puede utilizar de forma segura. Aprender a través del descubrimiento haciendo clic en objetos con información asociada, simulando el aprendizaje mediante el modelado de un proceso o interacción que se asemeja mucho al verdadero trabajo de laboratorio en términos de fidelidad y resultados. Refuerza el aprendizaje ofreciendo menús de ayuda, herramientas, etc., también asociados a objetos 3D. Basándose en los criterios propuestos por [Potkonjak *et al.* \(2016\)](#), el sistema VEMA entraría en los cuatro criterios pero se puede mejorar siguiendo las mismas recomendaciones.



Figura 3: Menú Equipo VEMA ([Valdez *et al.*, 2014](#)).

Continuando con la realidad virtual, siendo una tendencia emergente en aplicaciones para la ciencia como se observó en el revisado de [Potkonjak *et al.* \(2016\)](#), ningún laboratorio ofrecía realidad virtual como tal, la máxima capacidad de algunos se acercaba a simular un mundo virtual 3D dentro de una plataforma. Pero a pesar de que la realidad virtual ya tiene algunos desarrollos principalmente en los videojuegos, es aquí donde se descubrió que la realidad virtual podría tener una adaptabilidad y un uso en la ciencia. Hay que destacar que algunas disciplinas son mas flexibles en adaptar nuevas tecnologías como el área medica, un ejemplo de ello es el artículo de [Hamrol *et al.* \(2013\)](#) donde su trabajo presenta un proceso de construcción de un prototipo de aplicación educativa para la enseñanza de la anatomía con elementos de la fisiología, utilizando el entorno “EON Studio” de Realidad Virtual para realizar un atlas en tercera dimensión (Figura 5). Se describe la preparación de modelos de malla de órganos humanos individuales, junto con otras técnicas de visualización, que se utilizaron para asegurar un nivel adecuado de representación visual. También se describen las etapas de preparación de la interfaz de usuario, como la presentación del proceso de preparación de datos gráficos dinámicos - es decir, la animación de los alvéolos pulmonares - y su integración con el entorno virtual. El trabajo se centra principalmente en

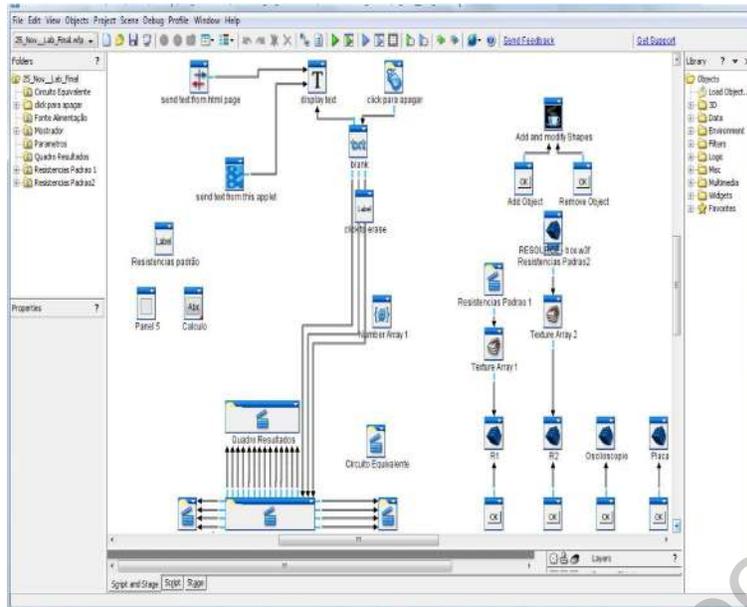


Figura 4: Vista general de la interfaz de usuario de Wirefusion (Valdez *et al.*, 2014).

la metodología de creación del tipo de aplicación “atlas virtual de cuerpo 3D”, presentando todas las etapas del trabajo de forma detallada. Su documento también describe métodos para probar el atlas y sus aplicaciones educativas planeadas. La cooperación con el equipo médico permitió evaluar la utilidad de la aplicación preparada, evaluando dos aspectos: la calidad de los conocimientos médicos contenidos en la aplicación y la practicidad de su uso.



Figura 5: Vista previa de la funcionalidad de intersección dinámica (Hamrol *et al.*, 2013).

Otra aplicación aún mas practica en el ramo de la medicina se puede ver con el investigador Bernardo (2017), donde hace uso de los avances biotecnológicos recientes, incluyendo microscopía y endoscopia tridimensional, realidad virtual, simulación quirúrgica, robótica quirúrgica y neuroimagenología avanzada, y destaca como han seguido moldeando la relación entre cirujano y computadora. Propone éste trabajo para el desarrollo de neurocirujanos, ya que estas herramientas pueden reducir la curva de aprendizaje, mejorar la comprensión conceptual de la anatomía compleja y mejorar las habilidades visuoespaciales del neurocirujano en desarrollo.

La integración de estas modalidades es beneficiosa tanto en el entrenamiento como en el desempeño de los procedimientos quirúrgicos, y cuando se combina con la simulación táctil junto con la realidad aumentada. Esta revisión destaca la tecnología que se utiliza en el entrenamiento quirúrgico moderno

para la visualización enriquecida del campo quirúrgico, y la oportunidad de aplicar y fortalecer la integración de estos datos en la neurocirugía (Figura 6). Se exploran los roles actuales y futuros y la aplicación de la realidad virtual y la simulación en el entrenamiento neuroquirúrgico.

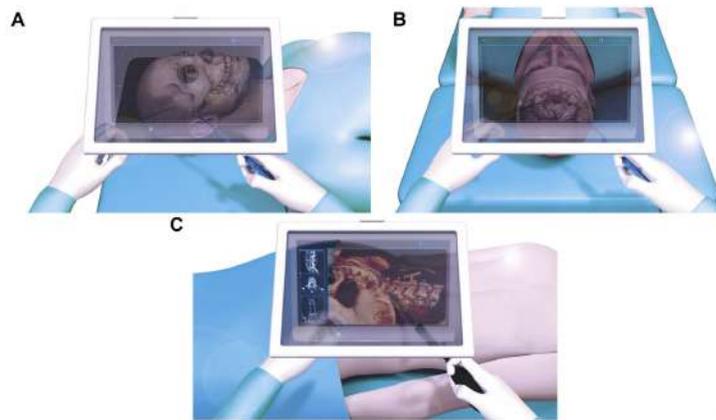


Figura 6: Proyección tridimensional en tiempo real de estructuras anatómicas internas en el campo quirúrgico con HoloSurgical (Bernardo, 2017).

Como último ejemplo el investigador Kasurinen (2017) propone un nuevo entorno de aprendizaje virtual para enseñar la mecánica y las estrategias correctas de prevención contra las amenazas comunes a la seguridad física y cibernética en la asistencia sanitaria, es decir, en un hospital (Figura 7). El investigador indica que el concepto es funcional y que a nivel de hardware existen los componentes necesarios y destaca que los problemas se encuentran en los aspectos de usabilidad e inmersión del usuario, los cuales se discuten en el documento y se estudian más a fondo en el ámbito de la investigación propuesta.



Figura 7: Personal simulado dentro del edificio virtual (Kasurinen, 2017).

Como se puede observar en el área médica hay bastantes desarrollos sobre la realidad virtual y en algunas otras apenas esta en auge tratando de descubrir el camino para integrar la realidad virtual con sus respectivas disciplinas, como el caso de los investigadores Quishpe-Armas *et al.* (2015) que presentan un entorno de aprendizaje tridimensional (3D) virtual inmersivo basado en la pantalla de cabeza Oculus rift para analizar la estructura atómica de materiales monocristalinos relevantes para los sistemas microelectro-mecánicos (MEMS), por ejemplo, silicio (Si), cromo (Cr), titanio (Ti) y cobre (Cu). Este entorno permite la visualización en tiempo real y el análisis interactivo de los parámetros clave

de la malla cristalina (Figura 8), p. ej. número de átomos en la red, factor de empaquetado atómico, densidad atómica lineal, densidad atómica superficial, densidad volumétrica, etc. ó los investigadores [Caputo et al. \(2017\)](#) que proponen la creación de escenarios virtuales 3d que permitan realizar tareas de montaje por maniqués virtuales para ser evaluados desde diferentes puntos de vista. En particular, los datos procedentes de la simulación pueden utilizarse para evaluar varios índices ergonómicos, mejorando la seguridad, la calidad y el diseño (Figura 9). Por lo tanto se presenta éste método como una innovación para el diseño centrado en el ser humano del lugar de trabajo en el desarrollo de nuevos productos, reduciendo costes y mejorando la calidad del trabajo y aumentando la producción.

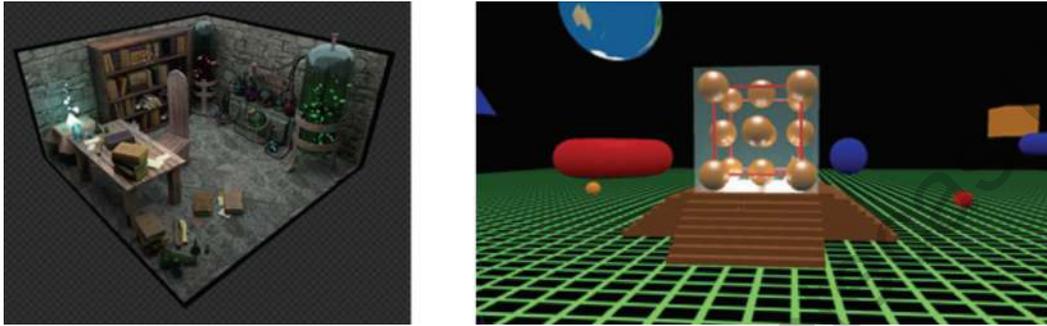


Figura 8: Escenarios desarrollados: (izquierda) Vista de laboratorio y (derecha) Vista de celosía ([Quishpe-Armas et al., 2015](#)).



Figura 9: Montaje del túnel central del Jeep renegade: simulación virtual vs. trabajo real. ([Caputo et al., 2017](#)).

Después de estas aplicaciones practicas surge una cuestión que es si el ser humano es capaz de tener una percepción espacial dentro de la realidad virtual que se asemeje a la realidad. En esto consiste el trabajo realizado por [Paesa et al. \(2017\)](#) que proponen un método que compara la percepción espacial de los usuarios usando una estación de trabajo convencional versus una plataforma inmersiva (Figura 10). Los resultados indican una mejor percepción espacial general del modelo virtual cuando se utiliza el entorno inmersivo. El estudio concluye que el entorno inmersivo puede beneficiar las prácticas de diseño actuales al mejorar la comprensión de los profesionales de la disposición espacial del modelo virtual. El método de investigación consiste en una prueba de usabilidad y rendimiento (o prueba de uso de herramientas). Se define formalmente como un diseño de experimento cuantitativo, comparativo y dentro de los sujetos, con pruebas estadísticas inferenciales. El objetivo de la prueba es medir hasta qué punto la RV envolvente apoya y mejora la capacidad de los usuarios para comprender el modelo 3D (la base del proceso de revisión del diseño). Asumiendo que el modelo 3D es una representación y pretende comunicar una idea, y también que el sistema de realidad virtual es el medio de comunicación, las pruebas de usabilidad y rendimiento pretenden verificar hasta qué punto la representación es entendida

por el usuario (o, el mensaje asimilado por el receptor). El experimento en sí mismo consiste en pedir a los participantes que identifiquen aspectos espaciales y elementos del espacio virtual como dimensiones, proporciones, aperturas, etc.; tanto en el entorno inmersivo como en el no inmersivo, respondiendo a los cuestionarios de percepción. Para determinar la efectividad del sistema inmersivo en comparación con el no inmersivo, el experimento se realiza en tres escenarios diferentes: a) en el entorno virtual no inmersivo; b) en el entorno inmersivo y c) en el entorno físico (construido). Los participantes identifican los mismos elementos en estas tres situaciones mientras completan los cuestionarios. El método de investigación utiliza el análisis estadístico inferencial (dos muestras t-test) para verificar el significado (valor-p) de las diferencias en la percepción, lo que valida los hallazgos de este trabajo.



Figura 10: Participante explorando el ambiente no inmersivo (izquierda) y el ambiente inmersivo (derecha) (Paesa *et al.*, 2017).

Ahora que ya se tiene el sentimiento de una inmersión profunda y una capacidad de percepción espacial adecuada, llega la necesidad de encontrar un hardware ad hoc que no te abstraiga de la inmersión conseguida. Por lo tanto los investigadores Martel & Muldner (2017) presentan los esquemas de control que son fundamentales para la experiencia del jugador, ya que dictan cómo interactúan los usuarios con el mundo del juego. Se comenta que desde los años 90, los juegos 3D de escritorio han utilizado un esquema de control estándar (a menudo llamado "mouselook") en el que el ratón rota la vista de la cámara, apunta y dirige el avatar del jugador simultáneamente. Los juegos de realidad virtual requieren un enfoque diferente porque las pantallas montadas en la cabeza introducen un canal de entrada adicional. Hasta la fecha, la investigación se ha centrado en la usabilidad de los dispositivos de entrada de hardware, mientras que los esquemas de control han recibido poca atención. Además, los juegos de RV todavía no tienen un esquema de control estándar y para abordar esta laguna, se realiza un estudio de métodos mixtos que comparó tres tipos de esquemas de control para los juegos de realidad virtual en primera persona y midió su efecto sobre la experiencia, el rendimiento y la inmersión del jugador (Figura 11). Se encontró que los esquemas de control familiares que combinaban el control de la cámara y la dirección del avatar, no tenían zona muerta (es decir, "ojo de cerradura" o área de la pantalla en la que no se reconocía la entrada), y permitían a los jugadores utilizar el ratón para controlar todas las funciones del juego eran preferidos y resultaron en el mejor rendimiento y las puntuaciones de inmersión. Los investigadores analizaron los Esquemas de control y los parámetros que midieron fueron los errores en el recorrido, tiempo en lograr el recorrido completo, apuntar al objetivo y destruirlo, puntuación de inmersión y valoración del esquema de control.

Aunque los investigadores Martel & Muldner (2017) concluyeron que las personas se sentían más cómodas con el esquema tradicional de control para moverse a través de un mundo virtual inmersivo, cabe destacar que no se probó con otros tipos de controles como el propuesto por los investigadores Beattie *et al.* (2015) donde presentan el Laboratorio de Interacción CAD (CIL en inglés) que emplea el



Figura 11: Visualización de la zona muerta para el esquema de control 2 (izquierda) y los esquemas de control 3 y 4 (derecha). La caja representa el tamaño aproximado de la zona muerta (Martel & Muldner, 2017).

Oculus Rift “Head Mounted Display” (HMD) y el “Leap Motion Controller” (LMC) para proporcionar un método de bajo coste que permite a los usuarios utilizar sus manos para diseccionar un modelo mecánico para manipular e inspeccionar componentes individuales en 3D realistas. Las observaciones cualitativas de las interacciones del usuario con el CIL muestran que los usuarios fueron capaces de manipular intuitivamente el modelo CAD utilizando movimientos naturales de la mano con una instrucción mínima. Los autores Beattie *et al.* (2015) presentan el diseño y la implementación del CIL (Figura 12), desarrollado para proporcionar una visualización CAD en 3D intuitiva utilizando hardware de consumo, nuevas técnicas de interacción y que puede ser fácilmente utilizado por diseñadores de todas las edades y niveles de habilidad. Propone que el trabajo a futuro podría aplicarse directamente a la educación, en particular en los campos en que se requiere el mantenimiento y el montaje de objetos grandes y/o caros.



Figura 12: Diagrama de arquitectura del sistema (Beattie *et al.*, 2015).

Una propuesta que adquirió gran relevancia fue para los educadores de enfermeras que los autores Butt & Kardong-Edgren (2018) referenciaron que se enfrentan al reto de encontrar métodos innovadores para ayudar a los estudiantes de enfermería a desarrollar y recordar habilidades fundamentales mientras se asegura la seguridad del paciente. Con el equipo de realidad virtual (VR) y la tecnología háptica personalizada combinada con principios de aprendizaje basados en juegos se puede proporcionar un enfoque innovador para promover el aprendizaje y la retención de la habilidad. Los autores plantearon una metodología que consistía en el tiempo dedicado a la tarea y el número de procedimientos completados en una hora y se compararon con los sujetos que practicaban tradicionalmente, usando un instructor de tareas con la supervisión del profesorado.

Después de que se revisaron los laboratorios virtuales, la realidad virtual y las interfaces con las que se puede interactuar y generar un ambiente aún más inmersivo, los investigadores [Makranskya et al. \(2017\)](#) predicen que la realidad virtual (RV) creará un cambio de paradigma en la educación y la capacitación, pero hay poca evidencia empírica de su valor educativo. Los principales objetivos de su estudio eran determinar las consecuencias de añadir la RV inmersiva a las simulaciones de aprendizaje virtual e investigar si los principios del aprendizaje multimedia generalizan a la RV inmersiva (Figura 13). Además, se utilizó el electroencefalograma (EEG) para obtener una medida directa del procesamiento cognitivo durante el aprendizaje. Una muestra de 52 estudiantes universitarios participó en un diseño experimental de 2x2 paneles transversales en el que los estudiantes aprendieron de una simulación científica a través de una pantalla de escritorio (PC) o una pantalla de cabeza (VR); y las simulaciones contenían texto en pantalla o texto en pantalla con narración. A través de ambas versiones de texto, los estudiantes reportaron estar más presentes en la condición VR ($d=1.30$); pero aprendieron menos ($d=0.80$), y tuvieron una carga cognitiva significativamente más alta basada en la medida EEG ($d=0.59$). A pesar de sus propiedades motivadoras (como se refleja en las calificaciones de presencia), la ciencia del aprendizaje en la RV puede sobrecargar y distraer al alumno (tal como se refleja en las mediciones del EEG de la carga cognitiva), lo que resulta en menos oportunidades de generar resultados de aprendizaje (como se refleja en el peor rendimiento de las pruebas de resultados de aprendizaje).

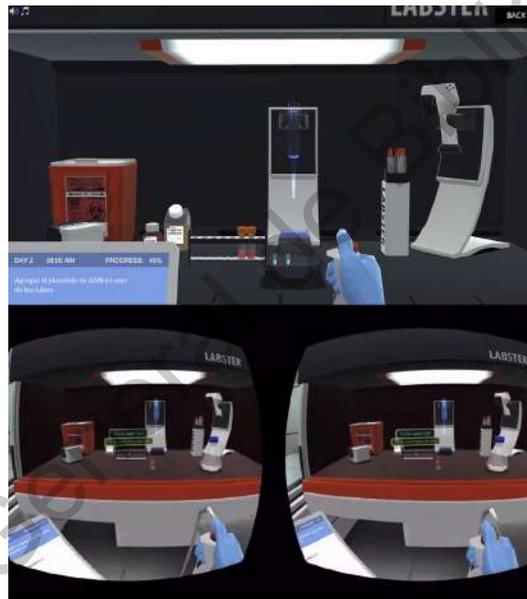


Figura 13: Captura de pantalla de la simulación utilizada en este estudio: La imagen superior es una captura de pantalla de la versión PC y la inferior es una captura de pantalla de la RV, que muestra la tecnología de visualización estereoscópica utilizada. ([Makranskya et al., 2017](#)).

Con el estudio de [Makranskya et al. \(2017\)](#) se podrían detallar los caminos a seguir para elevar la eficiencia tanto de la educación como de la capacitación, ya que se demostró que la realidad virtual tiene una gran ventaja respecto a los métodos tradicionales en varias disciplinas, como obtener el primer acercamiento a máquinas que puedan presentar un riesgo obteniendo así la experiencia de manera segura y sin riesgos de dañar equipo costoso. Como lo identificaron los investigadores [Castillo-Rosa et al. \(2016\)](#) que identificaron importantes relaciones entre los factores personales y las consecuencias de estos accidentes para varios sectores eléctricos (Figura 14). Se basa en un análisis de las relaciones de dependencia en 55 categorías clasificadas en 10 variables asociadas a 14,022 accidentes eléctricos ocurridos en España entre 2003 y 2012. Los resultados obtenidos muestran que los tres sectores de actividad se ven afectados de forma diferente por accidentes eléctricos, tanto por contacto directo como indirecto. Se confirma que hay una mayor proporción de accidentes graves y mortales en comparación

con el número total de accidentes en el país (España); esta cifra se multiplica por 22 en el caso de accidentes mortales debidos al contacto eléctrico directo en el sector primario. El análisis de los factores personales identificó relaciones estadísticamente significativas entre este tipo de accidentes y el sexo, edad, experiencia, nacionalidad y ocupación de los trabajadores. Estos resultados muestran que las autoridades competentes deben promover acciones para verificar que las instalaciones y equipos utilizados en el lugar de trabajo cumplen con la legislación aplicable en materia de protección contra el contacto eléctrico directo e indirecto. Asimismo, mencionan que es necesario diseñar estrategias para proporcionar a los trabajadores una educación y formación adaptada a sus circunstancias personales y laborales que les proteja contra los riesgos eléctricos. Por lo tanto, la investigación sobre este tipo de accidentes debe ser una prioridad, así como los esfuerzos para asegurar que los trabajadores realicen sus puestos de trabajo en entornos de trabajo seguros en relación con la electricidad ya que esta situación es en los tres sectores de actividad: primario, secundario y terciario, con 7 veces más casos con consecuencias graves y 22 veces más fatales.

Los investigadores [Castillo-Rosa et al. \(2016\)](#) resaltan la proporción de accidentes eléctricos debidos a descargas eléctricas directas (54,6 %) y de contacto indirecto (45.4 %). El sector secundario es donde ocurren la mayoría de este tipo de accidentes, probablemente debido a la mayor presencia de riesgos. Y en el terciario donde también hay un alto número de accidentes eléctricos, el factor diferenciador en este caso es el mayor número de los trabajadores expuestos.

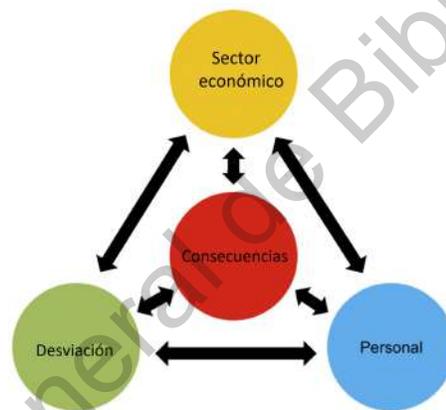


Figura 14: Estructura de análisis agrupada por variables asociadas al accidente. ([Castillo-Rosa et al., 2016](#)).

En cuanto a los factores personales, se encontró que el sexo, la edad, la experiencia, la nacionalidad y el tipo de trabajo son variables que afectan a los accidentes eléctricos. En consecuencia, las estrategias para proporcionar a los trabajadores la formación y la educación para protegerlos contra los riesgos eléctricos deben adaptarse a sus circunstancias personales y profesionales. Esta educación y formación se referiría a la identificación de los riesgos, los procedimientos de trabajo, el mantenimiento de los equipos e instalaciones, el uso de elementos de seguridad (disyuntores, puesta a tierra, protección contra rayos, etc.) y el uso de equipos de protección personal (guantes, gafas, casco, ropa, etc.).

La empresa CFE en su departamento de Subdirección de distribución, contactó con el instituto de investigaciones eléctricas para la realización de un proyecto de software propio, es aquí donde el instituto de investigaciones eléctricas evaluó los conocimientos que debería contener el laboratorio virtual. Ahora que la información ya se encontraba concentrada, faltaba la interfaz con la que el trabajador debería interactuar y el instituto de investigaciones eléctricas dejó a cargo al Grupo de realidad virtual (GRV). De esta manera nació el laboratorio virtual de Adiestramiento de líneas energizadas en tercera dimensión (Alen3D). Este programa contiene maniobras del departamento de distribución y del departamento de subestaciones y líneas. También cuentan con material de apoyo como manuales de procedimientos

como el manual de procedimientos de la instalación de cuchillas unipolares en estructura AD30 con dos canastillas, que cuenta con un código de colores de semáforo indicando el riesgo que presenta la maniobra para el trabajador y el manual de procedimientos como el Programa integral de prevención de accidentes graves (PAG).

1.2. Descripción del problema

En esta época, se ha hecho evidente la brecha educacional entre los estudiantes que cuentan con el equipo e instalaciones especializadas para la enseñanza en el área eléctrica contra los que no lo poseen, por tal motivo, la carencia de equipo e instalaciones que por sus muy altos costos de adquisición afectan el desarrollo del estudiante, repercutiendo altamente en la seguridad del mismo, dado que, no cuenta con la capacitación y entrenamiento necesario para realizar la actividad, contando que, principalmente se da una enseñanza teórica o una laxamente práctica con equipo desactualizado.

1.3. Justificación

Cada vez es más importante buscar nuevas maneras innovadoras y seguras de capacitar a los estudiantes, y la realidad virtual ofrece ambas, ya que es una tecnología que permite solucionar la carencia de equipos e instalaciones costosas al ser capaz de simularlos de manera virtual, proporcionando un espacio para la capacitación y el entrenamiento de manera segura, asequible y novedosa, ya que su modo de operación crea un ambiente de inmersión que capta la atención del estudiante, logrando superar la barrera de monotonía que algunas herramientas educativas producen con el tiempo. Por lo cual, al crear un ambiente virtual, donde se puedan realizar prácticas eléctricas sin poner en riesgo a los alumnos, equipo e instalaciones, hará que se solvete la carencia de estas en la realidad, además, se podrá realizar una infinidad de veces dado que no presentan un desgaste dentro del ambiente virtual.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar un procedimiento para la prueba de resistencia de aislamiento a transformador de potencia haciendo uso de la técnica de realidad virtual, modelando las herramientas, equipos e infraestructura necesaria. Integrar los modelos en un motor gráfico para asignar características y situar los objetos de manera que generen un entorno virtual realista. Programar los componentes de interacción e interfaces de usuario para la realización de la prueba.

1.4.2. Objetivos particulares

- * Desarrollar un ambiente virtual para la capacitación (aula) y entrenamiento (subestación eléctrica), donde los usuarios realizarán las acciones requeridas.
- * Desarrollar los modelos 3D necesarios, recreando las herramientas y los equipos sobre los cuales se hace la práctica.
- * Desarrollar la prueba de resistencia de aislamiento a transformador de potencia con sus respectivas instrucciones.
- * Jerarquizar el entrenamiento para la adquisición de habilidades y destrezas.
- * Desarrollar y aplicar un cuestionario para la validación del sistema a los usuarios.

1.5. Hipótesis

Desarrollando un ecosistema, que integre la prueba de resistencia de aislamiento a transformador de potencia y la realidad virtual para el entrenamiento y capacitación del usuario, es posible mejorar la obtención de habilidades y destrezas al utilizar un ambiente altamente realista, por lo cual permitirá el acceso a equipo e instalaciones costosas al poder hacer uso de ellas sin costo extra ni riesgos.

Dirección General de Bibliotecas UAQ

2. Fundamentación teórica

2.1. Realidad virtual

Fue Krueger (1991) a finales de los años 60 en usar el termino “Realidad artificial” para describir sus entornos inmersivos e interactivos, basados en técnicas de reconocimiento de video, que ofrecían al usuario contacto total y sin estorbos con el mundo digital, por lo tanto, es considerado como una Figura clave para la innovación temprana de la realidad virtual. En la actualidad el termino realidad artificial se suele utilizar para describir una realidad virtual que es indistinguible de la realidad. Es lo que distingue la “Realidad artificial” de “Realidad virtual”, que se suele aplicar a la tecnología que es como la realidad pero fácilmente identificable como una simulación.

El término realidad virtual (RV) se popularizó a finales de la década de 1980 por Jaron Lanier, uno de los pioneros del campo. Entonces la Enciclopedia Británica Lowood (2015) describe la realidad virtual como “el uso del modelado y la simulación por computadora que permite a una persona interactuar con un entorno sensorial tridimensional (3D) artificial u otro entorno sensorial”. Además, establece que “las aplicaciones de realidad virtual sumergen al usuario en un entorno generado por computadora que simula la realidad mediante el uso de dispositivos interactivos, que envían y reciben información y se usan como gafas, auriculares, guantes o trajes para el cuerpo”. Por ejemplo, los autores Witmer & Singer (1998) comentan que un usuario que usa una pantalla montada en la cabeza con un sistema de proyección estereoscópica puede ver imágenes animadas de un entorno virtual generando un término importante que es la presencia o telepresencia, que se puede describir como una ilusión de “estar allí”.

Pero el verdadero origen de la realidad virtual se remonta a la Segunda Guerra Mundial. Los Autores Guillen *et al.* (2007) redactan que la Marina de Guerra de Estados Unidos contacta con el MIT (Massachusetts Institute of Technology) para la posible creación de un simulador de vuelo apto para el entrenamiento de pilotos de bombarderos. El proyecto fue denominado Whirlwind y su construcción finalizó algunos años más tarde en 1951. No fue hasta 8 años después cuando USAF (United States Air Force) retomó el proyecto bajo el nombre de “Claude Project” y apareció un uso civil de la tecnología 3D.

2.1.1. Tipos de realidad virtual

La realidad virtual puede llevarse a cabo a través de diferentes métodos como pueden ser: un simulador o un avatar, la proyección de imágenes reales, mediante un ordenador o la inmersión en un entorno virtual.

- Simuladores: El primer tipo es a través de un simulador de realidad virtual. Los simuladores de conducción de vehículos, por ejemplo, dan a los usuarios a bordo la impresión de que están llevando un vehículo real, ya que predice el movimiento del vehículo al dar una orden y recibir la correspondiente respuesta visual y auditiva (apretamos el acelerador y vemos en la pantalla cómo el coche se mueve más rápido y escuchamos como suben las revoluciones del vehículo). Los simuladores se han estado utilizando de forma efectiva para desarrollar sistemas, para mejorar la seguridad y estudiar factores humanos.
- Avatares: Con los avatares los usuarios pueden unirse al entorno virtual de dos formas: 1) Eligiendo un avatar prediseñado con gráficos de ordenador. 2) Realizando una grabación de sí mismo a través de un dispositivo de vídeo. En el caso de la grabación a través de una cámara web, el fondo de la imagen se elimina para contribuir a una mayor sensación de realidad. La realidad virtual a través

de avatares mejora la interacción entre la persona en sí y el ordenador, ya que esta forma es más efectiva que el sistema convencional de ordenador de escritorio.

- **Proyección de imágenes reales:** En la proyección de imágenes reales aplicadas en la realidad virtual, el diseño gráfico de entornos reales juega un papel vital en algunas aplicaciones como por ejemplo: Navegación autónoma y construcción del diseño gráfico de simuladores de vuelo. Este tipo de RV está ganando popularidad sobre todo en gráficos diseñados por ordenador, ya que mejora el realismo utilizando imágenes foto-realistas y el proceso de modelado es bastante más sencillo. A la hora de generar modelos realistas, es esencial registrar con exactitud datos en tres dimensiones (3D). Normalmente se utilizan cámaras para diseñar pequeños objetos a corta distancia.
- **Por ordenador:** Este tipo de realidad virtual conlleva mostrar un mundo en tres dimensiones en un ordenador ordinario sin usar ningún tipo de sensor de movimiento específico. Muchos juegos de ordenador actuales utilizan recursos como personajes y otros dispositivos con los que se puede interactuar, para hacer sentir al usuario parte del mundo virtual. Una crítica común a este tipo de inmersión es que no se tiene sentido de visión periférica, ya que el conocimiento que el usuario tiene de lo que pasa a tu alrededor se limita a su entorno más cercano.
- **Inmersión en entornos virtuales:** La mejor opción para vivir la RV es a través de una interfaz cerebro-máquina, que permite una comunicación directa entre el cerebro y un dispositivo externo. Un paso intermedio sería producir un “espacio virtual” usando un casco de realidad virtual donde las imágenes que aparecen en el casco están controladas a través de un ordenador. Los únicos límites son la propia capacidad del ordenador que sirva la experiencia, la calidad de las gafas RV y el contenido disponible en la plataforma de realidad virtual.

2.1.2. Tipos de hardware para realidad virtual

Cascos o gafas

Conocidos también como “HMD” (del inglés head-mounted display o Pantalla Montada en la Cabeza en español), se distinguen fundamentalmente dos tipos: los que llevan pantalla incorporada y los que son esencialmente una carcasa destinada a que el usuario introduzca un smartphone como el dispositivo Gear Vr, ver Figura 15.



Figura 15: Imagen del dispositivo Samsung Gear VR ([Samsung, 2018](#)).

La escritora [Charara \(2016\)](#) redacta que en cuanto al display, solía utilizarse tecnología “LCD” (del inglés Liquid Crystal Display o Pantalla de Cristal Líquido en español), aunque empiezan a aparecer algunos como el Razer OSVR HDK 2, el propio PlayStation VR, o el nuevo Oculus Rift (Figura 16) con pantallas “OLED” (del inglés organic light-emitting diode o Diodo Orgánico de Emisión de Luz en español). Mientras que algunos “HMD” utilizan dos displays “LCD” (uno para cada ojo), otros optan por un único display con una división en el centro. Algunos tienen unas lentes colocadas entre los ojos y el display, y pueden ajustarse a la distancia de los ojos. Las lentes modifican la imagen para cada ojo, cambiando el ángulo de la imagen 2D de cada display para crear un efecto 3D, simulando las diferencias con las que se ven las cosas con un ojo respecto al otro.



Figura 16: Imagen del dispositivo Oculus rift con el sistema de controles llamados Touch ([Oculus, 2018](#)).

Otro aspecto importante de los “HMD” es el campo de visión. Como lo comenta [Jay \(2016\)](#) los seres humanos tienen un campo de visión horizontal de unos 180° a 220° , en ocasiones más, aunque varía de persona a persona. Esta visión es monocular, es decir solo es percibida por uno de los dos ojos. El campo de visión percibido por ambos ojos (y que por tanto se ve en 3D) es de unos 114° . Por este motivo, un campo de visión de 360° sería innecesario. La mayoría de los “HMD” funcionan con un campo de visión de entre 110° y 120° .

Por último [Charara \(2016\)](#), destaca dos puntos: los fotogramas por segundo (FPS) y la latencia. Es imprescindible un mínimo de 60 “FPS” para que el ojo perciba las imágenes de manera natural y no provoque mareo. Todos los “HMDs” importantes superan este mínimo. El otro punto es la latencia, que ha de ser inferior a 20 ms para que el usuario no experimente una sensación de retraso entre lo que hace y lo que ve.

En internet, se encuentran algunos modelos de gafas con pantalla incorporada que se venden de manera comercial, tales como:

- Oculus Rift: aparato de realidad virtual para usos tanto lúdicos como profesionales, desarrollado por la empresa “Oculus VR” (adquirida por Facebook por casi 2 mil millones de euros en 2014). Está en fase de desarrollo pero se puede comprar su modelo experimental. Funciona conectado a un ordenador, donde se ejecuta el software, lo que le permite aprovechar toda la potencia de aquel para su recreación del mundo virtual ([Oculus, 2018](#)).
- Playstation VR: conocido también como Morpheus, es un casco de realidad virtual actualmente en desarrollo por Sony. Está diseñado para ser plenamente funcional con la consola “Playstation 4” y “Playstation 4 Pro”. Puesta en venta el 13 de octubre de 2016 ([Playstation, 2019](#)).
- HoloLens 2: gafas de realidad aumentada y realidad mixta en desarrollo por Microsoft dentro de su plataforma “Windows Holographic”. Presentadas al público en 2015. A diferencia de otras, llevan incorporado su propio hardware de procesamiento y su sistema operativo (Windows), por lo

que son independientes de cualquier aparato externo. Utiliza su propia plataforma que se ha bautizado “Windows Holographic”, la cual fue abierta a otros fabricantes a principio de junio de 2016 ([Microsoft, 2019a](#)).

- Vive: proyecto conjunto de “Valve Corporation” y “HTC”, actualmente en desarrollo, de un “HMD” con una resolución anunciada de 1080x1200 para cada ojo, tasa de refresco de 90 Hz (Hertz), y más de 70 sensores de posición y orientación. Forma parte del proyecto “SteamVR” de Valve ([HTC, 2019](#)).
- StarVR: desarrolladas por Starbreeze Studios y compatibles con el sistema abierto “SteamVR”. Destacan por su amplio campo de visión (210°) al utilizar dos pantallas de 2560 x 1440, una para cada ojo ([StarVR, 2019](#)).
- FOVE VR: poseen la peculiaridad de incorporar seguimiento del (seguimiento ocular), lo que permite cosas como enfocar la imagen de acuerdo con el lugar adonde se mira, o nuevas formas de interacción visual ([Fove, 2016](#)).

Ahora se describirán las Gafas de RV móvil:

- Gear VR: aparato de realidad virtual desarrollado por Samsung en colaboración con “Oculus VR”. A diferencia del Oculus Rift, no incluye pantalla, sino que es mayormente una carcasa con algunos botones y un sensor avanzado de movimiento, concebida para colocar en ella teléfonos avanzados de la propia Samsung que harán las funciones de pantalla y de procesador informático ([Samsung, 2018](#)).
- Daydream View: visor de la plataforma de realidad virtual “Daydream” desarrollada por Google. Lanzado en noviembre de 2016 y compatible con unos pocos dispositivos certificados, va acompañado de un mando con sensores de orientación. Primeros servicios presentados son “Google Street View” o una experiencia de cine ([Google, 2019b](#)).
- Cardboard: carcasa de cartón desarrollada por Google y destinada a poder experimentar de manera barata la realidad virtual a nivel doméstico colocando en ella un smartphone de cualquier marca ([Google, 2019a](#)).
- Carcasas de plástico y otros materiales: con una funcionalidad similar a las Cardboard de Google, existen numerosas carcasas de otros fabricantes, tanto en cartón (los modelos más baratos) como en plástico. Ejemplos de esto último son las Homido ([HMD-Tech, 2018](#)), Durovis Dive ([Durovis, 2019](#)), Lakento ([Lakento, 2019](#)), VR One de Zeiss ([Zeiss-international, 2019](#)), y un largo etcétera.

Sensores de posición

Los “HMD” más avanzados se venden acompañados de unos dispositivos conocidos como sensores de posición que, colocados en la habitación, permiten al sistema determinar la ubicación del “HMD” y de otros periféricos que pueda portar el usuario, dándole así a este la posibilidad de moverse libremente en el espacio virtual.

Entre los más conocidos están el “Lighthouse” utilizado por las gafas “HTC Vive”, o el “Constellation” usado por las Oculus Rift. Compatible con otros sistemas es “Nolo VR” ([Nolo, 2019](#)), un sistema de seguimiento de posición para visores de móvil que se compone de una estación base, un marcador para el visor, y dos mandos, y es compatible con juegos de “Steam VR” ([Valve, 2019](#)).

Controladores

Los sistemas de realidad virtual suelen incorporar dispositivos de control que permitan interactuar con el entorno visualizado, y que consisten normalmente en unos mandos con botones que se agarran con las manos y que tienen seguimiento posicional absoluto. Así es el caso de los “Touch” de Oculus, o los mandos del “HTC Vive” o los del “PSVR” de Sony. También existen guantes, o bien sensores de posición capaces de detectar la posición del cuerpo o partes de este.

- Leap Motion: accesorio consistente en un sensor que percibe a distancia los movimientos de la mano, convirtiéndola así en un dispositivo de entrada (un controlador), ver Figura 17.
- STEM System: sistema para la detección inalámbrica de los movimientos corporales, desarrollado por la empresa Sixense. Supone una ampliación sobre el controlador “Razer Hydra”, del mismo fabricante (Sixense) ([Sixense, 2019](#)).
- PrioVR: similar en funcionalidad al anterior, se ocupa de trasladar al entorno virtual los movimientos corporales del usuario ([Yost-labs, 2019](#)).
- Gloveone: dispositivo háptico de realidad virtual desarrollado por “NeuroDigital Technologies”. Es un guante que permite recibir sensaciones hápticas mediante diez actuadores dispuestos estratégicamente en las yemas de los dedos y la palma de la mano con la finalidad de hacer llegar al usuario la sensación de tener o sentir un objeto en su propia mano. Gloveone además proporciona seguimiento de manos y dedos por sí mismo, aunque es 100% compatible con “Leap Motion” y “RealSense”. permite interactuar con objetos mostrados en la pantalla de un ordenador o en un “HMD” como las “Oculus Rift”, “Gear VR”, “Vive” u “OSVR” ([NeuroDigitalTechnologies, 2017](#)).
- Kinect: es “un controlador de juego libre y entretenimiento” creado por Alex Kipman, desarrollado por Microsoft para la videoconsola “Xbox 360”, y desde junio del 2011 para PC a través de “Windows 7” y “Windows 8”. “Kinect” permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador de videojuegos tradicional, mediante una interfaz natural de usuario que reconoce gestos, comandos de voz, y objetos e imágenes ([Microsoft, 2019b](#)).



Figura 17: Imagen del dispositivo Leap Motion ([Buckwald & Holz, 2017](#)).

2.1.3. Técnicas de realidad virtual

Seguimiento de cabeza

El seguimiento de cabeza como lo menciona la escritora [Charara \(2016\)](#) permite a una aplicación reconocer los movimientos de cabeza del usuario, y realizar un desplazamiento de la imagen cuando éste mueve la cabeza en cualquier dirección. Para realizar este seguimiento se utilizan unos acelerómetros, giroscopios y magnetómetros incorporados en los “HMDs”. Además, cada compañía utiliza una técnica propia para determinar la posición de la cabeza.

El “Oculus Rift” utiliza su propio sistema de posicionamiento llamado “Constellation”. Consiste en un conjunto de veinte ledes infrarrojos colocadas alrededor del casco formando un patrón reconocible y un sensor. El sensor va captando fotogramas y analizando la posición de todos los ledes, permitiendo así el seguimiento.

Algo parecido es lo que usa “PlayStation VR” (PSVR), excepto que son solo nueve ledes. La desventaja del “PSVR” es que ha de ajustarse con la cámara cada vez que una persona de diferente estatura (por ejemplo) lo utiliza. Además, la “PlayStation Camera”, necesaria para poderlo utilizar, ha de estar bastante cerca del usuario para funcionar bien. De hecho, Sony recomienda que se utilice el “PSVR” sentado, a aproximadamente 1.5 metros de la cámara y con espacio suficiente para realizar algunos movimientos ligeros. De hecho, a partir de esta distancia el rendimiento disminuye, y Sony no garantiza que la cámara detecte correctamente el movimiento a partir de los 9.8 pies (unos 3 metros).

El método que utiliza las “Vive” es bastante más novedoso como lo describe [Buckley \(2015\)](#). Se trata de un sistema de seguimiento llamado “Lighthouse”, desarrollado por “HTC” y “Valve”. No requiere de ninguna cámara, y el “HMD” no emite luz. El sistema consiste en dos cajas que se colocan en la pared con un ángulo de 90°, estas cajas contienen unos ledes y dos emisores de láseres, uno horizontal y uno vertical. Por otro lado, el “HMD” y los dos mandos (son necesarios dos para poder determinar la posición de ambas manos y brazos) disponen de sensores que captan la luz y los láseres emitidos por las cajas que se sitúan en las paredes de la habitación. Los ledes se iluminan y los dispositivos receptores empiezan a contar. Uno de los dos láseres emite un barrido por toda la sala. Los dispositivos detectan que sensores han sido alcanzados por el barrido y cuánto tiempo ha pasado desde el flash de los ledes y utilizan esta información para calcular su posición respecto a las cajas. Al acercarte demasiado a un muro, una cuadrícula translúcida aparece avisando de que estás cerca de una pared real. Todo esto con un “jitter” (la imprecisión de las mediciones cuando el objeto está inmóvil) de tan solo 0.3 mm.

Rastreo de movimiento

Aparte del prometedor y ya mencionado Lighthouse de Valve ([Buckley, 2015](#)) existen otras opciones, por ejemplo el Leap Motion Orion de [Buckwald & Holz \(2017\)](#). Éste es un sistema extremadamente preciso de seguimiento de las manos. Detecta todos los movimientos de los dedos y las articulaciones incluso sobre entornos difusos y con niveles variables de luz, aunque tiene algunas desventajas, como el hecho de que has de estar mirando tus manos para que el sistema las detecte. Otro problema, no exclusivo de Orion, es la falta de algo tangible en las manos. En la vida real, cuando se entra en contacto con algo, el sentido del tacto se activa y se siente ese algo. En la realidad virtual en cambio, las manos están vacías y por tanto no se tiene forma de saber si se está sujetando el objeto de la manera que se quiere, o la fuerza que se está aplicando sobre él. Los desarrolladores están intentando suplir esta falta de respuesta táctil mediante señales auditivas que indiquen cuándo y cómo se entra en contacto con un objeto, pero la sensación no es la misma.

La alternativa de Oculus es Touch (Figura 16), un sistema de control que consiste en dos mandos empuñados y con una correa de sujeción para la muñeca, con los que se hace sentir al usuario que está usando sus propias manos. Cada uno de estos dos controles tiene forma de medialuna y dispone de dos botones, un mando analógico y un gatillo analógico, además de un mecanismo denominado

disparador de mano, que replica la sensación de disparar un arma. Touch también hace uso del sistema de posicionamiento Constellation y a diferencia del Orion de Leap Motion (Figura 17), sí que dispone de respuesta táctil. Los mandos además disponen de unos sensores que permiten detectar una serie de gestos con las manos, como cerrar el puño, señalar con el índice o alzar el pulgar. La desventaja de Touch respecto a Orion es que, a pesar de ser muy avanzado, no deja de ser un mando y por tanto queda lejos de la libertad de movimiento que ofrece este último.

También cabe destacar la contribución de la empresa española [NeuroDigitalTechnologies](#) (2017) con su GloveOne (Figura 18). Es un guante que pretende dar al usuario ese feedback táctil tan deseado. Actualmente no dispone de sistema de seguimiento propio, así que se vale de un Leap Motion para ello, pero permite al usuario percibir el peso, la forma, el volumen y la textura de los objetos con los que interactúa. Para ello se vale de unos sensores situados cerca del pulgar, índice y los dedos centrales, además de la palma de la mano. Además, contiene 10 actuadores distribuidos entre la palma y las puntas de los dedos. Cada uno de ellos vibra de manera individual, con distintas frecuencias e intensidades, reproduciendo de manera precisa las sensaciones del tacto. Actualmente el GloveOne tiene integración con varios sistemas de seguimiento como el Lighthouse de Valve y Touch de Oculus rift.



Figura 18: Imagen muestra del uso del GloveOne con integración del HTC Vive ([NeuroDigitalTechnologies](#), 2017).

Existen otros sistemas de rastreo de movimiento, como trajes, controles por voz o incluso cintas de correr como Virtuix Omni, que permiten al usuario explorar grandes distancias caminando (o corriendo).

Seguimiento ocular

Se trata de una tecnología que las principales compañías no han incorporado aún, pero que está presente en el HMD [Fove](#) (2016) VR. Este HMD incorpora unos sensores infrarrojos interiores que captan los movimientos del ojo. Esto permite un abanico de opciones que van desde replicar los movimientos de tus ojos en tu avatar virtual, hasta provocar reacciones de otros personajes según la manera en la que los miras. Lo que es más impresionante es el realismo que ofrece el seguimiento ocular.

En la vida real, los ojos tienen un punto de enfoque central, mientras que el resto está desenfocado. Esto es muy difícil de replicar, lo que provoca un exceso de enfoque en los sistemas de otras compañías, que reduce la sensación de inmersión. El seguimiento ocular soluciona este problema, permitiendo enfocar solo aquello que el usuario está observando. Además, podría dar lugar a hipotéticas optimizaciones: la aplicación podría utilizar sus recursos en un renderizado de alta calidad de los objetos que están en el campo de visión del usuario, aplicando pocos recursos para todo aquello que está desenfocado en ese momento. Esta tecnología requiere no obstante de pantallas de alta resolución, ya que el punto enfocado por el usuario debería ser lo más realista posible. El exceso de enfoque de los otros sistemas puede producir mareo por movimiento como se presentó en [Charara](#) (2016), algo que el seguimiento ocular también podría evitar.

2.2. Áreas de capacitación eléctrica

Dado que la empresa mas importante en generación y distribución en el país es Comisión federal de electricidad (CFE), fue pertinente acercarse a ellos y aún más con la facilidad de contar con una relación directa con ellos que se pudo tener acceso a manuales de procedimientos y software de la empresa, como se observa en la Figura 19, que corresponde al organigrama de Comisión Federal de Electricidad para observar que departamento era el más apropiado a simular, ya sea por los riesgos que se toman como la forma en la que los capacitan, se eligió el departamento de distribución en la oficina de subestaciones y líneas.

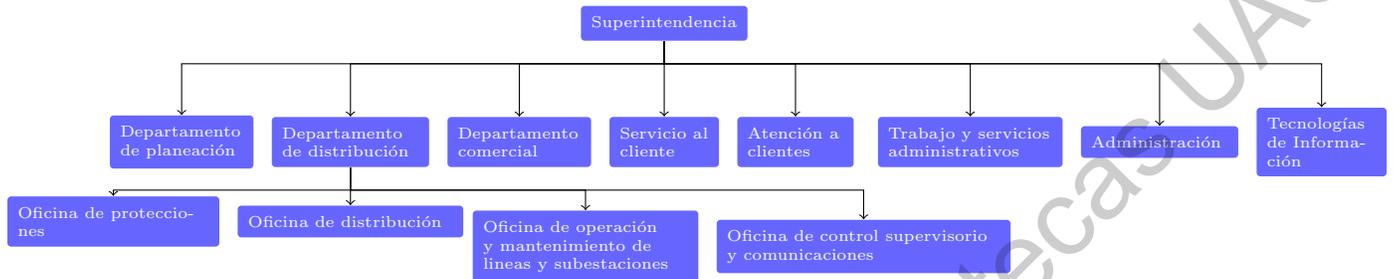


Figura 19: Organigrama de Comisión Federal de Electricidad (CFE)

Oficina de subestaciones y líneas

Es responsable de realizar y llevar a cabo los programas de mantenimiento, preventivos y correctivos a las líneas de subtransmisión y a los transformadores e interruptores de potencia, con la finalidad de evitar fallas que afecten la continuidad y calidad del servicio de energía eléctrica a los clientes. Dicha oficina, realiza trabajos en línea energizada o en línea desenergizada según sea la situación de riesgo de los trabajadores, sin discontinuar el servicio. Los trabajadores que laboran en esta oficina deben estar siempre debidamente capacitados para evitar accidentes de trabajo.

Por tal motivo los trabajadores primerizos deben tomar un curso de 40 horas en Guanajuato, aunque ahora se puede tomar por Internet en formato de videoconferencia. Terminando ese curso se les solicita que realicen la capacitación a través del software Alen3D para complementar la capacitación y familiarizando al trabajador con los procedimientos requeridos en las maniobras, cabe destacar que CFE cuenta con infinidad de manuales de procedimientos como el Programa integral de prevención de accidentes graves (PAG) ó el manual de procedimientos de la instalación de cuchillas unipolares en estructura AD30 con dos canastillas.

3. Metodología

Con la metodología propuesta se desarrollará un simulador de realidad virtual donde el usuario se podrá capacitar y entrenar al realizar una actividad de alto riesgo en un ambiente seguro, con el fin de obtener y mejorar sus habilidades y destrezas. Se diseñará un aula y una subestación eléctrica en un ambiente virtual que contenga todos los elementos y herramientas, pudiendo el usuario mediante controles de movimientos manejar las manos virtuales y realizar la prueba empleando el aprendizaje kinestésico.

Como se puede observar en la Figura 20, representa los pasos a seguir para validar la hipótesis propuesta. Así que de manera general se procederá a desarrollar el procedimiento:

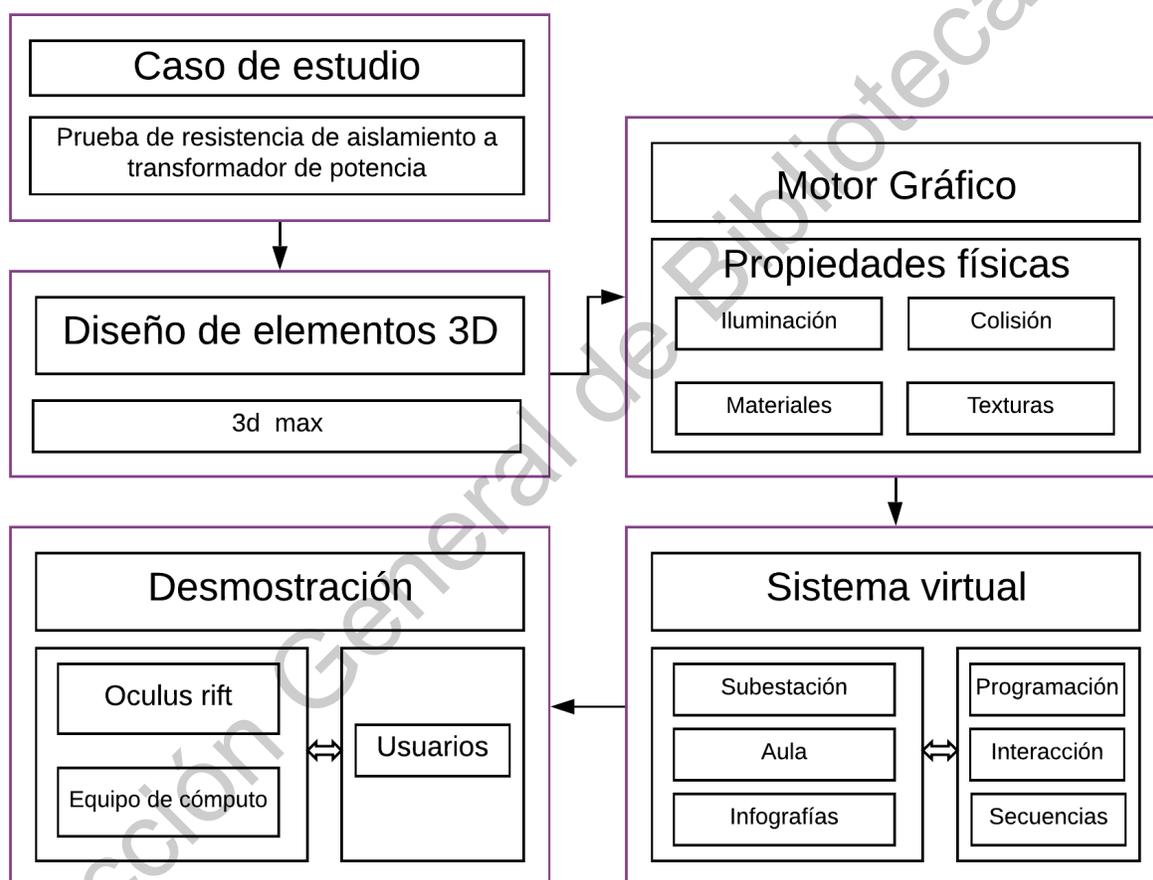


Figura 20: Diagrama a bloques de la metodología general (elaboración propia).

3.1. Caso de estudio

Se analizarán las competencias a aprender, estudiando los procedimientos, instalaciones, herramientas y equipos necesarios para llevar a cabo con éxito las pruebas, seleccionando de esta manera la más adecuada a ser desarrollada como un primer acercamiento en el desarrollo de la realidad virtual, en este caso se realizará la prueba de resistencia de aislamiento a transformador de potencia.

3.2. Diseño de elementos 3D

En esta sección se llevará acabo el diseño de los modelos virtuales 3D (Figura 21), tanto del equipo, instalaciones y herramientas previamente seleccionadas a causa del caso de estudio. Por ejemplo, algunos de los modelos básicos son:

- * Casco.
- * Segueta.
- * Conos delimitadores.
- * Cubeta de lona.
- * Guantes.
- * Desarmador de cruz.
- * Juego de llaves mixtas.
- * Juego de dados.

Cabe mencionar que estos modelos son propuestos debido que son comúnmente encontrados dentro de un laboratorio y forman parte de la herramienta básica. Esto no limita que las herramientas o equipos funcionen como un mero accesorio estético para generar la sensación de un ambiente, como un aula o una subestación, que servirán como trasfondo para asemejar la realidad logrando una inmersión bastante alta.

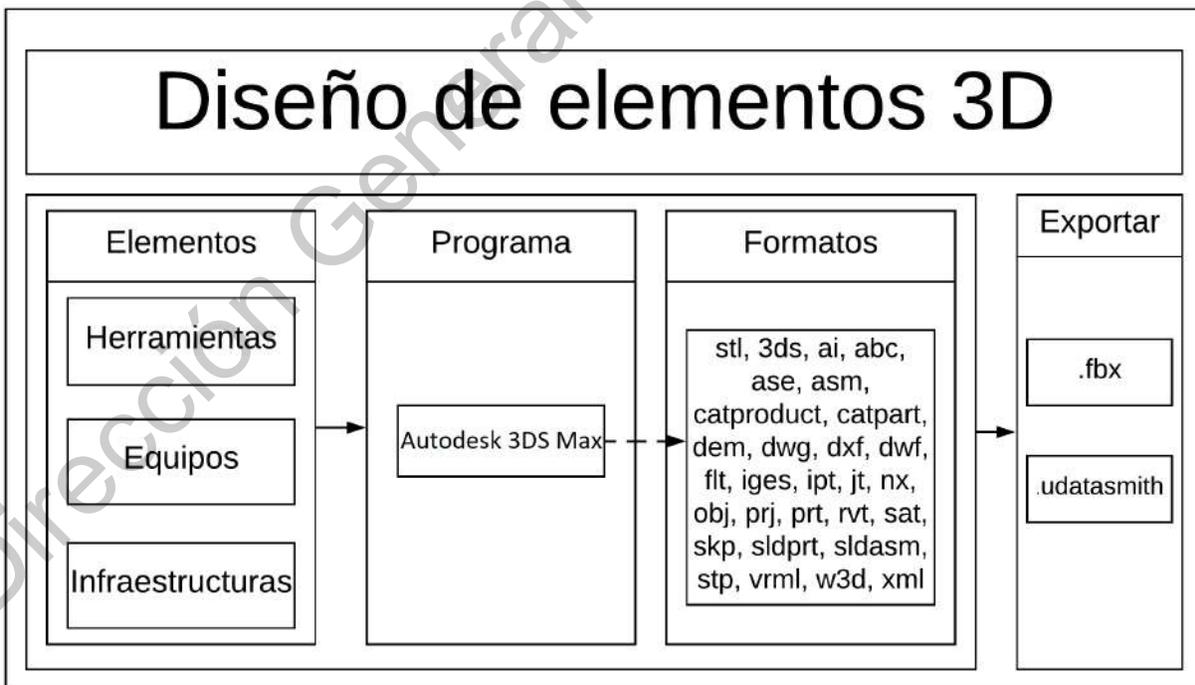


Figura 21: Diagrama a bloques del diseño de elementos 3D (elaboración propia).

Algunos modelos del aula serían:

- * Infraestructura 4 paredes y un techo
- * Mesas
- * Luces/farolas
- * Repisas
- * Marcos

La subestación contaría con algunos modelos como:

- * Prado
- * Bardas
- * Estructura metálica que soporta las líneas aéreas
- * Aisladores
- * Transformadores
- * Contactores
- * Cuchillas

Los software de diseño 3D son variados, como Blender, cinema 4D, Autodesk 3ds Max, etc. Estos software tienen la capacidad tanto de diseño 2D, 3D e isométrico, pudiendo realizar renders, animaciones y cinemáticos además de contar con herramientas para simular materiales y texturas en los modelos realizados, otra característica que se debe tener en cuenta al elegir un software de diseño es que algunos son comerciales como cinema 4D, Autodesk 3ds Max y otros de uso libre como Blender que lo mantiene una comunidad de programadores. En este caso y como lo indica la Figura 21, se hizo uso del software de Autodesk 3DS Max para hacer uso del formato .fbx que es compatible con el motor gráfico seleccionado.

3.3. Motor gráfico

Se hará uso del motor gráfico seleccionado para la creación del ambiente y la práctica a realizar, de esta manera a los modelos previamente renderizados se les otorgarán las propiedades necesarias para la integración e interacción con el ambiente, tales como: iluminación, colisión, materiales y texturas. Con estas propiedades y las herramientas del propio motor gráfico se podrá simular las físicas o programar el comportamiento deseado, como se observa en la Figura 22.

El motor gráfico seleccionado para la realización de este desarrollo es “Unreal Engine” de la empresa Epic Studios, ya que el uso de su licencia de software es gratuita con la opción de uso y venta hasta las 10,000 unidades, a partir de ese número se deberán pagar regalías a la empresa.

3.4. Sistema virtual

En el apartado del sistema virtual, es donde se situarán los modelos importados para la creación del ambiente virtual tanto del aula como de la subestación y el diseño de la interfaz gráfica con los pasos a seguir, así como la programación requerida para hacer uso de la realidad virtual dentro del motor gráfico. Como se puede observar en la Figura 23, es el diagrama a bloques del procedimiento realizado para alcanzar el desarrollo del caso propuesto, donde en primera instancia se hará uso de la plantilla básica de VR que incluye el motor gráfico e importamos los “assets” o modelos 3D para empezar a

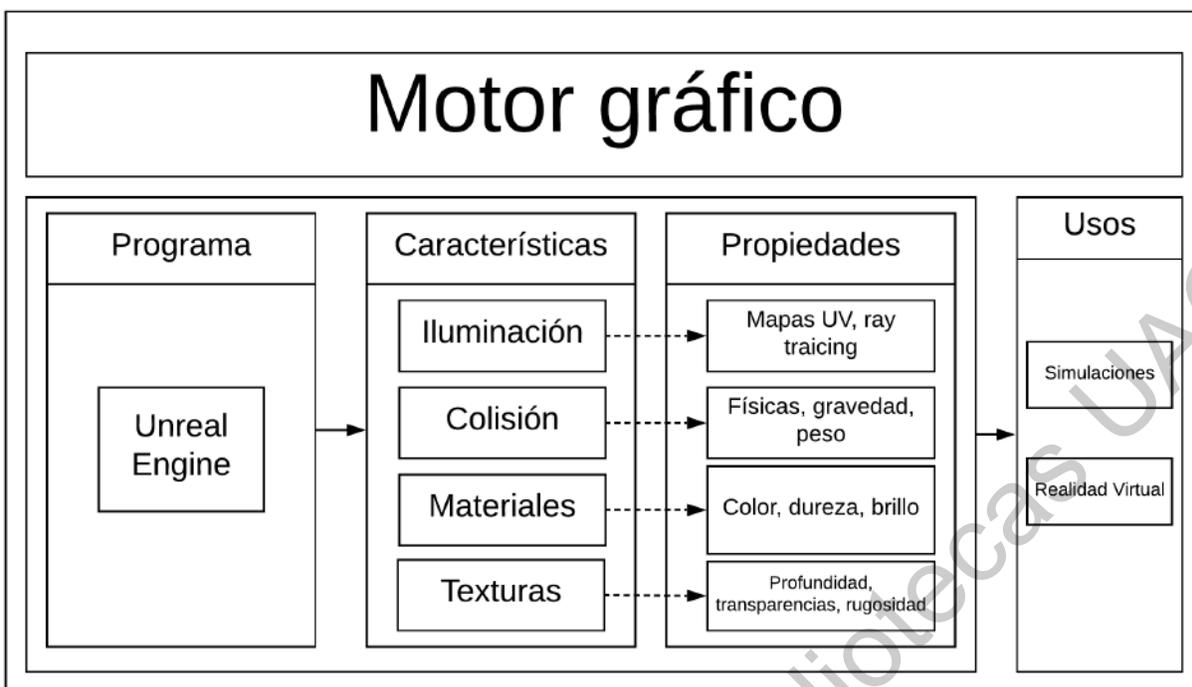


Figura 22: Diagrama a bloques del motor gráfico (elaboración propia).

armar los entornos virtuales, además se comprobará si los modelos contienen las propiedades mínimas para el funcionamiento dentro de la simulación de la realidad virtual, si no es así se le asignarán para su pleno funcionamiento. Ahora que los ambientes están construidos y los modelos están situados, se asignará a los modelos las propiedades de Actor dentro del motor gráfico creando su blueprint, que es como su huella que identifica y se le asigna programación específica a este objeto, de esta manera le proporcionaremos las propiedades necesarias para la interacción con los controles de movimiento para la interacción usuario - objeto, en vez de objeto - objeto, después se hará la programación de la secuencia de pasos que necesita el caso propuesto y crear la interfaz gráfica con la que el usuario interactuará para que obtenga la información para proseguir en la prueba.

3.5. Demostración

En esta sección el usuario podrá probar el software haciendo uso del sistema de realidad virtual pudiendo realizar la prueba en un ambiente altamente inmersivo y realista. En la Figura 24, se observa que el usuario con una computadora y el sistema de realidad virtual puede tener la experiencia completa enfocando su uso en los estudiantes, técnicos e ingenieros.

Consideraciones para el experimento con humanos

Se le entregará un formato de consentimiento informado al trabajador, que al ser firmado se le asignará un expediente como usuario y se le ubicará en una zona segura y cómoda donde se realizarán las pruebas y se le darán algunas indicaciones de seguridad y salud al usar los sistemas de realidad virtual (ver Tabla 3.5). Se tomara como referencia el sistema existente y se pondrá una persona que haya practicado con el software existente y se pondrá a practicar con el desarrollado en este proyecto, al final se repetirán los pasos en un modo de evaluación donde ya no se dará a detalle los métodos de

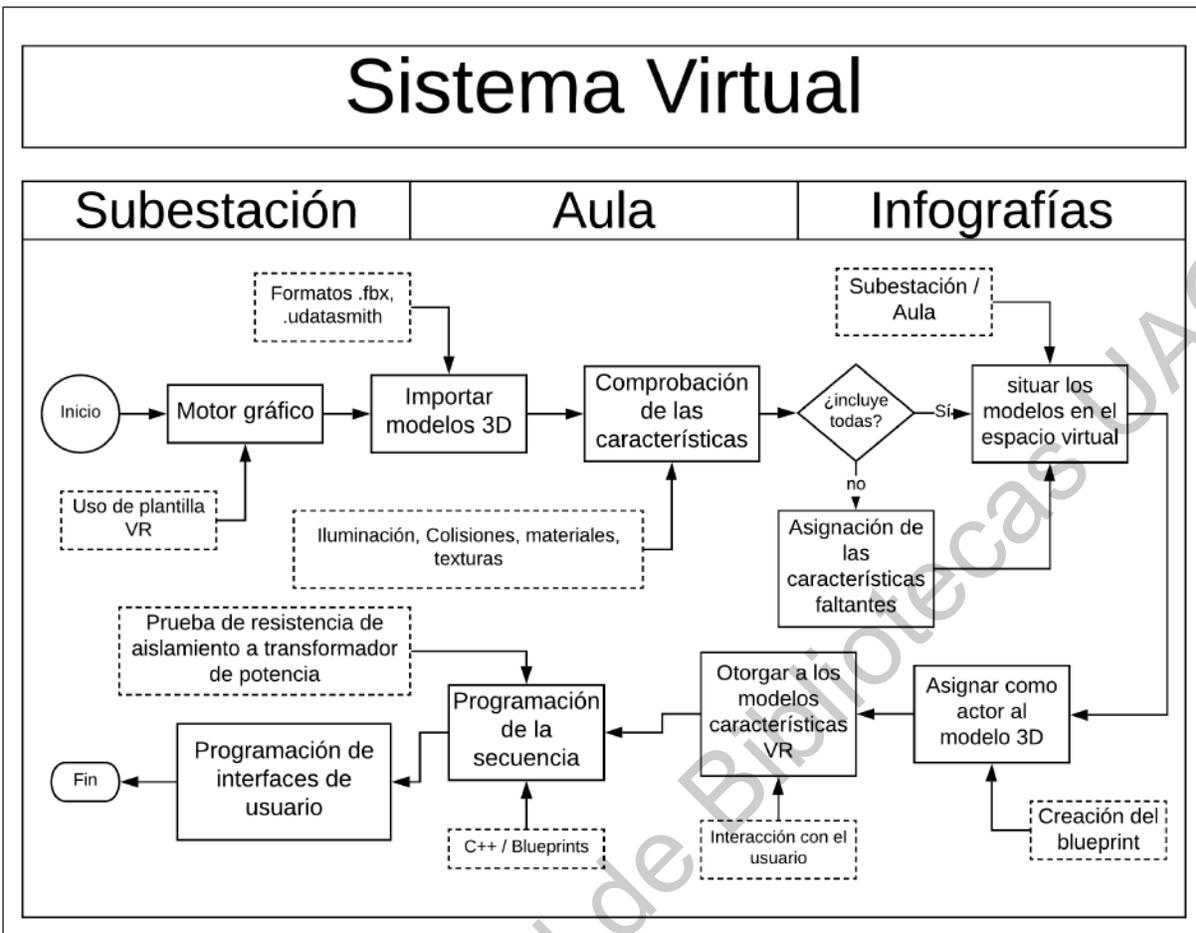


Figura 23: Diagrama a bloques del sistema virtual (elaboración propia).

seguridad ni las herramientas a utilizar sino el operario tendrá que hacerlo sin ninguna ayuda los pasos a seguir y de ahí se podrá medir la destreza adquirida.

Medidas a considerar dentro de los reglamentos de un comité de ética

Se considera el uso de una carta de consentimiento informado para los trabajadores para medir sus capacidades y otra carta de confidencialidad para el buen uso de la información que pueda ser otorgada por parte de la empresa CFE, evitando así posibles mal entendidos en caso de verse amenazada tanto física o moralmente. Cabe destacar que se cuenta con un contacto directo con la empresa CFE aportando la información necesaria del trabajo en campo para que de esta manera se realicen pruebas lo mas fiel posible a la realidad, que sirvan de modelo para una capacitación de manera segura evitando así posibles futuros accidentes. Debido a que los resultados obtenidos con el desarrollo del simulador le interesan mucho a CFE, se realizará un formato de consentimiento informado, donde los trabajadores den su permiso para poder aplicarles las pruebas y compartir sus resultados con la empresa.

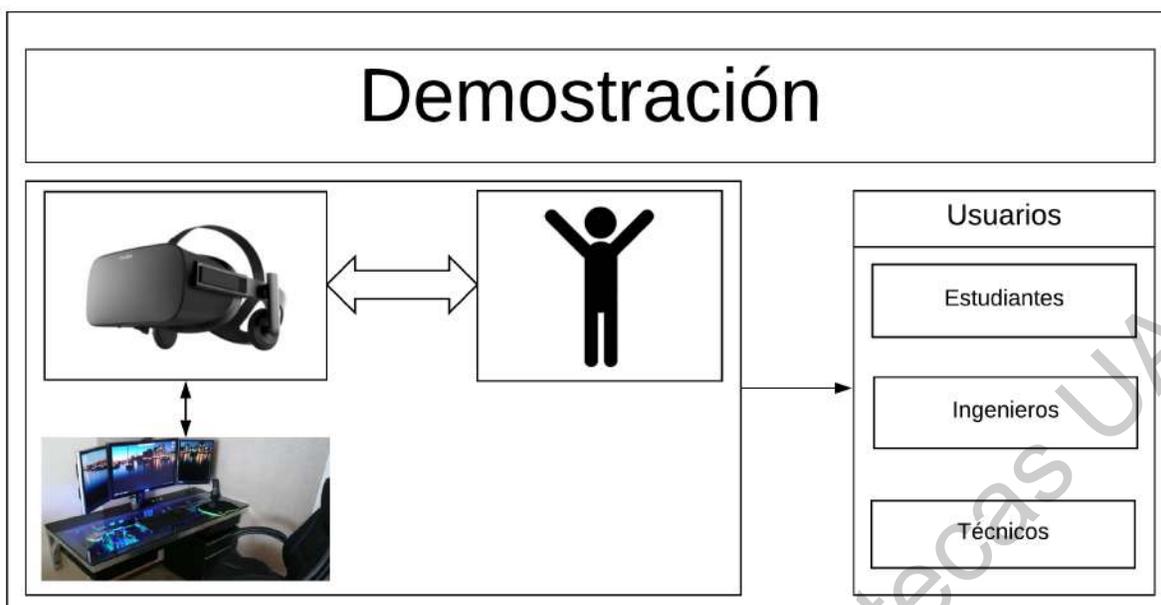


Figura 24: Diagrama a bloques de la demostración (elaboración propia).

Tabla 1: Recomendaciones al usar RV (Oculus, 2018).

Advertencias de seguridad y salud al usar dispositivos de realidad virtual
1. No deberá utilizar el Gear VR cuando esté cansado/a o falto/a de sueño, bajo la influencia de alcohol o drogas, con resaca, con problemas de digestión, bajo estrés emocional o en estado de ansiedad o cuando padezca resfriado, gripe, dolor de cabeza o dolor de oído, ya que puede contribuir a su susceptibilidad a los síntomas adversos.
2. Para una experiencia lo más segura posible, permanezca sentado/a o parado mientras utilice el Gear VR.
3. Asegúrese especialmente que no esté cerca de otras personas, objetos, escaleras, balcones, puertas o ventanas abiertas, muebles, ventiladores de techo, interruptores de luz u otros objetos con los que podría topar o que podría tirar al suelo mientras utilice el Gear VR o justo después.
4. Quite del área todo aquello con lo que pueda tropezar antes de utilizar el Gear VR.
5. Algunas personas (1 de cada 4.000) puede reaccionar con mareos, convulsiones, ataques epilépticos o pérdidas de conocimiento provocados por los flashes o patrones de luz y puede ocurrir al ver la televisión, jugar a videojuegos o experimentar la realidad virtual, incluso si nunca han sufrido convulsiones ni pérdidas de conocimiento ni tengan historial de convulsiones o epilepsia. Dichas convulsiones son más frecuentes en niños y jóvenes por debajo de la edad de 20 años. Cualquier potencial usuario que haya sufrido convulsiones, pérdida de conocimiento u otros síntomas relacionados con una condición epiléptica debería visitar al médico antes de utilizar el Gear VR.
6. Haga un descanso de 10 a 15 minutos cada media hora, aún en caso de que no sienta la necesidad. Cada persona es diferente, así que haga más frecuentes o más largos los descansos en caso de sentir malestar.

4. Resultados

4.1. Caso de estudio

Se seleccionó el caso de estudio de una prueba de resistencia de aislamiento a transformador de potencia, analizando la prueba y tomando nota de los pasos necesarios que se deben emular para el desarrollo en realidad virtual, estos pasos son:

Paso 1 Se cortocircuitan las boquillas de baja tensión X.

Paso 2 Se cortocircuitan las boquillas de alta tensión H.

Paso 3 Se conecta la terminal roja en las boquillas de alta tensión H1.

Paso 4 Se conecta la terminal negra en las boquillas de baja tensión X0.

Paso 5 Se conecta un puente auxiliar de las boquillas de baja(X0) a tierra del transformador.

Paso 6 Se inicia la prueba.

4.2. Diseño de elementos 3D

Con el caso seleccionado, se procede a modelar los objetos necesarios para crear el ambiente virtual tanto del aula como de la subestación, como se puede observar en la Figura 25 correspondiente a la subestación, que ya cuenta con los transformadores, la estructura metálica, cuchillas, contactores, aisladores y buses. Los modelos realizados se listan en la tabla 2 y 3 y serán exportados al motor gráfico con los formatos .fbx y .udatasmith.

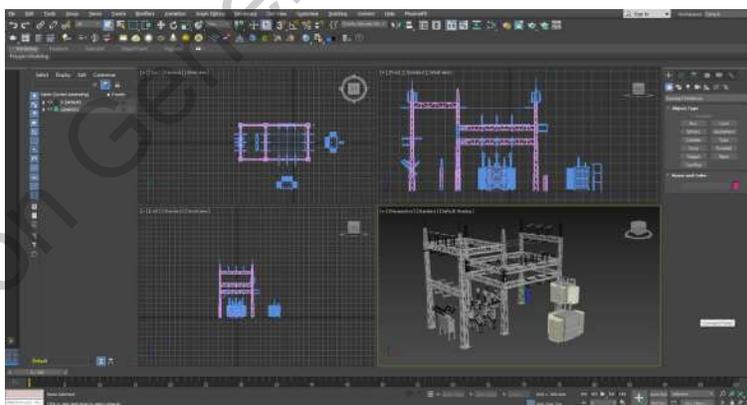


Figura 25: Diseño de los modelos 3D de la subestación en el software de diseño Autodesk 3ds max (elaboración propia).

4.3. Motor gráfico

El motor gráfico “Unreal Engine”, es un programa bastante versátil, que permite la importación de varios formatos compatibles con los programas de edición 3D, los usados en este desarrollo son los .fbx y los .udatasmith y como se observa en la Figura 26 se muestra el menú de importación que nos

Tabla 2: Modelos 3D de herramienta y equipo (elaboración propia).

Modelos 3D Herramienta y Equipo		
Casco	Segueta	Conos delimitadores
Cubeta de lona	Guantes	Desarmador de cruz
Juego de llaves mixtas	Juego de dados	Calculadora
Pinzas caimán	Camioneta	Desarmador plano
Detector de voltaje	Megger	Equipo de radio
Escalera de 16 peldaños	Escalera tipo tijera	gafas protectoras
Llave perico	Matraca de media	Pinzas de presión
Pinzas de electricista	Sombrilla	Tornillo de banco

Tabla 3: Modelos 3D del Aula y la Subestación (elaboración propia).

Modelos 3D Aula y Subestación		
Infraestructura salón	Mesas	Farolas
Marcos	Predio subestación	Paredes subestación
Contactores	Transformadores	estructura metálica
Aisladores	Buses	Cuchillas
Lineas aéreas	Zona subestación	Malla de tierras

despliega “Unreal Engine” para seleccionar las características con las cuales queremos trabajar dentro del programa, como se observa en la Figura 26 esta marcada la opción de “Auto generate collision” con lo cual el software asignará una colisión con una forma primitiva, ya sea una de tipo caja, esfera o de capsula si es que el modelo no cuenta con ella. También en la misma Figura 26 se observa que el objeto importado no cuenta con material por lo tanto se tendrá que asignarle uno o crearlo con la herramienta que posee “Unreal Engine”.

Ahora que los modelos han sido importados, el siguiente paso es abrir la herramienta de malla estática (Figura 27), que es una ventana donde se observan las características del objeto tal como la iluminación, colisión, materiales y la edición básica de la malla.

Con esta ventana abierta se procede a realizar la comprobación de que los modelos contengan las características necesarias, como se ve en la Figura 28 es un modelo de un casco que no cuenta con un material asignado ni una malla de colisión, por lo cual el objeto no tendrá interacción con el usuario ni con los demás objetos.

Con este motivo, se procede a asignarle un tipo de colisión como se ve en la Figura 29, donde se seleccionará la malla de colisión más adecuada y su resolución de puntos de la malla. Cuando se requiere que la malla de colisión sea lo más ajustada para otorgarle unos límites más cercanos al objeto y tener una mejor interacción, se recomienda el uso de la herramienta en el menú desplegable “Auto Convex Collision” que hace uso de un algoritmo basado en el principio de la envoltura, esto es que el software intentará hacer unas iteraciones tratando de envolver al objeto y otorgarle esa malla de colisión. En la Figura 30 se nota como al casco lo envuelve unas líneas de color naranja formando una malla de colisión generada con la herramienta “Auto Convex Collision”.

Otra de las características que se encuentran en esta herramienta son los mapas de luz, que son los que le indican a la luz que simula el motor gráfico la manera en la que debe rebotar y crear las zonas de luz y sombra, tal como la Figura 31 se denota el cuadro inferior izquierda como es el mapa de luz del objeto e inclusive con esta herramienta se puede editar aunque de manera muy básica, aunque en

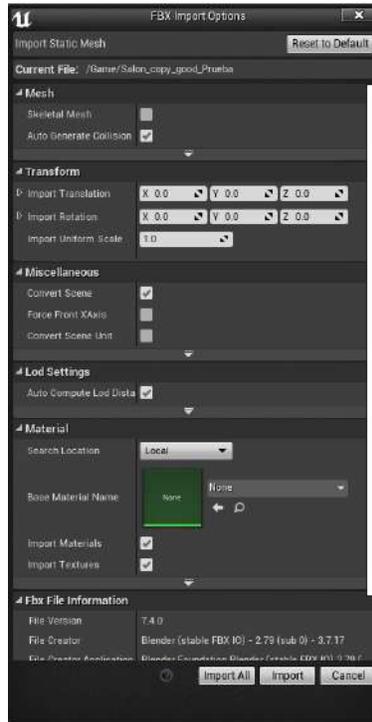


Figura 26: Menú de importación del motor gráfico “Unreal Engine” (elaboración propia).

versiones más actuales del motor gráfico inclusive ya se puede trabajar con la técnica de RayTracing dando aún mas realismo a los entornos creados.

Ahora que ya se cuenta con esas características, se le asignará los materiales ya sea que vengan desde el programa de edición 3D o usar la herramienta de creación de “Unreal Engine” como se ve en la Figura 32. Con esta herramienta, del lado izquierdo se observa el material ya renderizado y en tiempo real de las modificaciones que se vayan realizando en los nodos de programación. Estos nodos vienen del blueprint del material donde a través de un nodo que contiene muchas opciones de entrada, se puede agregar pequeños nodos de información para la combinación y creación del material. Ahora se entiende que un material se compone de un color base, un vector de normales, la rugosidad si el material lo ocupa y otorgarle una apariencia metálica, estos pequeños valores se conocen como texturas que son como el corazón de un material creado y de aquí se pueden hacer muchísimas variaciones acorde a las necesidades.

Por ultimo en la Figura 33 se puede observar los modelos ya importados y con sus características ya asignadas así como sus materiales.

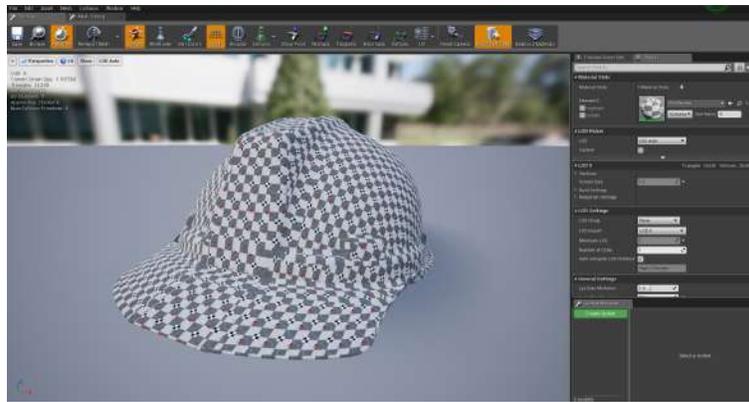


Figura 27: Herramienta de “Unreal Engine” para las mallas estáticas (elaboración propia).



Figura 28: Modelo 3D de un casco de protección sin material ni malla de colisión (elaboración propia).

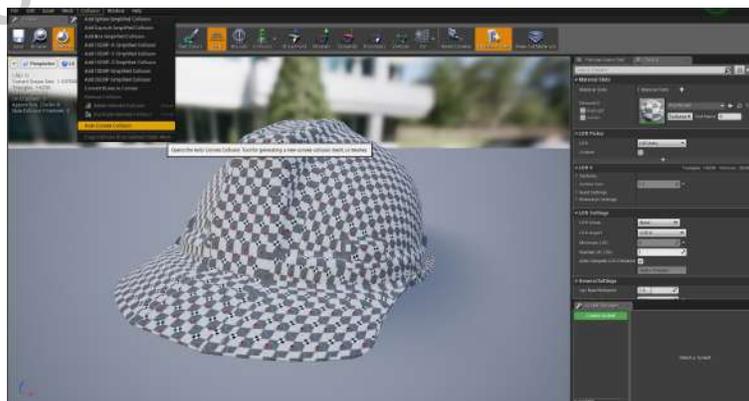


Figura 29: Menú cascada del botón de los tipos de colisión (elaboración propia).

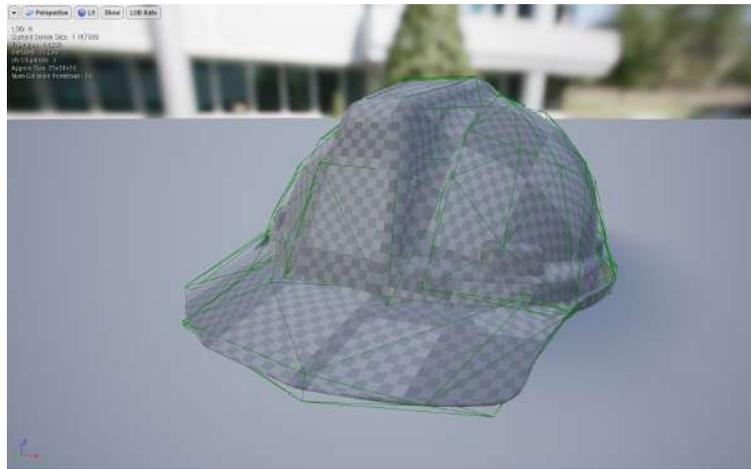


Figura 30: Modelo 3D de un casco de protección sin material con malla de colisión (elaboración propia).

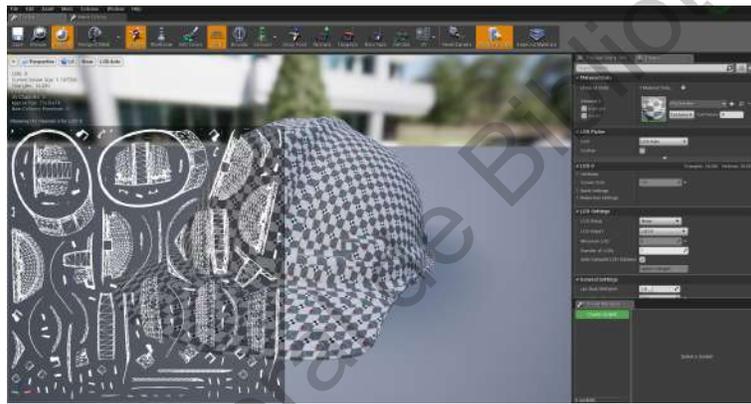


Figura 31: Mapa de Luz del modelo 3D del casco (elaboración propia).

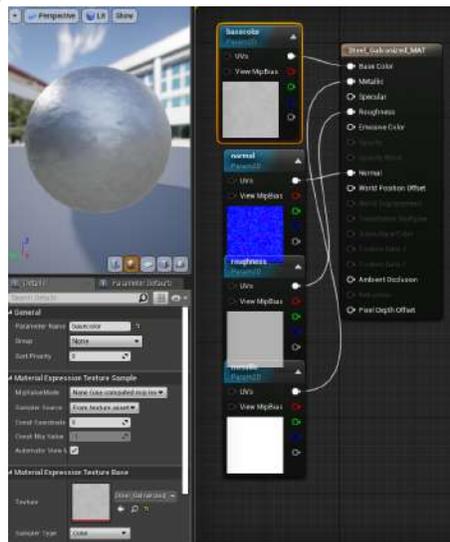


Figura 32: Herramienta de creación de materiales (elaboración propia).



Figura 33: Diseño final de los modelos 3D del aula y su importación al motor gráfico “Unreal Engine” (elaboración propia).

4.4. Sistema virtual

4.4.1. Motor gráfico, uso de plantilla VR

Al momento de abrir el programa del motor gráfico, se debe crear un nuevo proyecto de realidad virtual para hacer uso de la plantilla VR que viene dentro del mismo, en la Figura 34 se presenta el menú de “Unreal Engine” para seleccionar el tipo de proyecto que se desea crear y no empezar desde cero, facilitando el acceso a las herramientas y los blueprints de programación necesarios para empezar el tipo de proyecto seleccionado (Figura 35), dentro de cada archivo de blueprint se encuentran los nodos de programación que le dan vida al sistema de realidad virtual y posibilita su interacción con el ambiente.

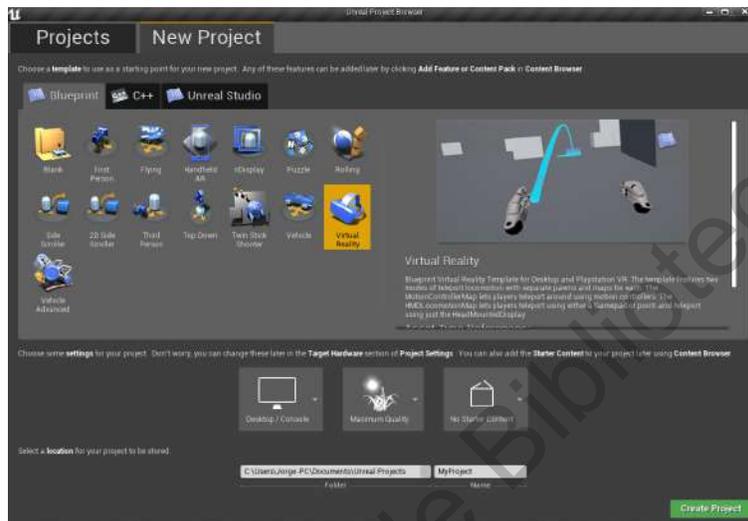


Figura 34: Menú de “Unreal Engine” de selección de plantillas para nuevos proyectos (elaboración propia).



Figura 35: Blueprints incluidos en la plantilla VR de “Unreal Engine” (elaboración propia).

Empezando con el blueprint de la Figura 35 llamado “BP_ MotionController”, es el encargado principalmente de la interacción y control de las manos virtuales por medio del dispositivo de control del sistema virtual, tal como, ocultar el cilindro de teletransportación hasta que lo active el usuario, la vibración del control como retroalimentación cuando se toma un objeto válido, la animación de la mano virtual y su actualización dependiendo de lo que haga el usuario tanto como si abre la mano o la cierra al tomar un objeto o dejarlo, también se encarga de manejar el arco de teletransportación cuando el usuario desee transportarse. Este blueprint esta meramente enfocado al usuario y se podría decir que es código que corre de manera invisible al usuario final.

El siguiente blueprint llamado “BP_ PickupCube” es el actor ejemplo base para conocer como convertir tus modelos 3D en actores con propiedades de interacción de realidad virtual, esto es poder tomar y soltar los objetos con las manos y que además en la simulación se siga calculando las físicas realistas de como se comportaría el objeto debido a la perturbación que aplicamos al sistema.

Continuando con el siguiente blueprint llamado “HMDLocomotionPawn”, se encarga de realizar principalmente el movimiento del usuario, esto es, que el personaje pueda de estar en un punto y teletransportarse a otro punto obtenido del blueprint “BP_MotionController”, es decir que trabajan juntos para obtener los puntos y si son válidos ejecutar la acción de teletransportación. Otra acción que realiza es determinar la altura de la persona que le asignará a la cámara virtual para que el usuario experimente tener la misma altura tanto fuera como dentro del software.

El penúltimo blueprint llamado “MotionControllerPawn”, es el blueprint que se encarga de generar el avatar que el usuario va a manejar, esto quiere decir que es el blueprint que obtiene los datos de los demás blueprint y los hace converger para aplicarlos en el uso que quiera otorgar el usuario. Las funciones que realiza este blueprint haciendo uso de los datos obtenidos son: Determinar la altura del usuario y asignarlo dependiendo del sistema virtual que se este usando y poder asignarle un origen al avatar virtual, generar los controles y acoplarlos a los controles de movimiento del sistema virtual, maneja la teletransportación, maneja la recepción de acción de entrada de los controles de movimiento para determinar que botones a presionado el usuario y también hace la función de determinar la rotación de cada mano para poder trasladar y rotar las manos virtuales y tener mayor inmersión al sentir que se estan controlando las propias manos del usuario dentro del software.

El ultimo archivo llamado “PickupActorInterface” es tan solo una interfaz que al asignárselo a un actor permite el uso de los nodos de evento “Event pickUp” y “Event Drop” que son llamados por el blueprint “BP_MotionController” para generar la acción de tomar y soltar, indicandole al sistema que son objetos validos para la interacción.

Estos blueprints son únicamente para la programación interacción usuario - objeto, debido que Unreal también provee unas carpetas con las animaciones necesarias de la mano, así como el maniquí que representa cada articulación de la mano llamado por el sistema como un “Skeleton mesh”, donde el diseñador podría cambiar el aspecto de las manos a una mas acorde a la prueba, por ejemplo unos guantes de carnaza.

4.4.2. Importar modelos 3D - Comprobación de Características

Como ya contamos con la plantilla de realidad virtual, se pueden importar los modelos 3D que ya se tienen diseñados de la manera que se menciona en la sección 4.3 usando la herramienta que se ve en la Figura 26 y se deberá comprobar que ya cuentan con las características de iluminación, colisión, materiales y texturas de la misma forma que se cita en la sección 4.3, si no es así debemos hacer uso de la herramienta de malla estática para asignárselos.

4.4.3. Situar modelos en el espacio virtual

Ahora lo que procede es empezar a acomodar los modelos con la ayuda de los “ejes” que incluye el motor gráfico para la traslación, rotación y la escala, de tal manera que se vaya generando el ambiente deseado, como se puede ver en la Figura 36 es donde apenas se comenzaba con el desarrollo del ambiente de la subestación ya que faltaban texturas, materiales, diseño del cielo, el suelo, etc., en cambio en la Figura 37 se observa la versión final y la versión final del aula se puede observar en la Figura 38.

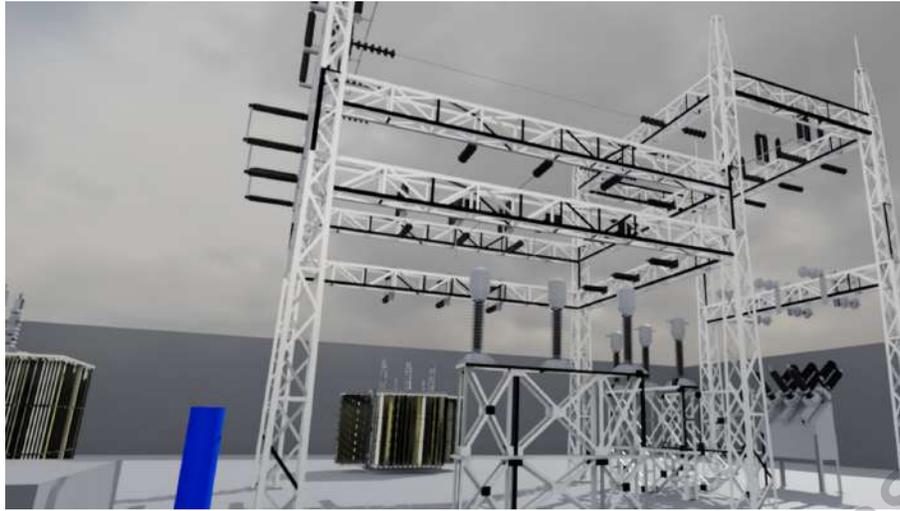


Figura 36: Versión alpha del ambiente de la Subestación (elaboración propia).



Figura 37: Versión final del ambiente de la Subestación (elaboración propia).

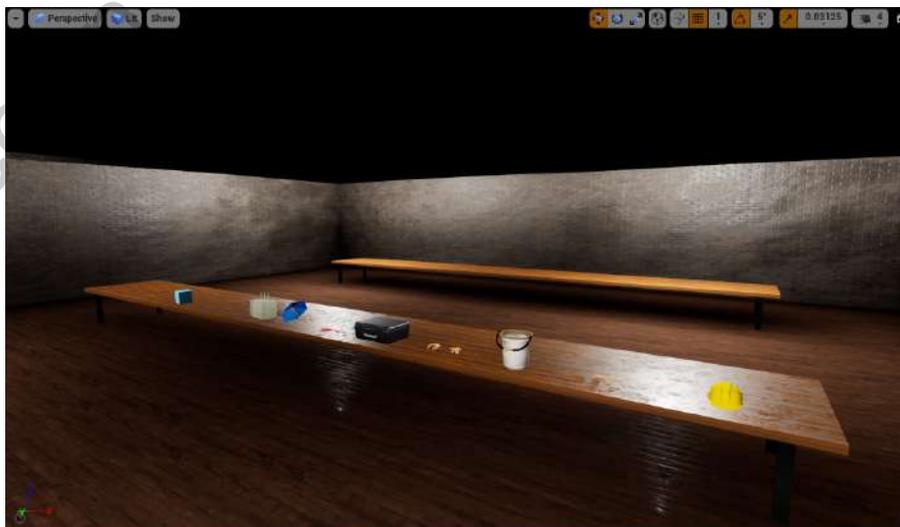


Figura 38: Versión final del ambiente del Aula (elaboración propia).

4.4.4. Asignación como actor al modelo 3D

Finalmente con los ambientes terminados, se debe prestar atención al actor base que Unreal proporciona para comprender como convertir los modelos 3D en actores, al dar doble clic en el blueprint “BP_PickupCube”, se abre el editor de blueprints (Figura 39), el cual cuenta con 3 secciones principales, la primera es el “Viewport” donde esta lo relacionado al modelo 3D, del lado izquierdo se pueden agregar componentes que son módulos que se pueden anexar a tu actor, por ejemplo otras mallas estáticas como modelos 3D, sonidos, partículas e inteligencia artificial por mencionar algunas. Del lado derecho tenemos las propiedades particulares de cada componente que constituye al actor, en la Figura 39 se observan las que constituyen al mallas estáticas como sus componentes de transformación, el modelo que compone la malla estática, los materiales si es más de uno, las físicas que tiene que ver con la gravedad, la masa del objeto, su fricción, su respuesta a la colisión, la manera en como se comporta a la luz y lo detallado de su mapa de luz además de asignarle un disparador de evento, por ejemplo una determinada llave que abra una puerta.

La segunda sección llamada “Construction script” esta responsabilizada de crear al actor, al momento de correr el software es el primer bloque de programación que se ejecuta y comprueba que los nodos de programación de la tercera sección llamada “Event Graph” no contenga errores.

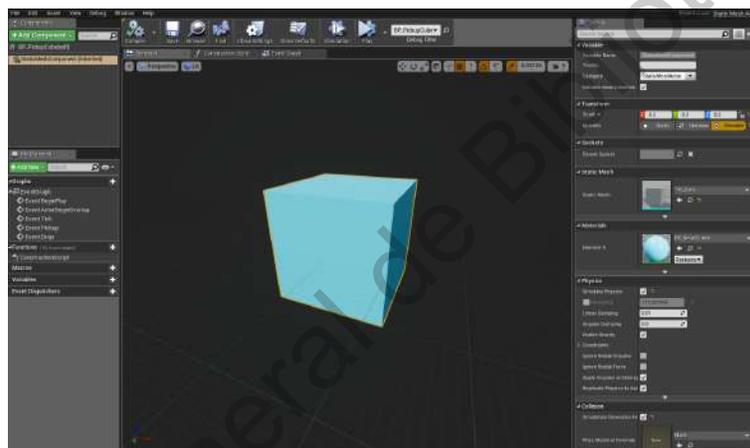


Figura 39: Editor de blueprint de “Unreal Engine” (elaboración propia).

Como se mencionó, en el “Event Graph” es donde se coloca la programación del actor para darle vida, cuenta con tres nodos por defecto como se ve en la Figura 40, estos son “Event BeginPlay” es el nodo que se ejecuta después del “Construction script” y se ejecuta una sola vez cada que se corra el software, esto quiere decir que es un nodo de una sola acción, por ejemplo, que el avatar cuando haga una acción específica a este actor se destruya o realice rotaciones como las manecillas de reloj.

El nodo “Event ActorBeginOverlap” es un nodo que siempre esta alerta en espera de su detonador, como su nombre lo dice ActorBeginOverlap su traducción sería cuando inicie el traslape del actor, por ejemplo, si se lanza una pelota a la diana y se requiere saber que puntaje obtuvo, se realizaría con este nodo debido a que en el momento en que la pelota toque al actor de la diana detonara el valor de la variable asignado a esa parte de la diana, pudiendo imprimir en pantalla el valor obtenido.

El ultimo nodo denominado “Event Tick” este nodo tiene una función particular y esta ligado a los FPS (Frame per second) que tu monitor maneja, si tu monitor es de 60 hz tendrás 60 cuadros por segundo, entonces el nodo “Event Tick” se ejecutará 60 veces en un segundo a diferencia del nodo “Event BeginPlay” que solo se ejecuta una vez. Un ejemplo del nodo “Event Tick” podría ser cuando se requiere saber cuanta vida útil le queda a una herramienta antes de romperse por el uso, ya que se estará refrescando la variable asignada a la vida de la herramienta y al llegar a cero se destruirá.

En este proyecto no se tienen actores que deban ejecutarse al momento de reproducir el software,

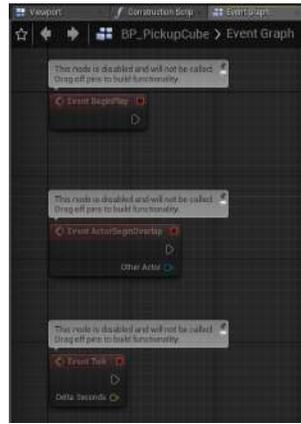


Figura 40: Editor de blueprints sección Event Graph (elaboración propia).

por esa razón en la Figura 40 tiene los nodos desactivados y tienen esa apariencia gris, si se llegarán a necesitar tan solo hay que activarlos arrastrando el “Exec” del nodo volviéndose de color rojo.

En la Figura 41 se contempla el código necesario para que los modelos convertidos en actores funcionen y obtengan las propiedades de realidad virtual para su interacción y manipulación, empezando con el nodo “Event Pickup” el cual esta a la espera de la indicación del blueprint “BP_ MotionController” el cual le comunica que el usuario cerro la mano virtual y se ejecuta el nodo “Event Pickup” si esta traslapado el actor y la mano virtual, este nodo también obtiene la referencia del actor o del objeto que se sobrepone para su uso posterior y lo guarda mientras el nodo este activo.

Siguiendo el “Exec” sigue el nodo “Set Simulate Physics” que como su nombre lo indica es un nodo que asigna físicas simuladas y mediante dos inputs o entradas se hace referencia a ellos, por ejemplo en el input target se le asigna la referencia de la malla estática que se le simulará o no las físicas y esto depende del valor booleano del segundo input llamado simulate, en este caso esta negado para desactivar las físicas del objeto tomado y poder manipular el actor.

Terminando con el ultimo nodo de la sección de tomar un objeto, esta el nodo de “attachToComponent” que es un nodo encargado de unir o acoplar dos objetos, en este caso con el input target se quiere que el actor tomado se acople a la mano virtual que hizo el gesto de tomar, entonces se debe obtener la raíz del componente del actor tomado y acoplarlo al input Parent que se obtiene su referencia desde el nodo ‘Event Pickup’.

Hasta este momento, si se dejara hasta aquí el código, el actor tomado se quedaría todo el tiempo en la mano virtual aunque el usuario haya soltado los botones de agarre, por esta razón se debe usar el nodo “Event Drop” que es lo contrario del nodo “Event Pickup”. Después de activarse el nodo “Event Drop”, siguiendo el “exec” esta el nodo “Set Simulate Physics” donde en el input target se vuelve a usar la referencia de la malla estática para indicarle mediante el input booleano en verdadero para que al soltar al actor vuelva a tener las físicas simuladas y como último paso esta el nodo “DetachFromActor” que su función es desacoplar el actor de las manos virtuales.

Ahora que ya se analizó el código básico para los actores en realidad virtual, se procede a la creación de los blueprint donde los modelos 3D se convertirán en actores, para esto hay dos maneras, una es la creación del blueprint desde cero y la segunda es hacer uso de la plantilla de blueprint “BP_PickupCube”. Se explicará la primera manera, en el “content browser” es el área donde se exploran, modifican o se crean los archivos que conforman el proyecto, de tal manera que al dar clic secundario se despliega un menú (Figura 42) y con el se puede crear carpetas, nuevas clases para la programación en C++, importar “assets” en la carpeta donde se dio clic derecho, crear las clases de blueprint, crear niveles nuevos, crear materiales e inclusive agregar un sistema de partículas y varios apartados con plantillas para la creación de avanzados “assets”.

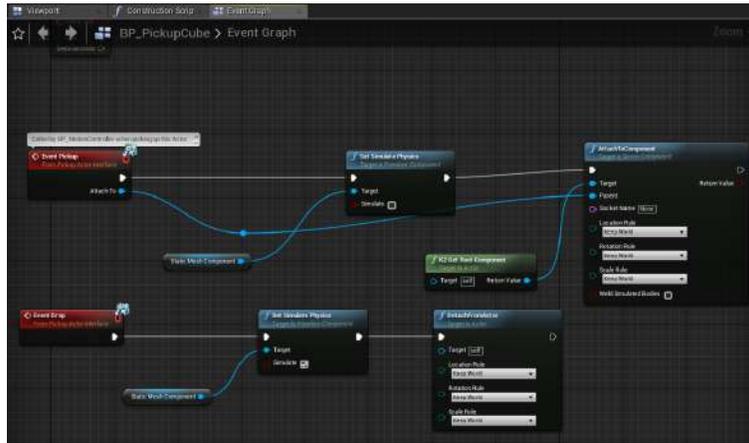


Figura 41: Código para habilitar al actor como objeto de interacción de realidad virtual (elaboración propia).



Figura 42: Menú para la creación de varios tipos “assets” o blueprints (elaboración propia).

El apartado que se necesita es el “blueprint class” que al darle clic se abrirá una ventana con las muchas variedades que se pueden crear de clases de blueprint, en la Figura 43 se puede observar las categorías englobadas en los blueprints, empezando con el actor que es un objeto que se puede situar o que se genere en el mundo, el peón o pawn en ingles es un actor que se puede poseer y puede recibir el comando de entrada de un mando controlador, el carácter es un tipo de peón que tiene la habilidad de caminar o desplazarse por el mundo, el controlador del jugador es un actor responsable de controlar al peón usado por el jugador, el modo base de juego define como sera jugado el juego, son las reglas, el puntaje y cualquier faceta que tenga el tipo de juego. Estos serian los más comunes, pero el que se necesita es un tipo de blueprint llamado “Static Mesh Actor”, por lo cual se dará clic al triángulo donde dice “All classes” y en su buscador se pondrá “StaticMesh” y se seleccionará el que se ubica en la sección de “Actor” llamado “StaticMeshActor” como se ve en la Figura 43, lo seleccionamos y le asignamos el nombre que se quiera.

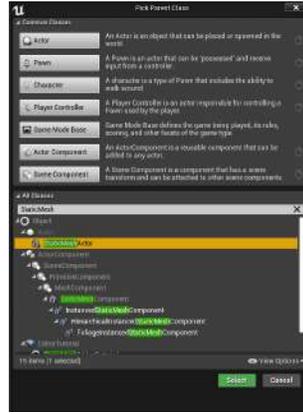


Figura 43: Menú para la creación de clases de blueprints (elaboración propia).

4.4.5. Otorgar a los modelos características VR

Se le da doble clic al recién creado blueprint para abrir su editor que esta complemente en blanco, para empezar a programar es necesario agregar la interfaz necesaria llamada “PickupActorInterface” explicada anteriormente, de esta manera en el menu superior del “Viewport” en la sección “Class Settings” en el panel de detalles a la derecha, en la sección “interfaces” se da clic en “add” y se busca la interface “PickupActorInterface” como se ve en la Figura 44, con esta interface se podrán añadir los nodos necesarios para empezar a programar.

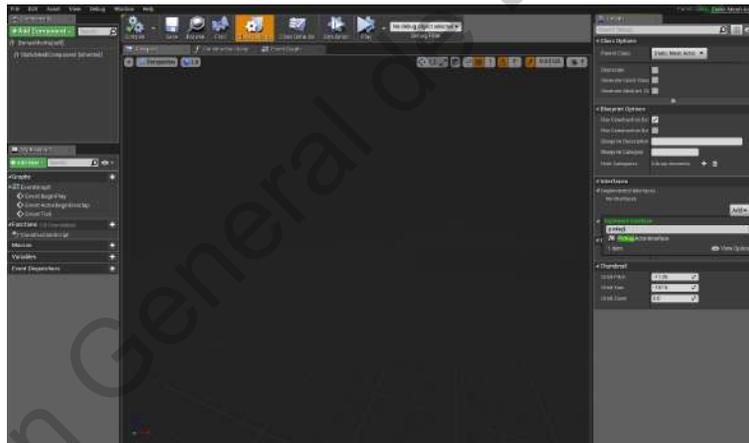


Figura 44: Editor de blueprint para añadir la interface de realidad virtual (elaboración propia).

Antes de empezar a programar, se añadirá el modelo 3D diseñado previamente para añadirlo al blueprint recién creado, para esto se selecciona el componente de “StaticMeshComponent(herited)” dentro de la sección de “Components” al lado izquierdo de la ventana, esto hará cambiar el panel de detalles y en la sección “Static Mesh” donde dice “None” se dará clic y se buscará el modelo que se desea asignar a este blueprint, como se observa en la Figura 45, donde a pesar que hay muchas palabras clave de transformador se debe seleccionar el que se designa como “Static Mesh” y ver la vista previa en el “Viewport”.

Ahora se pasa al “Event Graph” donde se empezará la programación, para añadir un nodo se tiene que dar clic secundario en el plano vacío para desplegar el menú donde se tiene que buscar el nodo que se requiera, como se aprecia en la Figura 46, de esta manera se irán agregando los nodos hasta que tengamos la misma configuración presentada en la plantilla del cubo de realidad virtual presentada por “Unreal Engine” y la Figura 41.

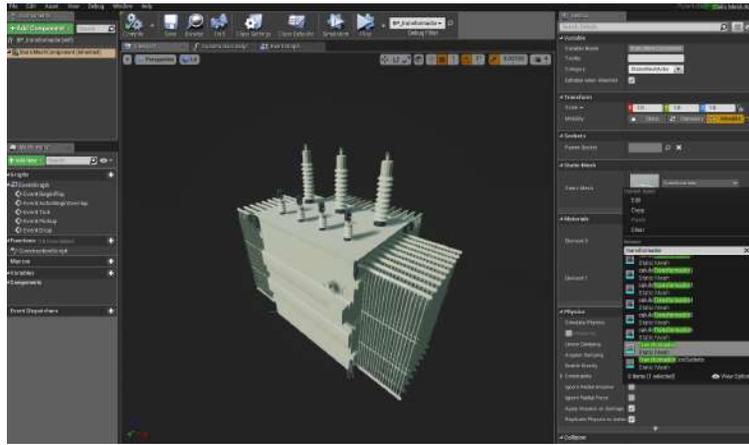


Figura 45: Editor de blueprint para asignar un Static Mesh (elaboración propia).

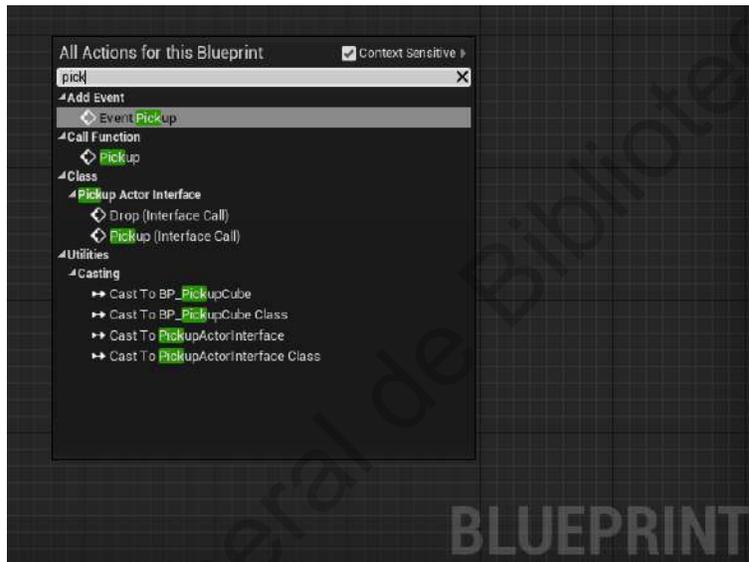


Figura 46: Menú para añadir nodos al blueprint (elaboración propia).

De esta manera se completa la primera opción de crear al actor de blueprint, la segunda manera que es la usada en este trabajo es sustituir el “Static Mesh” del cubo que viene en la plantilla y la sustituimos por el modelo deseado. Para empezar se debe copiar la plantilla “BP_PickupCube” a otra carpeta donde se quiera organizar los blueprints, se abre el editor con doble clic y se selecciona el “StaticMeshComponent(inherited)” en la sección de “components” y en el panel de detalles en la sección “Static Mesh” se da clic donde dice “1M_Cube” y se busca el modelo deseado para este blueprint, de esta manera se convierten los modelos que importamos en actores con propiedades para su interacción con realidad virtual.

4.4.6. Programación de la secuencia

El motor gráfico “Unreal Engine” es muy amigable al momento de programar ya que permite realizar proyectos de diversas índoles como la realidad virtual, empezando con el tema del Aula que ya se tiene el ambiente terminado (Figura 38) y esta zona que esta designada para la capacitación del usuario, se debe añadir la infografía necesaria a los actores para que al momento de sujetarlos se despliegue su respectiva información, por lo cual se retomará en la sección 4.4.7.

Haciendo uso de la maniobra mencionada en la sección 4.1 del Caso de estudio se procederá con la

programación, el método seleccionado es mediante un disparador de eventos el cual al momento de ser traspasado cargará el siguiente nivel.

El disparador de eventos es una geometría con forma de caja, estos disparadores de eventos se llaman “Box Trigger” (Figura 47) en ingles y como cualquier geometría dentro de “Unreal Engine” se puede modificar mediante los “ejes” para su traslación, rotación y escala.

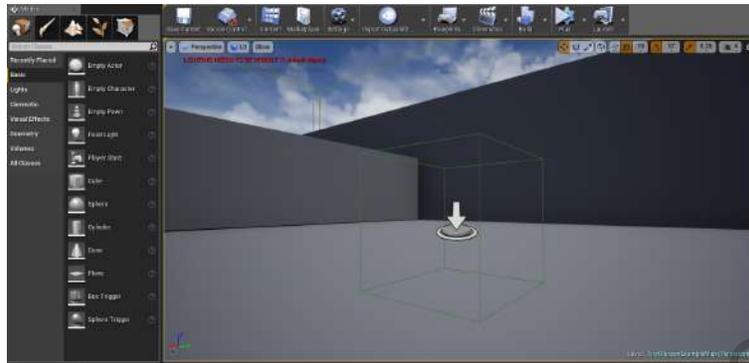


Figura 47: Caja disparador de eventos “Box Trigger” (elaboración propia).

Estos “Box Trigger” funcionan al ser implementados en el blueprint del nivel, esto quiere decir que es el blueprint de más alta jerarquía ya que es el que regirá sobre todo el nivel y todo lo que contenga, esto es, a los actores, modelos, luces, colisiones, etc. Para acceder al “Level Blueprint”, se debe dar clic izquierdo en la barra principal donde dice “Blueprints”, ahí se busca la sección “Level Blueprints” y se hace clic en “Open level Blueprints” como se ve en la Figura 48.

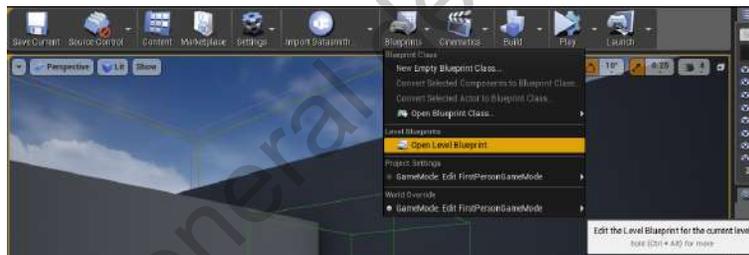


Figura 48: Localización del Level Blueprint (elaboración propia).

El “Level Blueprint” se programa de la misma manera que los blueprints normales, con la única diferencia que se puede hacer referencia a objetos y actores para hacer enlaces y que interaccionen entre si. Para hacer una referencia a un objeto o actor, se debe seleccionar el objeto en el “Viewport” o buscarlo desde el “World Outliner” y debe verse como la Figura 49.

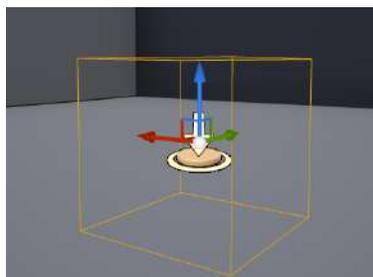


Figura 49: Box Trigger seleccionado para hacer referencia en el Level Blueprint (elaboración propia).

Con el “Level Blueprint” abierto se da clic secundario en la zona que tiene un grid y aparecerá un menú para añadir nodos (Figura 50), pero debido a que estamos en el blueprint del nivel se añaden

dos secciones más, estas son “Add Event for Trigger Box 1” y “Call Function on Trigger Box 1”, estas secciones toman el nombre del objeto que esta seleccionado por tal razón sale con el nombre “Trigger box 1”.

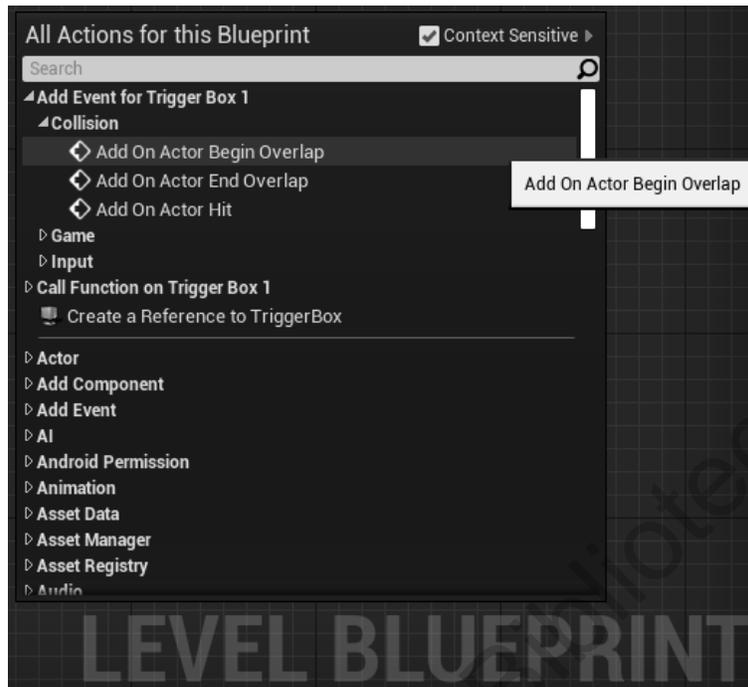


Figura 50: Menú para la creación de referencias a un actor o añadir tipos de eventos (elaboración propia).

En este menú de la Figura 50, se da clic en el triángulo de “Add Event for Trigger Box 1” para desplegar los tipos de eventos que se pueden agregar, los cuales son colisión, de juego y de entrada, el evento que interesa es el de colisión por lo cual se expande y se deberá dar clic en “Add On Actor Begin Overlap” y se añadirá el nodo del mismo nombre como se ve en la Figura 51.



Figura 51: Nodo de entrada para la detección de traslape para detonar eventos (elaboración propia).

Entonces con el “Box Trigger” y el uso del “level Blueprint” se logrará que el usuario progrese dentro de la prueba, por lo cual se procede a preparar la zona donde se realizará el primer paso. El primer paso consiste en cortocircuitar las boquillas de baja tensión X, por lo cual se necesita un transformador, un megger, 10 puentes auxiliares y una mesa donde colocar las herramientas quedando como se ve en la Figura 52.

Antes de agregar el salto de nivel con el “Box Trigger” primero se debe copiar el nivel actual, esto es para garantizar una continuidad y concordancia con los siguientes niveles, para hacerlo tan solo se busca el blueprint del mapa de nivel (Figura 53) que es el encargado de almacenar todo lo referente a los “assets” y objetos que se encuentren dentro. Para copiarlo tan solo hay que darle clic secundario y desplazarse hasta donde dice “Duplicate” o usar el atajo “Control + W” dentro del “Content Browser” y se le coloca un nombre para identificarlo.



Figura 52: Preparación del ambiente para el paso 1 de la prueba (elaboración propia).



Figura 53: Icono del blueprint de mapa del nivel (elaboración propia).

Agregamos el “Box Trigger” y se le coloca un nombre que ayude a localizarlo más rápido, después solo se debe trasladar y transformar con los “ejes” y llevarlo donde se necesita que en el primer paso es en las boquillas de baja tensión para empezar a cortocircuitarlo con los puentes auxiliares. En la Figura 54, se observa la posición y forma final de la primera “Box Trigger” ya que no importa por cual boquilla quiera empezar el usuario.



Figura 54: Colocación del trigger en su lugar para disparar el nivel 2 (elaboración propia).

Ahora con el “Box Trigger” seleccionado, se abre el “level Blueprint” donde se agregará la funcionalidad que al traslaparse con un objeto o actor, se cargue el siguiente nivel. En la Figura 55 se puede observar como quedan los nodos de programación para el “Box Trigger”.

El nodo “Open Level” es un nodo que puede ser modular, esto es, si un nivel tiene varias salidas,



Figura 55: Nodos de Programación para el “Box Trigger” y carga del siguiente nivel (elaboración propia).

se obtendrá la referencia de a qué nivel el usuario decidió desplazarse y se abrirá ese nivel mediante su referencia, la manera directa es si ya se cuenta con el nombre del nivel al cual el usuario debe acceder.

Para empezar con el segundo paso se debe copiar el nivel 1 y asignarle el nombre “Level2_Corto_X” y se abrirá el nuevo nivel dando doble clic sobre el, con el nuevo nivel abierto se añade un nuevo “Box Trigger”, se le asigna un nombre desplazándolo a las terminales de alta tensión y recordando el primer paso, ya deberían estar cortocircuitadas las boquillas de baja tensión en este nivel, se procede a mover mediante los “ejes” los puentes auxiliares quedando tal y como se ve en el la Figura 56, donde se aprecia el “Box Trigger” en su posición final y la boquilla de baja tensión en corto circuito.



Figura 56: Colocación del trigger para el nivel 3 y con las boquillas de baja tensión cortocircuitadas (elaboración propia).

Como se realizó con el primer nivel, se copiará el segundo y se nombrará nivel_3, ya que se tiene el nombre del nivel 3, se abre el “Level Blueprint” para añadir los nodos de programación (Figura 55) y se asigna el nombre del nivel que se debe abrir, en este caso “nivel_3”, Ahora se abre el nuevo nivel con doble clic y siguiendo el mismo método, se colocan los puentes auxiliares para dejar en corto circuito las boquillas de alta tensión (Figura 57) y se añade un nuevo “Box Trigger”. En este paso se solicita que la terminal roja del megger sea conectada a la boquilla H1 de alta tensión, así que esta boquilla será la posición final de la nueva “Box Trigger” como se ve en la figura 57, como se hizo anteriormente se duplica el nivel 3 y se nombra como nivel 4, con el nuevo nivel creado se abre el “Level Blueprint” del nivel 3 y debido a que se copio el nivel anterior los nodos de programación (Figura 55) ya se encuentran ahí, por lo cual tan solo hay que indicar el nuevo nombre “nivel_4”.

Repitiendo el mismo protocolo, se abre el “nivel_4” con doble clic y como se tienen los mismos elementos ya que fue duplicado el nivel 3, tan solo se debe mover el “Box Trigger” a su nueva posición que será la terminal X0 de baja tensión, ya que el paso 4 refiere que se debe conectar la terminal negra en la boquilla X0, se abre el “Level Blueprint” y en el nodo “Open Level” se le colocará el nombre tentativo de “nivel_5”, se cierra el “Level Blueprint” y se traslada la punta de caimán roja con los ejes

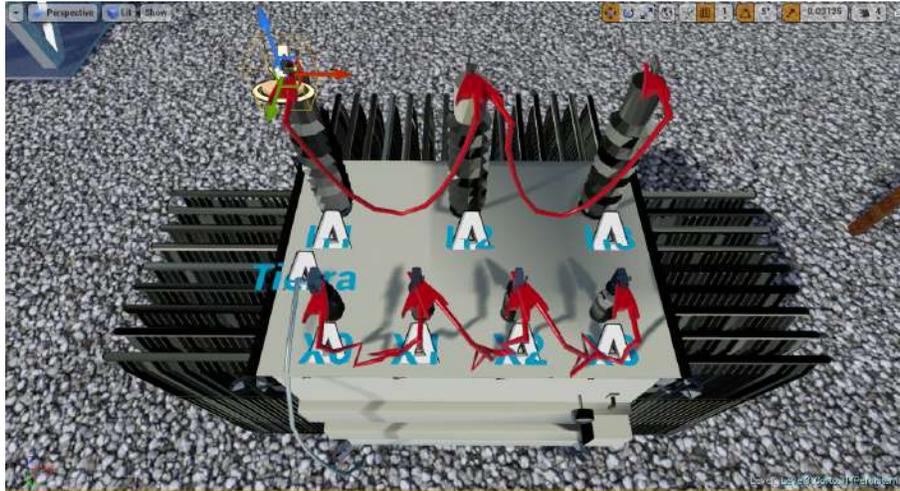


Figura 57: Colocación del trigger para el nivel 4 y con las boquillas de alta tensión cortocircuitadas (elaboración propia).

hacia la boquilla H1 de alta tensión, tal y como se observa en la Figura 58.

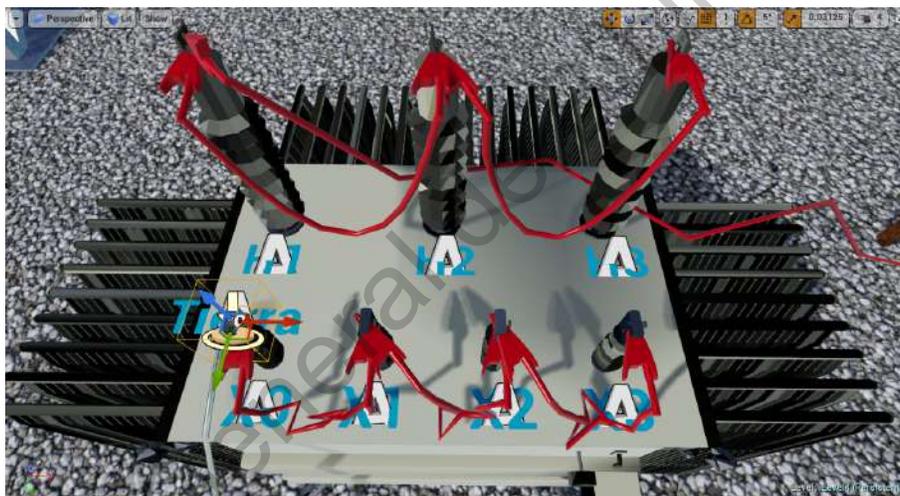


Figura 58: Colocación del trigger para el nivel 5 y conexión de la terminal roja del megger a H1 (elaboración propia).

Como penúltimo paso, se vuelve a copiar el “nivel_4” y se renombra el nuevo archivo a “nivel_5”, se hace doble clic en el para abrirlo. En este paso, lo que se solicita es conectar un puente auxiliar de las boquillas de baja tensión (X0) a la tierra física del transformador, el “Box Trigger” se quedará en la misma posición, se tomará la terminal negra del megger y se colocará en la posición de las boquillas de baja tensión (X0) como se ve en la Figura 59, ahora se abre el “Level Blueprint” y en el nodo “Open Level” se escribe “nivel_6”.

El último paso de esta maniobra consiste de igual manera copiar el “nivel_5” y cambiar su nombre por “nivel_6”, en este paso final es donde se solicita que el usuario comience la prueba con el megger para terminar la práctica. Ahora se abre el “nivel_6” y se selecciona un puente auxiliar, una punta se colocará en la boquilla de baja tensión (X0) y la otra punta en la tierra física del transformador, quedando algo así como se puede observar en la Figura 60.

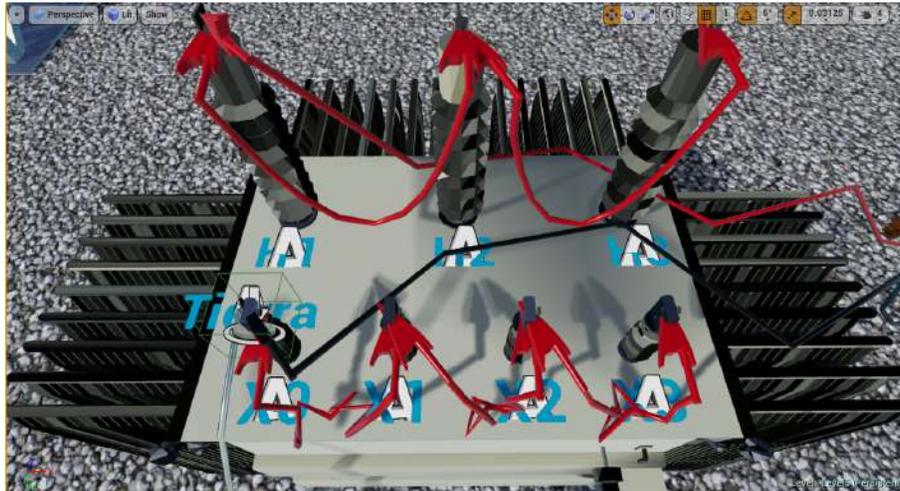


Figura 59: Colocación del trigger para el nivel 6 y conexión de la terminal negra del megger a X0 (elaboración propia).



Figura 60: Colocación del puente auxiliar de la boquilla X0 a la tierra física (elaboración propia).

4.4.7. Programación de interfaces de usuario

Las interfaces de usuario sirven para la guía y ayuda visual para el usuario, se puede decir que hay dos tipos, los estáticos y los 3D que permite la interacción con el usuario de realidad virtual. Como su nombre lo indica, los estáticos o “Widget” son normalmente usados para la creación de menús donde aparecen los botones comunes que debe contener al inicio un programa, tales como: reproducir o iniciar, opciones, salir. Por lo regular se interactúa con el puntero del ratón ya que se necesita que el usuario de un comando de entrada (Figura 61). Los “Widget 3D” fueron desarrollados para la realidad virtual donde no se cuenta con un puntero tradicional como el hardware de ratón, en estos dispositivos se cuenta con los controles de movimiento por lo cual la interacción es diferente y el “Widget 3D” se puede anclar a la vista del usuario o dejarla fija en alguna posición y funciona también como menú o elemento de interacción dentro del software como soporte.



Figura 61: Ejemplo de un Menú con interacción mediante puntero de ratón (elaboración propia).

Para empezar, se debe crear un “actor blueprint” vacío, esto con la finalidad de poder crear la instancia que usará el “Widget 3D” o “widget interaction” en ingles. Para crearlo tan solo hay que dar clic secundario en el “Content Browser” y seleccionar “Blueprint Class” (Figura 42), lo cual aparecerá una ventana con los tipos de blueprint como lo visto en la Figura 43 y se seleccionará “Actor” al principio de la lista y se le colocará un nombre fácilmente recordable para posteriormente dar doble clic para abrir el editor y aplicarle componentes. Para este Actor lo único componente que se le agregará es un “Widget” como se ve en la Figura 62 en la flecha de la izquierda, después se creará el “Widget Blueprint” que usará el actor por lo cual se volverá a hacer clic secundario en el “Content Browser” y en lugar de ir a “Blueprint Class” se coloca el puntero donde dice “User Interface” y saldrá otro menú contextual como se ve en la Figura 63 y se dará clic en “Widget Blueprint”, se le coloca un nombre y devuelta en el editor de blueprint del actor se selecciona el componente widget para que al lado derecho del editor aparezcan los detalles y en la flecha de la derecha de la Figura 62, donde dice “Widget Class” se le dará clic y se buscará el nombre del widget que recientemente se creó en la Figura 63, de esta manera los dos archivos ya estarían ligados para poder hacer uso de ellos y mandarlos llamar.

Ahora que esta listo el Actor, es tiempo de pasar a crear el widget que el usuario va a poder observar, para esto, se necesita el editor de widget (Figura 64) y este editor cuenta con una paleta (número 1) de herramientas que se pueden usar para el diseño gráfico del widget, en el número 2 es la jerarquía de los elementos que se añadieron al proyecto, el número 3 consiste en panel de animación desde la creación de nuevas animaciones, la línea de tiempo de cuanto debe durar una animación y editar su comportamiento, por ejemplo que al mandar llamar el widget aparezca de manera como si desenrollaras un pergamino y cuando lo cierras se vuelva a enrollar, en vez de que aparezca espontáneamente en el aire. En el número 4 es el lugar de la vista previa de como va quedando organizado el diseño visual, el número 5 es donde aparecerán los detalles de la herramienta seleccionada que se encuentre dentro del

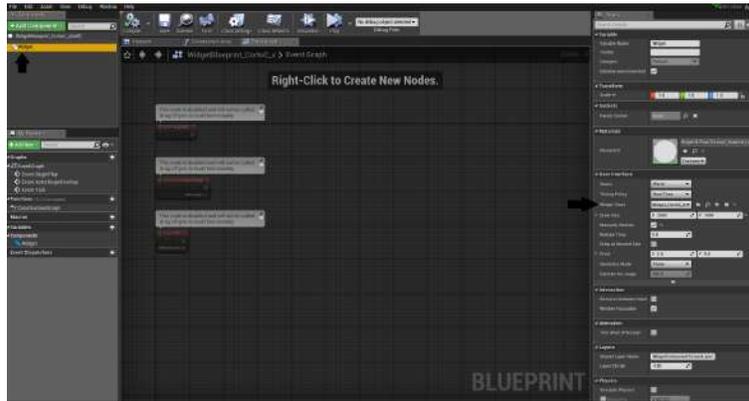


Figura 62: Añadiendo componentes y clases al blueprint del actor que contendrá el widget (elaboración propia).

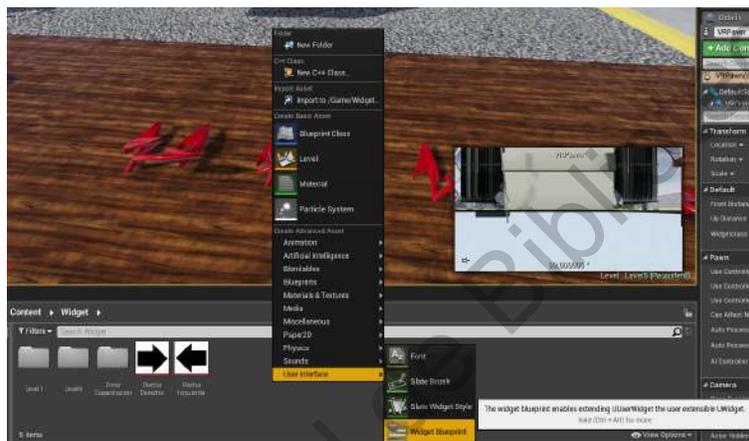


Figura 63: Creación del widget que usará el actor previamente creado (elaboración propia).

proyecto, por ejemplo el texto que se desplegaría dentro de un botón. El número 6 es una sección donde se configura de que modo se trabajará, si esta en color naranja el modo “Designer” es para la creación visual del widget y si se cambia a “Graph” lo que hará el software es cambiar al modo programador para que de esta manera se pueda programar que hace cada elemento, por ejemplo, el botón de iniciar se le agregaría un nodo de “Open level” al nodo del botón de inicio, para que al ser pulsado se abrirá el nivel 1 como se ve en la Figura 65.

Para empezar la creación del widget, se deben arrastrar las herramientas desde la paleta hasta el panel de jerarquía quedando como se ve en la Figura 66. Como se puede observar, el primer elemento de arriba hacia abajo es un “Border” será el límite visual del widget ya desplegado. Se agregan 4 botones, el botón para mandar al usuario a entrenamiento, otro para salir del programa, uno más para ir hacia atrás en las instrucciones y el último para ir hacia adelante en las instrucciones. Dentro de cada botón se debe anexar una paleta de texto para que el botón contenga dentro del mismo el texto que se desea expresar.

Después se agregan 6 bordes mas donde se desplegará la instrucción, para anadirle la instrucción que se requiere a cada paso también se deben agregar una paleta de texto por cada paso. Al dar clic en la paleta de texto, del lado derecho (número 5 Figura 64) aparecerán los detalles de ese panel, por lo cual en cada paso se seleccionará el panel de texto y se le asignará el texto correspondiente, como se ve en la Figura 67, cabe destacar que dentro de los mismo detalles se encuentran opciones de personalización para acomodar el texto, fuente, tamaño, transparencia y color.

Después de colocar los textos se procederá a acomodarlos dentro del panel de vista previa o “view-

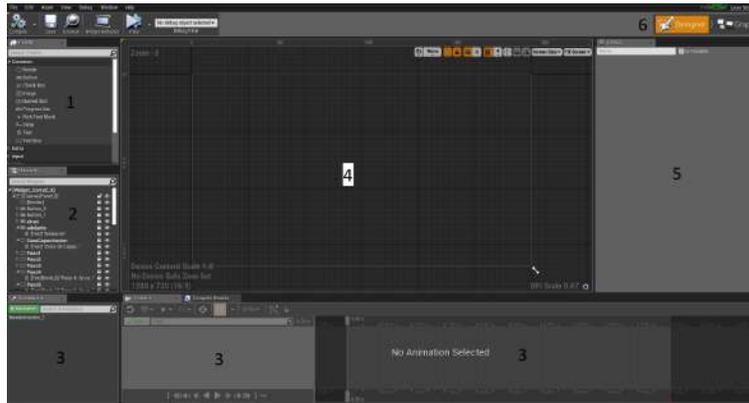


Figura 64: Editor de diseño para las interfaces de usuario (elaboración propia).

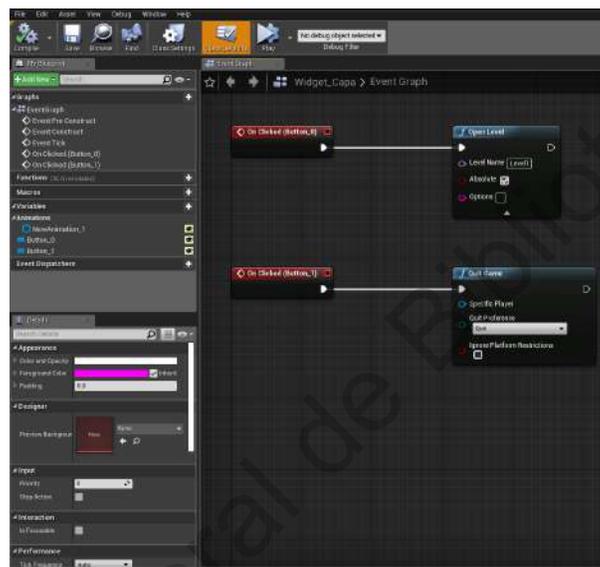


Figura 65: Ejemplo en el editor de diseño para las interfaces de usuario, programación de los botones (elaboración propia).

port” (número 4 Figura 64), se seleccionara el elemento de mas alta jerarquía de cada paso o botón y en el panel de detalles aparecerán las coordenadas para posicionarlos al gusto del diseñador. Un ejemplo de esto se puede encontrar en la Figura 68, donde se despliega la posición en XY y su tamaño, la alineación de lo que contendrá, su color y transparencias, inclusive acepta formatos externos como imágenes para que un botón tenga un diseño particular, etc. Cabe destacar algo importante, cuando se trabaja con widgets que interactúan ya sea con el proyecto en funcionamiento o el usuario interactúe directamente con el, es imprescindible dar clic en la “check box” que dice “is variable” en la parte superior donde esta la flecha en la Figura 68, ya que con esta opción activa es la que permitirá que se pueda ir editando el texto mostrado en pantalla. Ahora con todo listo, la vista previa del widget quedaría algo como se ve en la Figura 69.

Ahora con el diseño listo, se procederá a la programación que corresponden a los botones, como se vio en la Figura 64 en el número 6 se le da clic donde dice “Graph” para pasar al modo programación tal y como se observa en la Figura 70, en esta interfaz el número 1 es lo que compone el blueprint, esto es, funciones, macros y variables. En el número 2, son los detalles que contiene las variables seleccionadas y el número 3 es donde se conectarán los nodos requeridos.

Para comenzar se necesita mandar llamar el evento de presionar del botón, por lo cual en la sección 1 se da clic en el triángulo del “EventGraph” para desplegar el contenido y seleccionar el evento de los



Figura 66: Panel de herramientas usadas para la diseño del Widget (elaboración propia).



Figura 67: Panel de detalles de la herramienta de texto (elaboración propia).

botones que se tengan, ya con el botón seleccionado tan solo se arrastra hacia el “Viewport” o número 3 de la Figura 64 para que el nodo sea creado y conectar el nodo siguiente.

Empezando con los botones de “Empezar” y “Salir”, ya que se crearon los nodos de evento de los botones, se procede a añadir los nodos que realizaran el efecto deseado por el boton, en el caso del botón de entrenamiento se requiere que al presionarlo se abra el nivel 1 de la subestación por lo cual se necesita un nodo de “Open Level” y en el botón de salir se necesita un nodo de “Quit level” para que el programa termine. En la Figura 71 se puede observar la estructura final de nodos para los botones de Entrenamiento y Salir.

Ya con los eventos creados para los botones de “atrás” y “adelante”, se necesita crear una variable para ir guardando el valor del paso para mostrar el texto correspondiente a ese paso en pantalla. Para crear la variable en el número 1 de la Figura 64, en la sección de variables se da clic en el signo “+” y se nombra la variable, se selecciona el tipo de variable a entero y se debe compilar el blueprint para poder darle un valor, en este caso se le asignará un valor de “0” y se arrastra al “viewport” para hacer uso de ella, hay que resaltar que al arrastrar la variable saldrá un menú con dos opciones “Get” y “Set”, el “Get” es para obtener el valor de la variable y con “Set” es para asignar o modificar el valor de la variable, por lo cual al hacer uso de los dos se podrá modificar el valor y cambiar entre pasos.



Figura 68: Panel de detalles de la herramienta de Borde (elaboración propia).

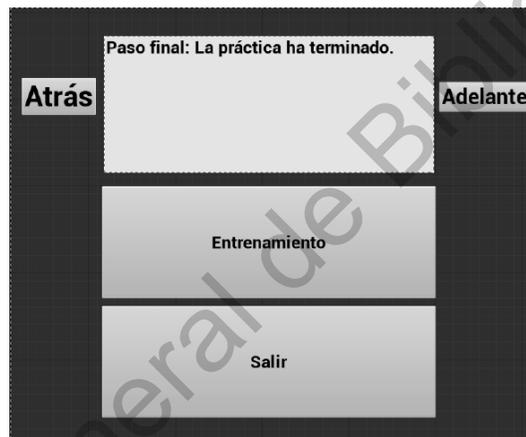


Figura 69: Diseño final del Widget para la interfaz de usuario (elaboración propia).

Entonces para el botón “adelante” se usa una variable en “Get” y con un nodo de “int + int” donde la otro terminal se le asignará el valor de 1, para que siempre que el botón sea presionado se suma 1 al valor de la variable y con un nodo de “Set” recibe la salida del nodo “int + int” para guardar el nuevo valor de la variable, del mismo modo con el botón de “Atrás” se usa la misma estructura cambiando únicamente el nodo de “int + int” a “int - int” para ir decreciendo el valor de la variable y ya con el valor de la variable establecido, ayudará al nodo “Switch on int” para ir eligiendo el caso determinado.

Ahora que ya se tiene tanto el valor de la variable y el switch listo, en cada caso se procede a programar los pasos que debe mostrarse, para esto debido a que se puso los borders como variables se pueden utilizar para mandarlos llamar como un “Get” y con dos nodos de “Set Visibility” se intercalará entre los textos mostrados, en la Figura 72 se muestra la manera en la que quedarían listos los nodos.

Como se explicó en secciones anteriores, hay blueprints esenciales para el funcionamiento del proyecto en este caso el que se necesita es el del funcionamiento del avatar o peón, porque ahora que ya se tiene la programación del widget se tiene que anclar a la cámara del avatar para que el usuario la vea de frente siempre que aparezca la interfaz, por lo tanto se abre el blueprint “MotionControllerPawn” donde se necesita añadir dos eventos personalizados uno que detecte cuando se presiona el botón para que aparezca el menú y otro para destruirlo, la manera de hacerlo se puede observar en la Figura 73.

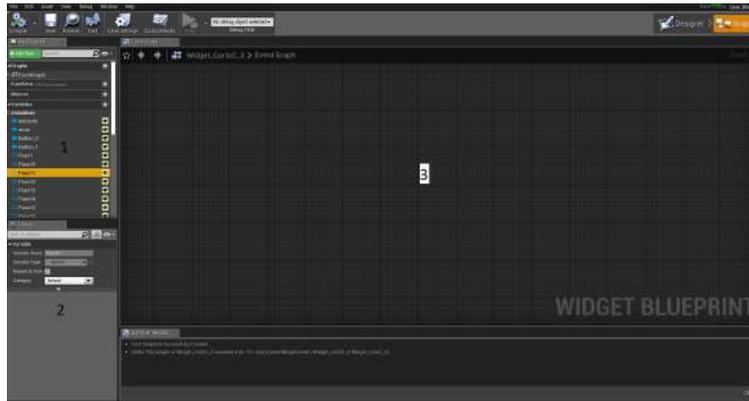


Figura 70: Interfaz de programación en blueprint para widgets (elaboración propia).

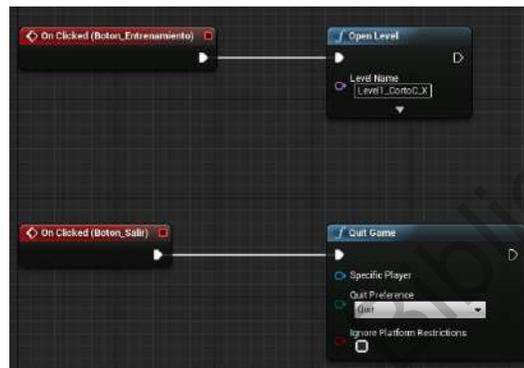


Figura 71: Nodos de programación para los botones de Entrenamiento y Salir (elaboración propia).

Ahora que ya se cuenta con la programación para generarlo o destruirlo, en la Figura 73 se puede encontrar un nodo que se llama “Show Widget” este nodo se encuentra colapsado que básicamente es que esta empaquetado para ahorrar espacio lo que contiene dentro se puede observar en la Figura 74, es código para generar el widget y anclarlo a la cámara de manera que se obtiene la posición del avatar y la rotación de mismo y se la asigna al widget recién generado, además se hace una validación para asegurar que el widget no se genere dentro de otra geometría y si se genera dentro de otra geometría rotar al avatar 180 grados para mostrar el widget.

De esta manera ya se tiene la manera de ver el widget pero no de interactuar con el, por tal motivo se debe ir a otro blueprint específicamente al “BP _ MotionController” para poder añadirle un componente de “SplineMesh” a las manos para que aparezca un láser y con este medio utilizarlo para la interacción.

Como se ve en la Figura 75, donde se ve la flecha del lado izquierdo es el componente de “SplineMesh” agregado además del código necesario para que al momento de usar el láser active los botones del menú.

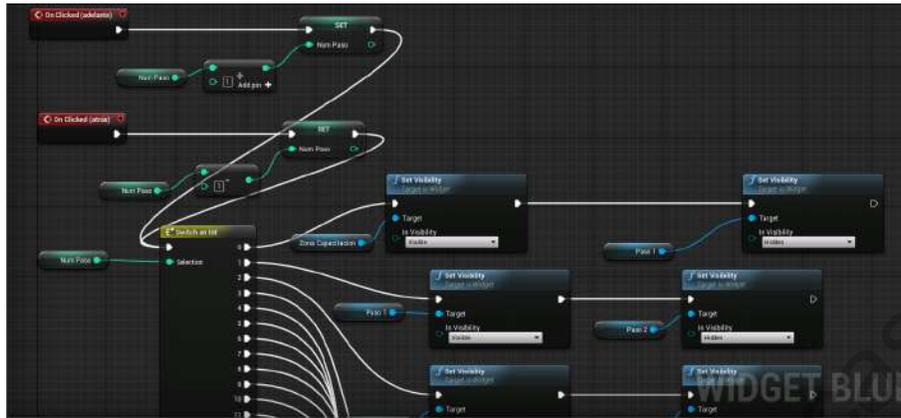


Figura 72: Nodos de programación para los botones de Adelante y Atrás (elaboración propia).

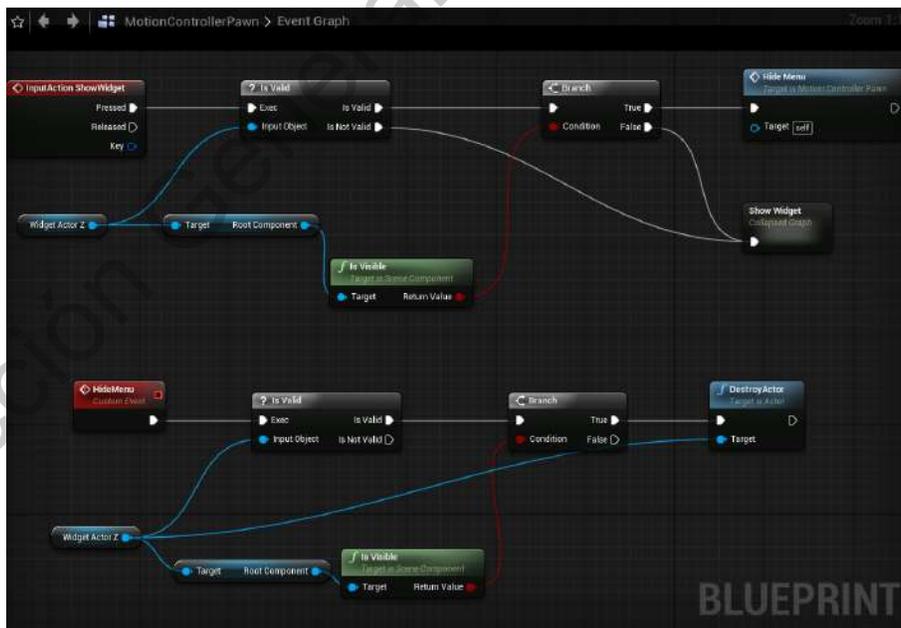


Figura 73: Fragmento del código que genera y destruye el Widget del menú (elaboración propia).

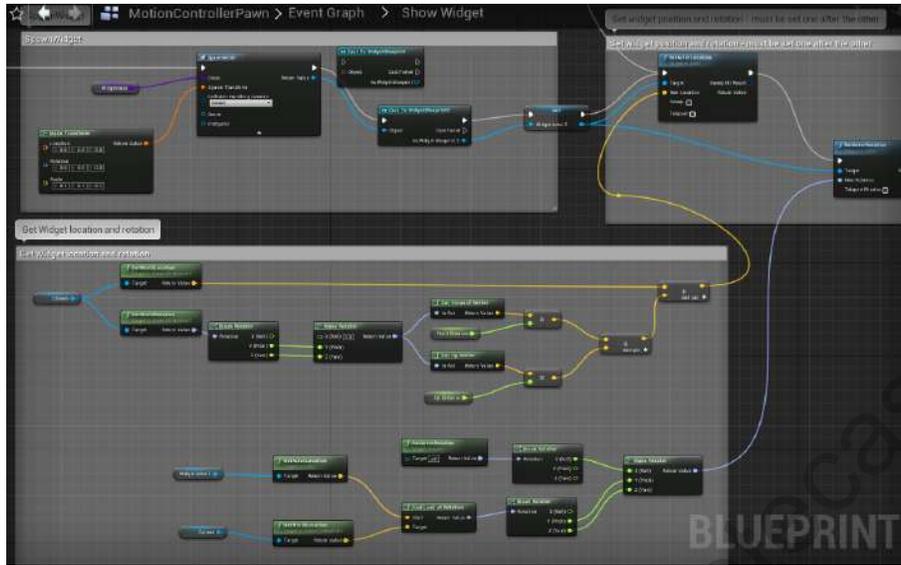


Figura 74: Fragmento del código que ancla el menú a la cámara y comprueba posiciones válidas (elaboración propia).

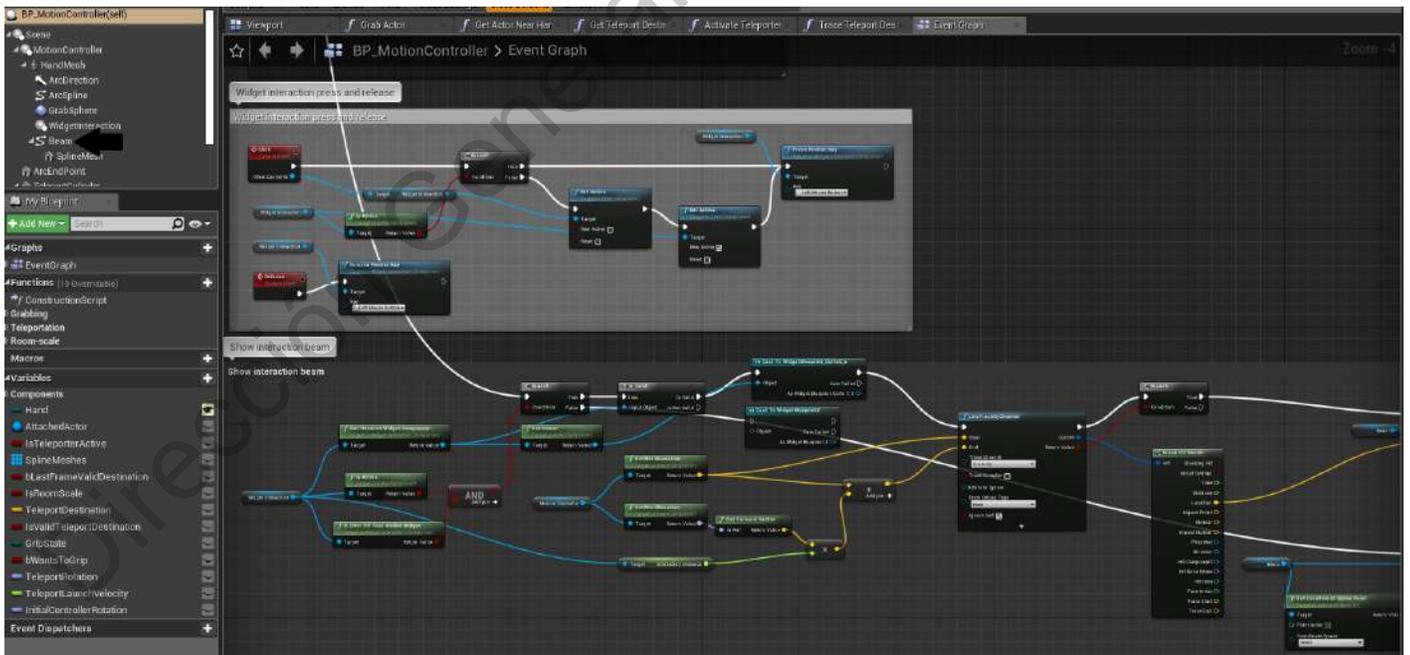


Figura 75: Fragmento del código para interactuar con el widget Menú (elaboración propia).

4.5. Demostración

Para hacer uso del sistema desarrollado, se necesita al menos una tarjeta gráfica GTX 980 de Nvidia y el sistema de realidad virtual que contenga las gafas y sus controles de movimiento. En la Figura 76, se puede observar al usuario utilizar el método convencional y en la Figura 77 es el usuario con el simulador desarrollado en este trabajo.



Figura 76: Usuario utilizando el software original con entrada teclado y ratón (elaboración propia).



Figura 77: Usuario utilizando el simulador desarrollado con sistema de realidad virtual (elaboración propia).

En la Figura 78, se puede observar la vista que el usuario percibe a través del sistema de realidad virtual cuando el simulador ya se está ejecutando y puede desplazarse como se ve en la Figura 79 para ir a cualquier zona dentro del Aula.

Con el usuario ya familiarizado con la movilidad, se puede acercarse a los objetos para inspeccionarlos y aprender de ellos, tal como se ve en la Figura 80 y la Figura 81, que al tomar los objetos se despliega la infografía pudiendo rotarlo en la mano para inspeccionar cada parte del objeto junto con la ayuda visual.

En la Figura 82 y Figura 83, se puede observar como el usuario emplea el widget menú para pasar de instrucción con la interacción del láser además de pasar a la siguiente sección que corresponde al entrenamiento en la subestación.

Cuando el usuario presiona el botón de “Entrenamiento” se carga el nivel de la subestación (Figura 84) la cual puede recorrer para conocer los elementos que componen una subestación eléctrica. En la



Figura 78: Vista que obtiene el usuario al usar el sistema de realidad virtual en la aula (elaboración propia).



Figura 79: Usuario desplazándose hacia la zona de cálculo de transformadores trifásicos (elaboración propia).

Figura 83 se observa como el usuario debe hacer uso del menú para leer la primera instrucción y en la posterior Figura 85 se muestra como realiza la tarea asignada.

En la Figura 86, se ve como ya realizó el primer paso y las boquillas de baja tensión están cortocircuitadas, por lo cual ahora debe pasar al segundo paso que consiste en cortocircuitar las boquillas de alta tensión como se ve en la Figura 87, así el usuario debe continuar con las indicaciones hasta completar la prueba.

Para poner a prueba el sistema y obtener el desarrollo del usuario mediante el uso del simulador, se obtuvo el tiempo total que requería para completar la maniobra, para esto se utilizó el nodo de "Get Game Time in Seconds" para obtener el tiempo transcurrido y el nodo "Print String" marcando la opción de "Print Log" para guardarlo en el registro, de esta manera se agregan al "Level Blueprint" quedando como se ve en la Figura 88 y en la Figura 89 se presenta la manera en la que se registra el nombre del nivel y el tiempo transcurrido.



Figura 80: Usuario sosteniendo el casco con su respectiva infografía (elaboración propia).



Figura 81: Usuario sosteniendo el transformador con su respectiva infografía (elaboración propia).

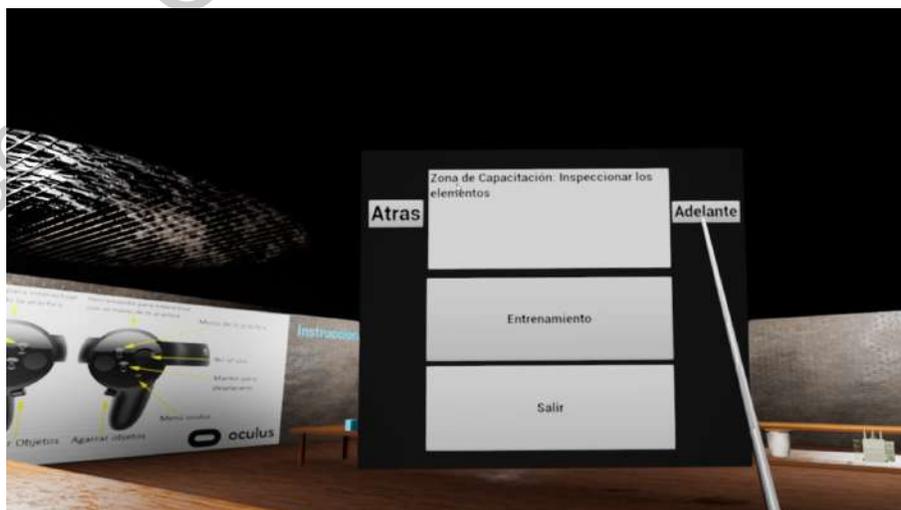


Figura 82: Usuario utilizando el widget con el láser para pasar a la zona de entrenamiento (elaboración propia).



Figura 83: Usuario usando el widget para pasar a la siguiente instrucción (elaboración propia).



Figura 84: Vista que obtiene el usuario al usar el sistema de realidad virtual en la subestación (elaboración propia).

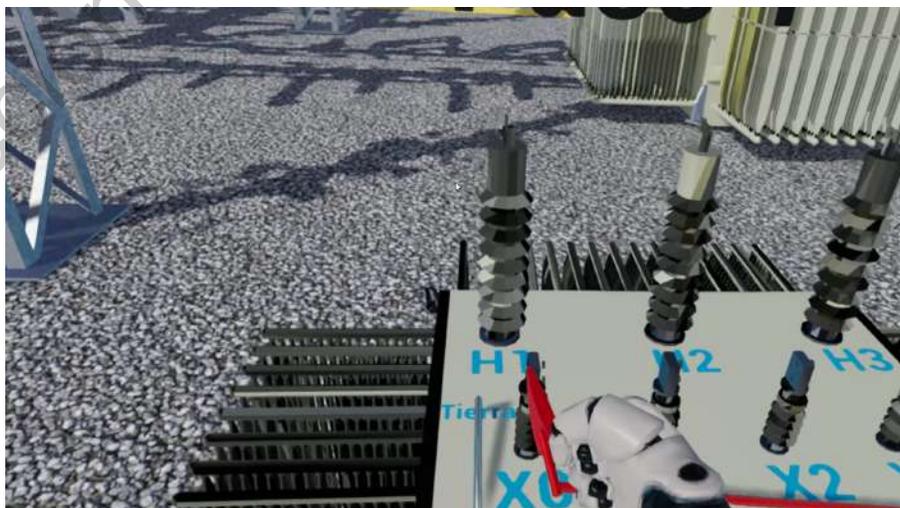


Figura 85: Usuario realizando la primer conexión en la zona de entrenamiento (elaboración propia).



Figura 86: Vista del usuario al completar la instrucción (elaboración propia).

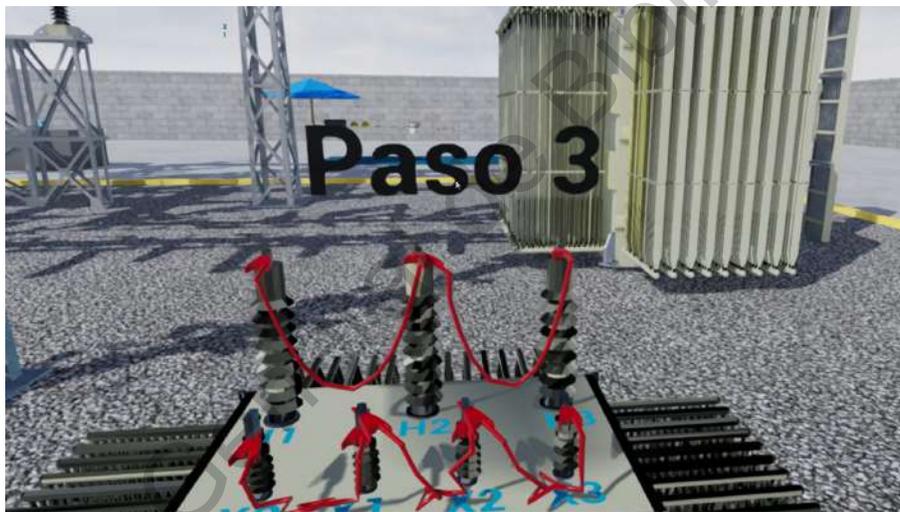


Figura 87: Vista del usuario al seguir las instrucciones (elaboración propia).

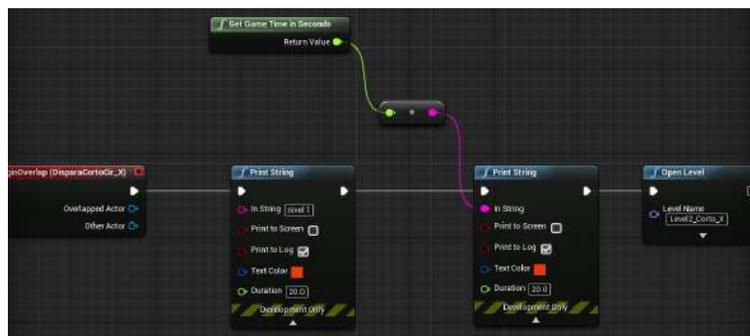


Figura 88: Código para la obtención del tiempo para cada nivel (elaboración propia).

[2019.08.19-23.19.20:062][623]LogD3D12RHI: Warning: Constant Allocator had to grow! Consider making it
[2019.08.19-23.19.25:963][152]LogD3D12RHI: Warning: Constant Allocator had to grow! Consider making it
[2019.08.19-23.19.32:844][769]LogBlueprintUserMessages: [Level1_CortoC_X_C_1] nivel 1
[2019.08.19-23.19.32:844][769]LogBlueprintUserMessages: [Level1_CortoC_X_C_1] 13.884185
[2019.08.19-23.19.32:846][770]LogGameMode: Display: Match State Changed from InProgress to LeavingMap
[2019.08.19-23.19.32:847][770]LogGameState: Match State Changed from InProgress to LeavingMap
[2019.08.19-23.19.32:847][770]LogNet: Browse: /Game/Levels/Level2_Corto_X/Level2_Corto_X
[2019.08.19-23.19.32:849][770]LogHMD: FSplash::OnShow
[2019.08.19-23.19.32:850][770]LogHMD: FSplash::OnShow
[2019.08.19-23.19.32:850][770]LogHMD: FSplash::OnShow
[2019.08.19-23.19.32:850][770]LogHMD: FSplash::OnShow
[2019.08.19-23.19.32:850][770]LogLoad: LoadMap: /Game/Levels/Level2_Corto_X/Level2_Corto_X

Figura 89: Registro obtenido del tiempo en segundos al completar el primer nivel (elaboración propia).

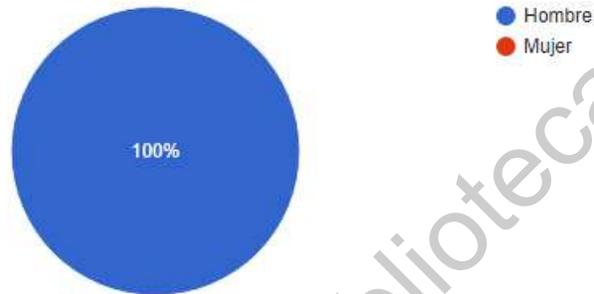
4.6. Certificación del simulador

Para certificar que el empleo del simulador, como una herramienta de enseñanza, que logra progresar las habilidades y destrezas del usuario con cada uso, se desarrolló una metodología la cual consiste en la repetición de la prueba en tres ocasiones, obteniendo su tiempo total y comparar su crecimiento.

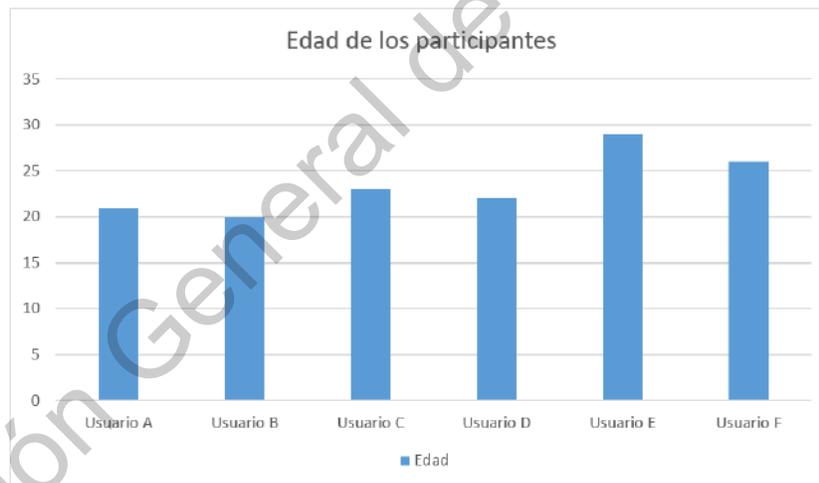
Después de que el participante terminara de realizar las pruebas en el simulador, se le solicita que realice una encuesta y obtener su retroalimentación. La encuesta consta de 10 preguntas las cuales son:

1 ¿Cuál es tu género?

6 respuestas

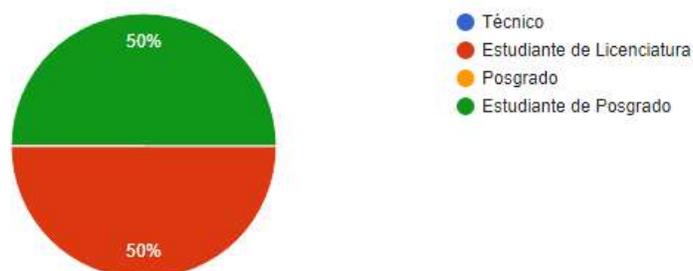


2 ¿Cuál es tu edad?



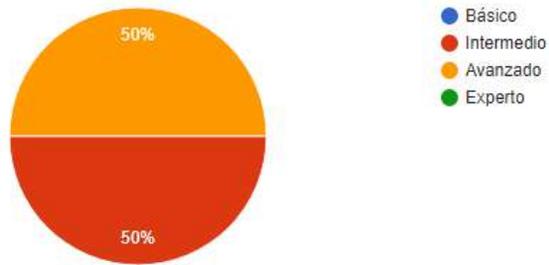
3 ¿Cuál es tu nivel de estudios?

6 respuestas



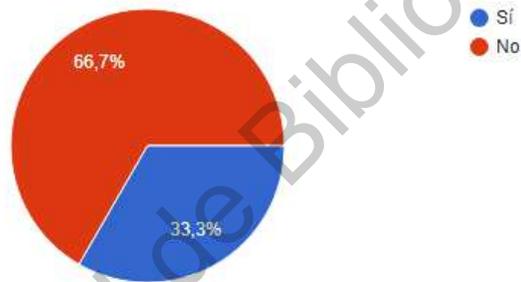
4 ¿Qué nivel de uso de dispositivos digitales manejas?

6 respuestas



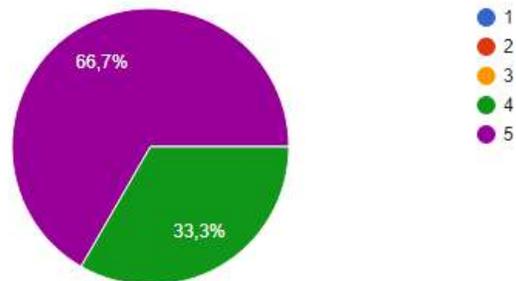
5 ¿Habías utilizado algún dispositivo de realidad virtual (Oculus, HTC vive, Samsung, leapmotion, otros)?

6 respuestas



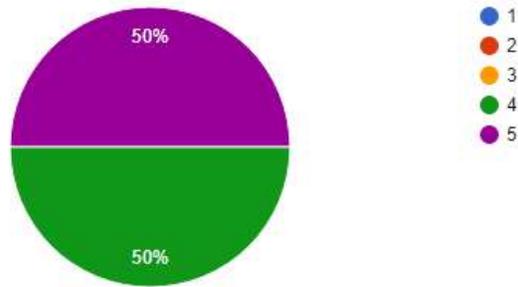
6 Del 1 al 5 ¿Se te hizo amigable el sistema?

6 respuestas



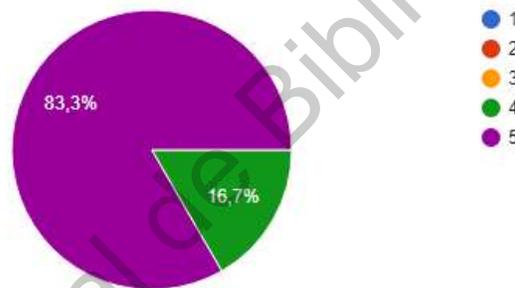
7 Del 1 al 5 ¿El sistema emula y ayuda a aprender lo necesario para realizar la prueba real?

6 respuestas



8 Después de realizar la práctica en el simulador del 1 al 5, ¿Qué tanta confianza te daría aplicarlo en el ámbito real?

6 respuestas



9 ¿Que mejorarías al sistema VR?

- Un poco los comandos.
- Más amplitud en el campo de trabajo.
- Un espacio simulado al de la habitación.
- Que pudiera ver el funcionamiento del dispositivo que conecté y que entregue resultados.
- El menú de instrucciones un poco más grande la letra y los botones.
- Ninguno.

10 ¿Que problema se te presento en el uso del sistema VR?

- Pensé que el botón de elegir era el A como en el Xbox.
- Ninguno.
- El espacio.
- Por ser la primera vez que uso un dispositivo de estos, el uso de los botones, es decir, ¿Qué es para cada cosa (para agarrar, soltar, etc.)?
- Los botones de las instrucciones, principalmente el botón de siguiente.
- Por el mismo sistema se tiene la confusión del espacio y en ciertos casos se debía girar en algunos casos, debería corregirse esa parte para no dañar el sistema y que no se dañe el usarlo al moverse demasiado, pues no esta consiente del exterior.

El registro de los tiempos en porcentaje se muestra en la tabla 4, tomando el tiempo total del primer intento de referencia contra el ultimo y analizar su desempeño al ver si su tiempo disminuyo.

Tabla 4: Porcentaje de mejora respecto al tiempo de los usuarios (elaboración propia).

Registro tiempos en porcentaje			
Usuario	Prueba inicial (Seg.)	Prueba final (Seg.)	Porcentaje de mejora
A	253.75	42.4	83.29 %
B	420.62	119.01	71.70 %
C	312.83	51.99	83.38 %
D	296.07	56.48	80.92 %
E	308.92	106.02	65.68 %
F	289.17	107.54	62.81 %

Se realizaron modificaciones posteriores a la prueba con usuarios al widget de instrucciones, se agrandaron los botones para facilitar la operación al usuario, como se puede observar en la Figura 91 respecto a la Figura 90.



Figura 90: Widget de instrucciones pasado con botones pequeños (elaboración propia).



Figura 91: Widget de instrucciones corregido con botones más grandes (elaboración propia).

5. Conclusiones

Se desarrolló un sistema de capacitación y entrenamiento para el área eléctrica utilizando el sistema de realidad virtual Oculus rift y el motor gráfico “Unreal Engine”. Los ambientes que se eligieron fueron el de un Aula para la capacitación y una subestación para la realización de la maniobra del caso propuesto, se aplicaron texturas y materiales de alta resolución así como mapas de luz realista para generar una alta inmersión.

Se desarrollaron los modelos de los equipos y herramientas necesarias a escala para el progreso de la prueba y generar la percepción de estar sujetando algo real, de igual manera con sus texturas y materiales de alta calidad.

La práctica que se propuso fue la prueba de resistencia de aislamiento a un transformador de potencia para que el alumno se familiarice con el procedimiento de mantenimiento, esto es, se realizó un instructivo paso a paso de como debe ejecutarse la maniobra para que el alumno obtenga el conocimiento y que pueda aplicarlo en un entorno seguro.

Se jerarquizó el entrenamiento de la persona para que obtenga las habilidades y destrezas imprescindibles en la praxis, de tal manera que para certificar la validez del uso del simulador, se empleó la metodología propuesta para medir el tiempo obtenido por usuario, en la tabla 4 se demuestra como fue la tendencia a mejora de cada usuario examinado, tomando de referencia el tiempo total de su primera prueba contra su tercer intento. La encuesta respondida por los usuarios en un rango de 20 a 29 años siendo todos hombres, son estudiantes de licenciatura y posgrado donde el 50 % tiene un uso avanzado de dispositivos digitales mientras tanto el otro 50 % son expertos, cabe resaltar que la realidad virtual es algo innovador dado que el 66.7 % de los usuarios no habían experimentado la realidad virtual frente al 33.3 % que si lo han hecho. El 66.7 % califico el simulador con la máxima calificación en cuanto si el sistema es amigable frente al 33.3 % que le dio la segunda mejor calificación demostrando que la realidad virtual es una tecnología sumamente llamativa y versátil. De igual manera el 50 % considera que el simulador si cumple al aportar lo necesario para realizar la prueba en la vida real y el otro 50 % considera que es muy buena aunque creen que algunos detalles pueden mejorar. En la pregunta sobre la confianza de aplicarlo en la vida real después de usar el simulador, el 83.3 % tienen plena confianza con una calificación de 5 contra el 16.7 % que sienten bastante confianza al calificarla con 4.

Cuando se les pregunto sobre que mejoras pondrían al sistema, se concentraron en dos principalmente, una en el espacio de trabajo ya que al usar el sistema caminaban en lugar de teletransportarse necesiándose ser recolocados al centro del área de trabajo, la segunda es acerca de la interfaz con los botones pidiendo que sean mas grandes aumentando el tamaño de letra como se puede ver en la Figura 91 respecto a la Figura 90, la opinión sobre poder ver el funcionamiento del dispositivo se realizará para próximos proyectos.

Respecto a los problemas que presentaron los usuarios, se resume en que fue la primera vez que utilizaban la realidad virtual, por lo cual los botones del control y lo que hacen al presionarlos llegaban a confundirlos, esto se podria solucionar al crear una animación que muestre como funcionan las manos antes de que realicen las pruebas, de nueva cuenta surge el tema del espacio que de igual manera la solución que se propone es la malla de seguridad que provee el software de “Oculus Rift” indicándoles cual es el espacio seguro de trabajo. Como prospectivas se podría hacer uso de los sonidos, añadir el típico sonido de subestación del efecto piel en los cables de alta tensión y música de fondo en el aula, se podría mejorar el anclaje de los puentes auxiliares así como añadir funcionalidad a los equipos e incluir la malla de seguridad.

Referencias

- BEATTIEA, N., HORANA, B. & MCKENZIEA, S. (2015). Taking the leap with the oculus hmd and cad - plucking at thin air? *Procedia Technology, ELSEVIER* .
- BERNARDO, A. (2017). Virtual reality and simulation in neurosurgical training. *From the Annals of Weill Cornell Neurological Surgery* .
- BUCKLEY, S. (2015). This is how valve is amazing lighthouse tracking technology works. *Web URL* <https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768>. Access Date: april 17, 2018.
- BUCKWALD, M. & HOLZ, D. (2017). Leap motion. *Web URL* <https://www.leapmotion.com/>. Access Date: april 17, 2018.
- BUTT, A. L. & KARDONG-EDGREN, S. (2018). Using game-based virtual reality with haptics for skill acquisition. *Clinical Simulation in Nursing, ELSEVIER* .
- CAPUTO, F., GRECO, A., D'AMATO, E., NOTAROA, I. & S.SPADAB (2017). On the use of virtual reality for human-centered workplace design. *Structural Integrity Procedia, ELSEVIER* .
- CASTILLO-ROSA, J., SUÁREZ-CEBADOR, M., RUBIO-ROMERO, J. C. & AGUADO, J. A. (2016). Personal factors and consequences of electrical occupational accidents in the primary, secondary and tertiary sectors. *Safety Science, ELSEVIER* .
- CHARARA, S. (2016). Explained: How does vr actually work? *Web URL* <https://www.wearable.com/vr/how-does-vr-work-explained>. Access Date: april 15, 2018.
- DUROVIS (2019). Durovis dive. *Web URL* <https://www.durovis.com/en/index.html>. Access Date: october 7, 2019.
- FOVE, C. (2016). Fove vr. *Web URL* <https://www.getfove.com/>. Access Date: april 17, 2018.
- GONZÁLEZ, I. (2019). La generación t. *Web URL* <https://eduka2.es/la-generacion-t/>. Access Date: september 14, 2019.
- GOOGLE (2019a). Cardboard vr. *Web URL* <https://vr.google.com/cardboard/>. Access Date: october 7, 2019.
- GOOGLE (2019b). Daydream vr. *Web URL* <https://vr.google.com/daydream/>. Access Date: october 7, 2019.
- GUILLEN, D. Z., DEL CARMEN MORENO SAEZ, M. & LOPEZBOSCH, M. A. (2007). Aplicaciones didacticas de la realidad virtual al museo pedagogico de arte infantil .
- HAMROL, A., GÓRSKI, F., GRAJEWSKI, D. & ZAWADZKI, P. (2013). Virtual 3d atlas of a human body – development of an educational medical software application. *Procedia Computer Science, ELSEVIER* .
- HMD-TECH (2018). Homido. *Web URL* <https://homido.com/#>. Access Date: october 7, 2019.
- HTC (2019). Vive vr. *Web URL* <https://www.vive.com/us/product/vive-virtual-reality-system/>. Access Date: october 7, 2019.

- IMMERSIVEEDUCATION (2018). Immersive education. Web URL <https://immersivelrn.org>. Access Date: april 18, 2018.
- INMERSIVELEARNING (2018). Immersive learning. Web URL <http://immersiveducation.org>. Access Date: april 18, 2018.
- JAY (2016). Field of view for virtual reality headsets explained. Web URL <https://vr-lens-lab.com/field-of-view-for-virtual-reality-headsets/>. Access Date: april 15, 2018.
- KASURINEN, J. (2017). Usability issues of virtual reality learning simulator in healthcare and cybersecurity. *Procedia Computer Science, ELSEVIER* .
- KRUEGER, M. (1991). Artificial reality ii. *Reading: Addison-Wesley* .
- LAKENTO, I. (2019). Lakento. Web URL <https://www.lakento.com/index-es.html>. Access Date: october 7, 2019.
- LOWOOD, H. E. (2015). Virtual reality. *Encyclopædia Britannica* URL <https://www.britannica.com/technology/virtual-reality>. Access Date: april 20, 2018.
- MAKRANSKYA, G., TERKILDSENA, T. S. & MAYERB, R. E. (2017). Adding immersive virtual reality to a science lab simulation causes more presence but less learning. *Learning and Instruction, ELSEVIER* .
- MARTEL, E. & MULDER, K. (2017). Controlling vr games: control schemes and the player experience. *Entertainment Computing, ELSEVIER* .
- MICROSOFT (2019a). Hololens 2. Web URL <https://www.microsoft.com/en-us/hololens>. Access Date: october 7, 2019.
- MICROSOFT (2019b). Kinect for windows. Web URL <https://developer.microsoft.com/en-us/windows/kinect>. Access Date: october 7, 2019.
- NEURODIGITALTECHNOLOGIES (2017). Gloveone. Web URL <https://www.neurodigital.es/gloveone/>. Access Date: april 17, 2018.
- NOLO, I. (2019). Nolo motion tracking for vr/ar. Web URL <https://www.nolovr.com/index>. Access Date: october 7, 2019.
- OCULUS (2018). Oculus rift. Web URL <https://www.oculus.com/rift/#oui-csl-rift-games=star-trek>. Access Date: october 7, 2019.
- PAESA, D., ARANTESB, E. & IRIZARRYC, J. (2017). Immersive environment for improving the understanding of architectural 3d models: Comparing user spatial perception between immersive and traditional virtual reality systems. *Automation in Construction, ELSEVIER* .
- PLAYSTATION, S. (2019). Playstationvr. Web URL <https://www.playstation.com/es-mx/explore/playstation-vr/>. Access Date: october 7, 2019.
- POTKONJAK, V., GARDNER, M., CALLAGHAN, V., MATTILA, P., GUETL, C., PETROVIC, V. M. & JOVANOVIC, K. (2016). Virtual laboratories for education in science, technology, and engineering: A review. *Computers & Education, ELSEVIER* .

- QUISHPE-ARMAS, J. A., CEDEÑO-VIVEROS, L. D., MELÉNDEZ-CAMPOS, J., SUÁREZ-MORA, C. A. & CAMACHO-LEON, S. (2015). An immersive 3d virtual learning environment for analyzing the atomic structure of mems-relevant materials. *Procedia Computer Science* .
- SAMSUNG (2018). Samsung gear vr. *Web URL* <http://www.samsung.com/es/wearables/gear-vr/>. Access Date: april 1, 2018.
- SIXENSE, I. (2019). The sixense full-presence system. *Web URL* <https://www.sixense.com/platform/hardware/>. Access Date: october 7, 2019.
- STARVR (2019). Starvr. *Web URL* <https://www.starvr.com/>. Access Date: october 7, 2019.
- VALDEZ, M. T., MARTINS, M. J. M. & BARBOSA, F. P. M. (2014). Virtual labs in electrical engineering education the vema environment. *Higher Education and Training (ITHET)* .
- VALVE, I. (2019). Steam vr. *Web URL* <https://store.steampowered.com/steamvr>. Access Date: october 7, 2019.
- WITMER, B. G. & SINGER, M. J. (1998). Measuring presence in virtual environments: A presence questionnaire. *Presence* **7**(3), 225-240.
- YOST-LABS (2019). Priovr. *Web URL* <https://yostlabs.com/priovr/>. Access Date: october 7, 2019.
- ZEISS-INTERNATIONAL (2019). Vr one plus. *Web URL* <https://www.zeiss.com/virtual-reality/home.html>. Access Date: october 7, 2019.