



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Software para la construcción de  
sistemas hidráulicos en un entorno  
virtual**

**TESIS**

Que para obtener el título de  
**Ingeniero Mecatrónico**

**P R E S E N T A N**

David de Jesus Cárdenas Rodríguez

Luis Fernando Pérez Herrera

**DIRECTOR DE TESIS**

M. en A. Luis Yair Bautista Blanco



**Ciudad Universitaria, Cd. Mx., 2019**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

# Agradecimientos

Agradezco en primer lugar a Dios por permitirme estudiar una carrera universitaria y para que esta me sirva como medio para mejorar la calidad de vida de las personas.

Agradezco infinitamente a mis padres David y Connie, por todas las enseñanzas que me han transmitido, por convertirse en mi principal motivo para vivir. Gracias por todos los valores que me enseñaron, por enseñarme a amar, a respetar, a ser honesto, comprometido, justo y responsable. Gracias por cuidarme cuando estoy enfermo, por darme todo lo necesario para vivir y sobre todo por el amor incondicional que me tienen.

Agradezco a mi hermana Sarai por ser mi cómplice de tantas aventuras, por ser la hermana mayor que siempre me enseña muchas cosas, por ser mi doctora de cabecera y por estar ahí siempre que la necesito.

Agradezco a mis abuelos Luis, Conchita, Lolo y Lola, por ser mi fuente de sabiduría, por preocuparse por mí y por tratar de darme siempre lo mejor para salir adelante. Especialmente a mi abuela Lola, por ser mi segunda madre.

Agradezco a mis tíos y primos, que siempre me han apoyado durante toda mi vida, por estar presentes y por ser parte fundamental de lo que soy.

Agradezco a mis amigos de la carrera y el bachillerato por compartir conmigo tantas alegrías, desveladas, partidos de fútbol, enojos y enseñanzas que fueron fruto de la amistad que hemos cultivado por estos 8 años.

Agradezco a mis amigos de secundaria y primaria, los cuales han estado en la mayoría de mis momentos más importantes y porque a pesar de la distancia nunca hemos roto esa amistad que nos une.

Agradezco a mi compañero de tesis y amigo Luis Fernando (Rock Lee) por aventurarse conmigo en esta tesis, por su apoyo y confianza incondicional. Gracias por ser parte de este suceso tan importante, por todos los momentos felices que tuvimos, por crear este lazo de amistad que nació en la escuela y se consolidó en la realización de esta tesis y por nunca decir que no a echar una reta.

Agradezco a mis profesores de la carrera, por transmitirme su conocimiento y consejos tanto académicos como de la vida. Especialmente quiero agradecer a: Ana Marissa, Rolando Peralta, Yukihiro Minami, Edmundo Ruiz, por ser de los mejores profesores que he tenido en mi vida.

Agradezco a mi asesor Luis Yair Bautista Blanco, por ser no solo mi asesor de tesis, sino también por haberme brindado su amistad. Gracias por todo el conocimiento transmitido y por compartir el gusto por los videojuegos.

Agradezco a la Facultad de Ingeniería y a la UNAM, por haberme adoptado desde el bachillerato y abrirme sus puertas para ser parte de la máxima casa de estudios. Gracias por haberme formado como ingeniero y como persona crítica con el fin de mejorar la situación del país.

**David Cárdenas**

A mis padres, por todo el amor, cariño y apoyo incondicional que me han brindado a lo largo de toda mi vida. Por todas las enseñanzas, educación, consejos y valores que he recibido de su parte, gracias a ellos soy quien soy como persona, por todas las horas sacrificadas de sueño para que yo tuviera siempre un desayuno listo y me llevaran a la escuela, por cuidarme cuando estoy enfermo, por ir siempre a los eventos importantes en los que participo, por eso y más los amo.

A mis hermanos, los cuales siempre han creído en mí y me han apoyado en los momentos donde más lo he necesitado.

A mis sobrinos, que al estar con ellos siento una gran felicidad y todos los problemas por los que puedo estar pasando se me olvidan.

A mi novia Zuri que me ha acompañado a lo largo de la carrera, brindándome todo su apoyo y cariño cuando más lo necesitaba, aguantándome en momentos de estrés y enojo, ayudándome en mis tareas y cuando tenía que estudiar para algún examen, pero sobre todo, por todos los momentos de felicidad que he pasado a su lado.

A todos mis amigos, porque gracias a ellos pude tener experiencias inolvidables a lo largo de la carrera. Por tantos trabajos, proyectos y desveladas que pasamos juntos, por ser personas en las que puedo confiar y que están dispuestas a siempre apoyarme.

A mi amigo y compañero de tesis David, por ser una gran persona que siempre me ha brindado su apoyo y su confianza, por compartir sus gustos y experiencias conmigo, por todos los momentos que pasamos juntos durante la realización de este trabajo, por aquellas noches en las que jugamos y nos divertimos como niños pequeños, por acompañarme a lo largo de esta aventura y enseñarme el valor de la amistad.

A mi asesor Luis Yair Bautista Blanco, que además de ser un excelente profesor, es un gran amigo que me ha apoyado antes y durante la tesis. Por compartir su pasión por los videojuegos conmigo y brindarme consejos cuando tenía alguna inquietud, pero sobre todo por poner su confianza en mí.

A cada uno de los alumnos y profesores que colaboraron de alguna manera con el proyecto e hicieron posible la realización del mismo.

A la Facultad de Ingeniería y a la Universidad Nacional Autónoma de México, por abrirme sus puertas y darme la oportunidad de pertenecer a la máxima casa de estudios del país y brindarme todas las herramientas y el conocimiento para mi éxito profesional.

**Luis Fernando**

**Agradecimientos de ambos.**

Trabajo realizado gracias al Programa de Apoyo a Proyectos para Innovar y Mejorar la Educación UNAM-PAPIME PE108419.

Índice	1
Lista de Figuras	3
Lista de Tablas	5
Introducción	6
<b>1. PLANTEAMIENTO DEL PROBLEMA</b>	<b>7</b>
<b>1.1 Problemática</b>	<b>7</b>
<b>1.2 Objetivos generales</b>	<b>7</b>
1.2.1 Objetivos específicos	7
<b>2. ANTECEDENTES</b>	<b>8</b>
<b>2.1 Sistemas hidráulicos</b>	<b>8</b>
2.1.1 Concepto de hidráulica	8
2.1.2 Componentes de un sistema hidráulico	9
<b>2.2 Software de modelado</b>	<b>12</b>
2.2.1 Blender	12
2.2.2 3ds Max	13
2.2.3 Maya	13
2.2.4 Selección del software para modelado	14
<b>2.3 Realidad Virtual</b>	<b>15</b>
2.3.1 Head-mounted Display (HMD)	15
2.3.2 Field of View (FOV)	16
2.3.3 Head tracking	17
2.3.4 Motion tracking	19
2.3.5 HMD actuales	21
2.3.5.1 HMD a utilizar	26
<b>2.4 Programas para el desarrollo de entornos virtuales</b>	<b>26</b>
2.4.1 Unity	26
2.4.1.1 Programación en Unity	27
2.4.2 Unreal Engine 4	29
2.4.2.1 Programación en Unreal Engine 4	29
2.4.3 Comparación entre Unity y Unreal Engine	31
2.4.4 Selección del software para el desarrollo del entorno virtual	32
<b>3. DESARROLLO ITERACIÓN 1</b>	<b>33</b>
<b>3.1 Modelado de elementos hidráulicos</b>	<b>33</b>
3.1.1 Tubería recta	34
3.1.2 Placas de orificio	35
3.1.3 Bomba de agua	37
<b>3.2 Componentes de Unreal Engine 4</b>	<b>39</b>

3.2.1 Terminología de Unreal Engine 4	39
3.2.2 Interfaz de Unreal Engine 4	42
3.2.3 Blueprints	43
3.2.4 Iluminación	44
3.2.5 Materiales	44
3.2.6 Sistema de partículas	45
3.2.7 Widgets	46
<b>3.3 Desarrollo del entorno virtual</b>	46
3.3.1 Diseño del entorno virtual	46
<b>3.4 Partícula del agua</b>	52
<b>3.5 Sistema de construcción</b>	53
3.5.1 Interacción del software	53
3.5.2 Desarrollo de Blueprints	54
3.5.2.1 Enumeraciones	54
3.5.2.2 Estructuras	55
3.5.2.3 Actores	56
3.5.2.4 Personaje	60
3.5.3 Desarrollo de la interfaz del usuario	60
3.5.4 Funcionamiento	62
<b>3.6 Pruebas Iteración 1</b>	66
<b>3.7 Resultados Iteración 1</b>	68
<b>4. DESARROLLO ITERACIÓN 2</b>	74
4.1 Cambios efectuados	74
4.2 Pruebas Iteración 2	76
4.2 Resultados Iteración 2	76
<b>5. CONCLUSIONES</b>	79
<b>6. TRABAJO A FUTURO</b>	80
<b>7. REFERENCIAS</b>	81
<b>APÉNDICE I: MODELOS DE ELEMENTOS HIDRÁULICOS</b>	86

# Lista de Figuras

## CAPÍTULO 2

Figura 2.1	Aplicación en maquinaria pesada [3].	11
Figura 2.2	Principio de Pascal [4].	12
Figura 2.3	Representación gráfica de una prensa hidráulica [4].	12
Figura 2.4	Bomba hidráulica [6].	13
Figura 2.5	Flujo después de una placa de orificio [7].	14
Figura 2.6	Placa de orificio estándar [9].	14
Figura 2.7	Tipos de placas de orificio [10].	15
Figura 2.8	Logotipo de <i>Blender</i> [11].	16
Figura 2.9	Logotipo de <i>3ds Max</i> [13].	16
Figura 2.10	Interfaz de inicio de <i>Maya 2019</i> .	17
Figura 2.11	HMD y dispositivos de interacción [16].	18
Figura 2.12	Principio de funcionamiento del <i>HMD</i> [17].	19
Figura 2.13	Campo de visión horizontal de los humanos [18].	19
Figura 2.14	Campo de visión vertical de los humanos [19].	20
Figura 2.15	Campo de visión en realidad virtual [20].	20
Figura 2.16	Balanceo [23].	21
Figura 2.17	Cabeceo [23].	21
Figura 2.18	Guiñada [23].	22
Figura 2.19	Representación gráfica de los tres grados de libertad [20].	22
Figura 2.20	Mando de movimiento <i>PlayStation Move</i> [25].	23
Figura 2.21	Controlador <i>Oculus Go</i> [26].	24
Figura 2.22	Ejemplo de manos virtuales [27].	24
Figura 2.23	Elementos que conforman el <i>HTC VIVE</i> [29].	25
Figura 2.24	<i>Google Cardboard</i> [30].	26
Figura 2.25	<i>PlayStation VR</i> [31].	26
Figura 2.26	<i>Oculus Go</i> [32].	27
Figura 2.27	Configuración y tienda de la aplicación <i>Oculus</i> .	28
Figura 2.28	Editor de niveles de <i>Unity</i> [36].	30
Figura 2.29	Ejemplos de <i>GameObjects</i> en <i>Unity</i> [37].	30
Figura 2.30	Componente de luz en el editor de niveles de <i>Unity</i> (izquierda) y la ventana <b>Inspector</b> de dicho componente (derecha) [37].	31
Figura 2.31	Logotipo de <i>Unity</i> [37].	31
Figura 2.32	Logotipo de <i>Unreal Engine</i> [39].	32
Figura 2.33	Ejemplo de programación por bloques mediante <i>Blueprints</i> [39].	33
Figura 2.34	Ejemplo de desarrollo de un entorno virtual mediante realidad virtual [39].	33

## CAPÍTULO 3

Figura 3.1	Sistema hidráulico del laboratorio de la Facultad de Química.	36
Figura 3.2	Abstracción de la tubería recta.	37
Figura 3.3	Modelo de tubería recta.	37
Figura 3.4	Número de triángulos de la tubería recta.	38
Figura 3.5	Modelo de la placa de orificio concéntrica.	39
Figura 3.6	Modelo de la placa de orificio segmentada.	39
Figura 3.7	Número de triángulos de la placa de orificio concéntrica.	40



Figura 3.8	Número de triángulos de la placa de orificio segmentada.	40
Figura 3.9	Motobomba <i>munich</i> centrífuga [40].	41
Figura 3.10	Modelo de la bomba de agua.	41
Figura 3.11	Número de triángulos de la bomba de agua.	42
Figura 3.12	Ejemplo de Objeto en <i>Unreal Engine</i> [41].	42
Figura 3.13	Clase creada en lenguaje <i>C++</i> y en <i>Blueprints</i> [41].	43
Figura 3.14	Ejemplos de Actores [41].	43
Figura 3.15	Componente de luz puntual [41].	44
Figura 3.16	Ejemplos de peones no controlados [41].	44
Figura 3.17	Movimiento bípedo de una Personaje [41].	45
Figura 3.18	Ejemplo de un Nivel creado en <i>Unreal Engine</i> [41].	45
Figura 3.19	Interfaz de <i>Unreal Engine 4</i> [42].	46
Figura 3.20	Ejemplo de las opciones del <i>Blueprint</i> tipo Personaje [43].	47
Figura 3.21	Actores de luz en <i>Unreal Engine 4</i> [45].	48
Figura 3.22	Ejemplo de un Material y el Editor de Material en <i>Unreal Engine 4</i> [47].	48
Figura 3.23	Editor de Sistemas de partículas <i>Cascade</i> [49].	49
Figura 3.24	Ejemplo de barra de estado (HUD) [51].	50
Figura 3.25	Entorno virtual.	51
Figura 3.26	Texturas del techo, piso y pared (de izquierda a derecha).	51
Figura 3.27	Programación del material original para la pared y su previsualización.	52
Figura 3.28	Instancia del material de la pared y su previsualización.	52
Figura 3.29	Material aplicado a la pared.	53
Figura 3.30	Material aplicado al piso.	53
Figura 3.31	Material aplicado a las vigas.	54
Figura 3.32	Material aplicado al techo.	54
Figura 3.33	Muestra de la placa de orificio segmentada junto con su descripción.	55
Figura 3.34	Instrucciones para interactuar en el entorno virtual.	55
Figura 3.35	Primer diseño del entorno virtual.	56
Figura 3.36	Diseño final de la partícula de agua.	56
Figura 3.37	<i>Minecraft</i> el videojuego tipo <i>sandbox</i> más popular.	57
Figura 3.38	Categorías de <i>Blueprints</i> .	58
Figura 3.39	Enumeración utilizada.	58
Figura 3.40	Valores de la enumeración “e_EstadosDeJugador”.	59
Figura 3.41	Estructuras realizadas.	59
Figura 3.42	Variables de la estructura “es_DatosDeConstrucción”.	60
Figura 3.43	Variables de la estructura “es_DatosDeRecurso”.	60
Figura 3.44	Categorías de Actores.	61
Figura 3.45	Componentes de Actor.	61
Figura 3.46	Contenido del Componente de Actor “BpC_Recursos_Jugador”.	62
Figura 3.47	Contenido del Componente de Actor “BpC_DirectorDeConstrucción”.	62
Figura 3.48	<i>Blueprints</i> de los elementos hidráulicos.	63
Figura 3.49	Recursos.	63
Figura 3.50	Propiedades de del recurso “Bp_Recurso_Vidrio”.	64
Figura 3.51	Contenido de la categoría Personaje.	64
Figura 3.52	Interfaz de prueba con usuarios.	65
Figura 3.53	Interfaz de construcción.	66
Figura 3.54	Interacción entre el controlador y la interfaz de construcción.	67
Figura 3.55	Previsualización de una tubería recta.	67

Figura 3.56	Construcción de una tubería recta.	68
Figura 3.57	Diagrama de flujo que representa el funcionamiento del sistema de construcción.	69
Figura 3.58	Transmisión del <i>Oculus Go</i> al celular en la etapa de preparación de recursos.	72
Figura 3.59	Transmisión en el desarrollo de la prueba.	73
Figura 3.60	Desarrollo de pruebas de iteración 1.	73
Figura 3.61	Gráfico de resultados de la pregunta 4 de la encuesta.	74
Figura 3.62	Gráfico de resultados de la pregunta 5 de la encuesta.	74
Figura 3.63	Gráfico de resultados de la pregunta 6 de la encuesta.	75
Figura 3.64	Gráfico de resultados de la pregunta 8 de la encuesta.	75
Figura 3.65	Gráfico de resultados de la pregunta 12 de la encuesta.	76
Figura 3.66	Gráfico de resultados de la pregunta 13 de la encuesta.	76
Figura 3.67	Gráfico de resultados de la pregunta 16 de la encuesta.	77
Figura 3.68	Gráfico de resultados de la pregunta 17 de la encuesta.	77

#### **CAPÍTULO 4**

Figura 4.1	Segundo diseño del entorno virtual.	78
Figura 4.2	Botones resaltados en las instrucciones.	79
Figura 4.3	Pantalla de carga.	79
Figura 4.4	Desarrollo de pruebas con dos <i>Oculus Go</i> .	81
Figura 4.5	Realización de pruebas con alumnos de la Facultad de Química.	81
Figura 4.6	Gráfico de resultados de iteración 2 de la pregunta 4 de la encuesta.	82
Figura 4.7	Gráfico de resultados de iteración 2 de la pregunta 12 de la encuesta.	82
Figura 4.8	Gráfico de resultados de iteración 2 de la pregunta 13 de la encuesta.	83

## Lista de Tablas

Tabla 1	Encuesta proporcionada a los usuarios.	71
---------	--	----

# Introducción

En el presente documento se explica el desarrollo de un software en realidad virtual, el cual permite la construcción de sistemas hidráulicos mediante una interfaz intuitiva y que utiliza las características que ofrece el visor de realidad virtual *Oculus Go*.

En el proyecto se realiza el modelo 3D de algunos elementos hidráulicos, el diseño de un entorno virtual que tiene similitud a un laboratorio, la programación y diseño de un fluido virtual y se describe cada una de las partes que componen el programa elaborado en un motor gráfico.

En el capítulo 1 se presenta la problemática por la cual se decidió realizar este trabajo, así como el objetivo general y los objetivos específicos. Por otra parte, en el capítulo 2 se presenta una recopilación de información necesaria para darle contexto al trabajo. Se expone la definición de un sistema hidráulico, así como los elementos que lo componen. Posteriormente se presentan algunos programas de modelado que se utilizan en la actualidad en industrias de videojuegos, cinematográficas, etc. De la misma manera, se presentan los conceptos primordiales que hacen posible el funcionamiento de la realidad virtual, así como algunos de los principales visores que dominan el mercado actual. Por último, se describen las características fundamentales de dos de los motores gráficos más importantes actualmente y las diferencias que tienen entre sí.

En el capítulo 3 se presenta el desarrollo de la primera iteración del software; en él se detallan los elementos hidráulicos que se modelaron y las características que tiene cada uno. Se expone información relacionada con el motor gráfico utilizado, la cual es importante para entender a qué se refieren los términos mencionados durante el desarrollo del entorno virtual, la partícula de agua y el sistema de construcción. Por otra parte, se realizan pruebas con alumnos de la Facultad de Ingeniería para validar diferentes aspectos del software.

En el capítulo 4 se realiza una segunda iteración del software, en donde se efectúan cambios a raíz de los comentarios generados en la primera iteración. Nuevamente se realizan pruebas, pero esta vez con alumnos de la Facultad de Química.

En el capítulo 5 se presentan las conclusiones que se obtuvieron al realizar este trabajo, las cuales consideran los resultados obtenidos de las diferentes pruebas y la validación del software. El capítulo 6 ahonda en el trabajo a futuro que se puede elaborar para mejorar el proyecto.

Finalmente, este documento cuenta con un apéndice en el cual se encuentran los modelos 3D implementados en el desarrollo del proyecto.

# 1. PLANTEAMIENTO DEL PROBLEMA

En este capítulo se presenta la problemática a la que se enfrentan los alumnos de la Facultad de Química durante la realización de prácticas de laboratorio. Del mismo modo, se exponen el objetivo general y los objetivos específicos del presente trabajo.

## 1.1 Problemática

Dentro de las estrategias de aprendizaje de la Facultad de Química, se encuentra el desarrollo de prácticas de laboratorio, las cuales tienen como objetivo complementar la educación en las aulas. De manera que los alumnos reafirmen conceptos fundamentales adquiridos en los salones de clase mediante la experimentación y la observación de comportamientos reales de los mismos. No obstante, los laboratorios tienen ciertas limitantes de espacio, de recursos o inclusive físicas. Para el caso particular de la Facultad de Química, las limitantes que poseen son interactivas y físicas.

Entre la diversidad de prácticas que se realizan en los laboratorios, existe una en específico que se fundamenta en el uso de tuberías, placas de orificios y una bomba de agua. El objetivo de esta práctica consiste en observar la carga de trabajo en la bomba y además analizar el comportamiento que tiene el agua al pasar por una placa de orificios. Es en el último propósito en donde surge la limitante. En primer lugar, el intercambio de las placas de orificios toma mucho tiempo, por lo que antes de su realización se debe tener la placa ya montada en el sistema hidráulico. Por otro lado, las tuberías del laboratorio son opacas, por lo que resulta imposible que los alumnos observen el comportamiento del agua.

En consecuencia, los alumnos no cuentan con los elementos idóneos para reafirmar por completo sus conocimientos y deben de tomar las explicaciones de los profesores como dogmas. Es aquí donde se ve un área de oportunidad.

## 1.2 Objetivos generales

Desarrollar un software para la construcción de sistemas hidráulicos, permitiendo reafirmar conceptos fundamentales mediante la experimentación y observación.

### 1.2.1 Objetivos específicos

- Desarrollar el modelo 3D de tuberías y placas de orificios.
- Diseñar y programar un sistema de construcción en un entorno virtual.
- Diseñar un entorno virtual asemejando a un laboratorio.
- Diseñar una interfaz intuitiva al usuario.

## 2 . ANTECEDENTES

### 2.1 Sistemas hidráulicos

A lo largo de la historia se ha usado la energía de los líquidos, en especial la del agua, para diversas aplicaciones, como norias o prensas hidráulicas, las cuales son capaces de sustituir la fuerza humana o animal por la fuerza del agua [1].

Hoy en día la hidráulica tiene aplicaciones muy variadas, debido a los grandes avances tecnológicos que se han presentado a lo largo de los años, logrando la fabricación de elementos de mayor precisión y de mejor calidad, lo que sin duda ha permitido un creciente desarrollo en la industria en general.

#### 2.1.1 Concepto de hidráulica

La hidráulica se entiende como la transmisión de fuerzas y movimientos por medio de líquidos, es decir, se utilizan como modo de transmisión de energía necesaria para mover y hacer funcionar mecanismos [2].

La diferencia entre hidráulica y neumática, radica fundamentalmente en el fluido de trabajo; en hidráulica, se utilizan líquidos a presión, en lugar de hacerlo con aire comprimido como en el caso de la neumática.

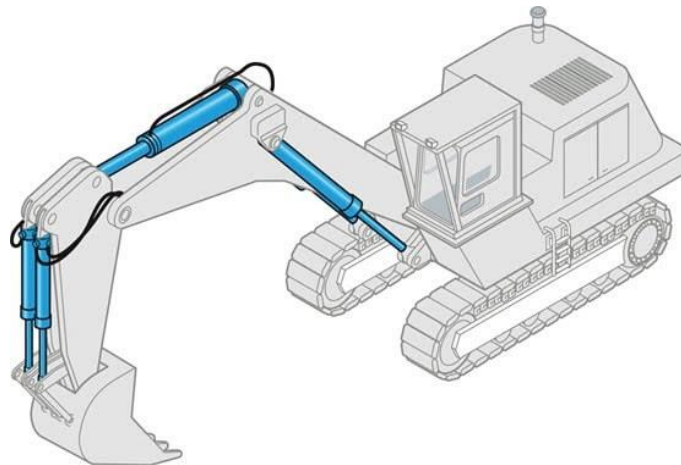


Figura 2.1 Aplicación en maquinaria pesada [3].

La incompresibilidad de los líquidos es lo que hace posible la energía hidráulica. Esto fue descubierto en 1647 por Blaise Pascal, quien hizo una observación que hoy es conocida como principio de Pascal: *“La presión ejercida sobre un fluido poco compresible y en equilibrio dentro de un recipiente de paredes indeformables se transmite con igual intensidad en todas las direcciones y en todos los puntos del fluido”* [4]. La prensa hidráulica

constituye la aplicación fundamental del principio de Pascal y también es un dispositivo que permite entender mejor su significado.

Consiste en dos cilindros de diferente sección, pero comunicados entre sí, y cuyo interior está completamente lleno de un líquido, el cual puede ser agua o aceite. Se hace uso de dos émbolos de diferentes secciones que se colocan respectivamente en cada uno de los cilindros, de modo que estén en contacto con el líquido [4].

Cuando se aplica una fuerza ( $F_1$ ) sobre el émbolo de menor sección ( $A_1$ ) la presión ( $P_1$ ) que se origina en el líquido se transmite de forma casi instantánea a todo el resto del líquido. Por el principio de Pascal, esta presión será igual a la presión ( $P_2$ ) que ejerce el líquido en la sección mayor ( $A_2$ ), es decir:

$$P_1 = P_2 \Rightarrow \frac{F_1}{A_1} = \frac{F_2}{A_2}$$

Figura 2.2 Principio de Pascal [4].

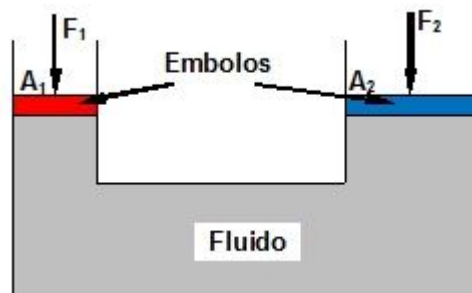


Figura 2.3 Representación gráfica de una prensa hidráulica [4].

## 2.1.2 Componentes de un sistema hidráulico

### Bomba hidráulica

Se activa gracias a la acción de un motor eléctrico y convierte la energía mecánica en energía hidráulica. Proporciona un caudal de líquido adecuado al sistema, con esto cabe señalar que la bomba no produce presión, únicamente flujo; la presión se produce por acción de la resistencia al flujo. La resistencia puede ser ocasionada por tuberías o mangueras utilizadas en el sistema [5].



Figura 2.4 Bomba hidráulica [6].

### Red de distribución

Conformada por tuberías y conexiones, una red de distribución permite la circulación del fluido a través del sistema, además debe garantizar la presión y velocidad en todos los puntos del fluido donde sea utilizado. Aunado a esto, permite el retorno del fluido para garantizar que se vuelva a utilizar una y otra vez [5].

### Placa de orificio

Es un instrumento que se utiliza para la medición del caudal dentro de una tubería. En un principio era usado con gases, sin embargo, debido a lo versátil que es, tiene la capacidad de ser utilizado con diferentes fluidos, comúnmente agua.

Es de los instrumentos más usados en la industria, debido a su bajo costo y a la gran facilidad que tienen para ser reemplazadas, no obstante, la falta de precisión es un gran inconveniente que poseen.

El funcionamiento de la placa de orificio consiste en incrementar la velocidad de flujo y decrementar la presión. Una vez que el flujo pasa por la placa de orificio, se observa una reducción de su sección transversal, y se alcanza su velocidad máxima y presión mínima en la vena contracta<sup>1</sup> (véase Figura 2.5) [7].

---

<sup>1</sup> **Vena contracta:** Es el punto en una corriente de fluido donde el área transversal de la corriente es menor, y la velocidad del fluido es máxima [8].

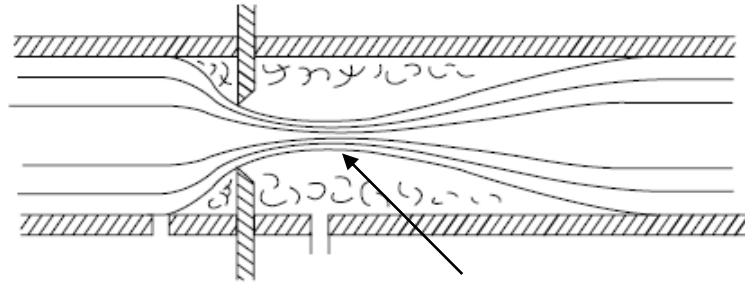


Figura 2.5 Flujo después de una placa de orificio [7].

La placa de orificio estándar en sí misma es un disco circular; generalmente acero inoxidable, de 0.12 a 0.5 pulgadas de espesor, dependiendo del tamaño y la velocidad del flujo, con un orificio en el centro y una pestaña que se proyecta hacia un lado [7].



Figura 2.6 Placa de orificio estándar [9].

Una placa de orificio tiene diferentes características [7], [9]:

1. Presión: Puede soportar presiones de hasta 40 MPa.
2. Temperatura: Puede llegar a soportar temperaturas de hasta 800°C, sin embargo, se suelen utilizar en rangos de -29°C a 121°C.
3. Tamaño: El tamaño máximo que puede tener, es el tamaño de la tubería utilizada.
4. Fluidos: Puede trabajar con líquidos, vapores y gases.
5. Material de construcción: No existen limitaciones para el material de construcción de una placa de orificio, no obstante, se suele utilizar acero, acero inoxidable, níquel, aleaciones níquel-cobre y una aleación de cobalto, cromo, níquel y molibdeno.

A su vez, existen cuatro tipos de placas de orificios [9], [10]:

1. Concéntricas: Es la forma básica de una placa de orificio, en donde el orificio se encuentra en el centro. Su uso se recomienda para fluidos limpios y de baja viscosidad.
2. Concéntricas cónicas: Al igual que las concéntricas, el orificio se encuentra en el centro, con la particularidad de que el diámetro se va reduciendo a medida que el fluido va atravesando el disco. Se recomienda su uso con fluidos que tienden a comportarse de forma turbulenta.



3. Excéntricas: Son aquellas en donde el orificio no se encuentra en el centro del disco, están elevadas levemente o se encuentran levemente debajo del centro. Suelen utilizarse en tuberías con diámetro pequeño.
4. Segmentadas: La diferencia con las placas concéntricas, es que el orificio no es un círculo, ya que tiene forma de un semicírculo o una “media luna”. Por otra parte, pueden ser concéntricas o excéntricas. Se utilizan principalmente para la medición de fluidos que contienen partículas en suspensión.

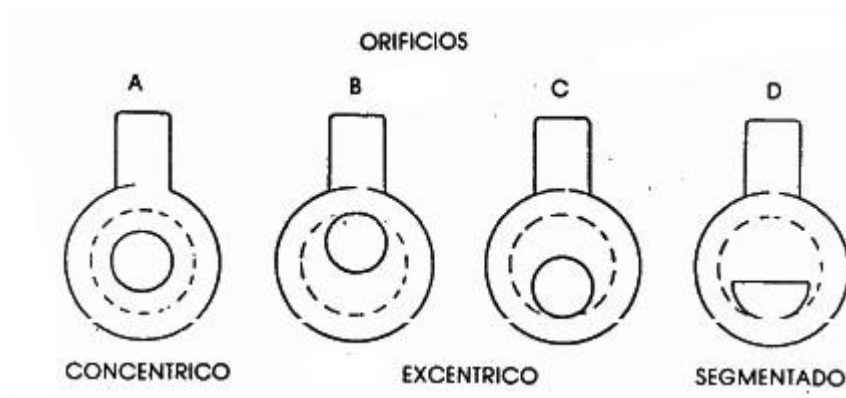


Figura 2.7 Tipos de placas de orificio [10].

## 2.2 Software de modelado

Los programas de modelado ofrecen la capacidad de esculpir, animar, iluminar, pintar y dar efectos visuales a objetos que quieran modelar. En este caso, las herramientas que se utilizarán en el software de modelado son las correspondientes a la función de esculpir.

### 2.2.1 Blender

*Blender* es un software gratuito dedicado al modelado, animación, simulación, renderizado<sup>2</sup>, seguimiento de movimiento, incluso edición de video y creación de juegos. Debido a que es un proyecto impulsado por la comunidad mediante la Licencia Pública General, el público está facultado para realizar pequeños y grandes cambios en la base del código, lo que conlleva a nuevas características, correcciones de errores y una mayor facilidad para usar las herramientas que contiene *Blender* [11].

Al ser un software gratuito, *Blender* es un muy usado por empresas pequeñas en la industria del cine y videojuegos, ya que existe una gran documentación que permite aclarar cualquier duda que se tenga durante el desarrollo de un proyecto.

<sup>2</sup> **Renderizado:** Es un término usado en computación para referirse al proceso de generar una imagen foto realista desde un modelo 3D [12].



Figura 2.8 Logotipo de *Blender* [11].

### 2.2.2 *3ds Max*

*Autodesk 3ds Max* es un software de modelado, renderizado y animación 3D, para la visualización del diseño y juegos. Proporciona un conjunto de herramientas eficaz y flexible para crear el mejor contenido en el menor tiempo posible. Es uno de los programas de animación 3D más utilizados, especialmente para la creación de videojuegos, anuncios de televisión, en arquitectura o en películas [13].

Debido a la gran cantidad de herramientas con las que cuenta, es un software que se recomienda a aquellas personas que ya tienen experiencia trabajando con programas de modelado. Empresas reconocidas a nivel mundial en la industria del cine, utilizan este programa para llevar a cabo grandes cintas cinematográficas, como es el caso de las películas *Avatar* y *Alicia en el país de las maravillas*.



Figura 2.9 Logotipo de *3ds Max* [13].

### 2.2.3 *Maya*

*Autodesk Maya* es un software de animación 3D con poderosas herramientas de modelado, renderizado, simulación, texturizado y animación para artistas, modelistas y animadores. El *software* proporciona una serie de herramientas integradas, utilizadas principalmente para animación, ambientes, gráficos animados, realidad virtual y creación

de personajes. Es empleado para generar assets 3D para usarlos en películas, televisión, desarrollo de videojuegos y arquitectura [14].

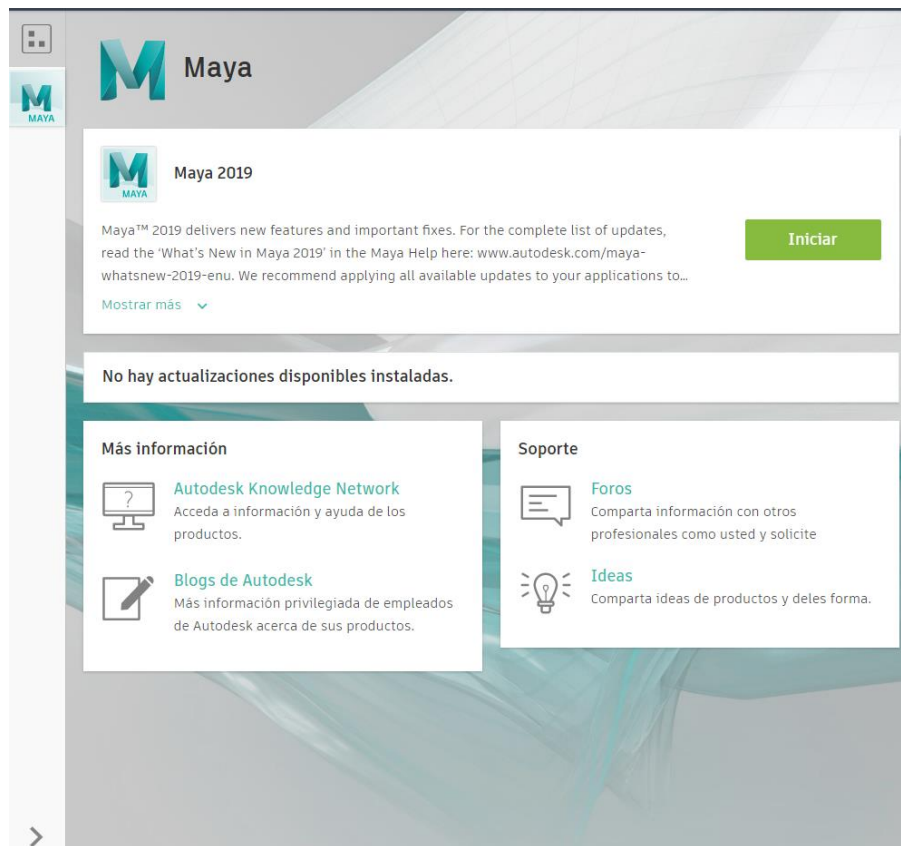


Figura 2.10 Interfaz de inicio de *Maya 2019*.

#### 2.2.4 Selección del software para modelado

Como se acaba de ver, los programas para modelado que se tomaron en cuenta fueron *Maya*, *Blender* y *3ds Max*. Estos 3 programas son de los más utilizados en cuanto a modelado y existe documentación suficiente para aprender a manejarlos. *Maya* y *3ds Max* son programas distribuidos por *Autodesk*, ambos cuentan con un potente apartado de modelado y también se encuentran disponibles con una licencia gratuita para estudiantes; lo que los diferencia es que *Maya* está orientado a la animación y entretenimiento, por lo que cuenta con más funciones para esas áreas. Por otro lado, *3ds Max* tiene funciones más avanzadas para la configuración de materiales lo que es más útil para las áreas de arquitectura, ingeniería y construcción [13]. Por último, se tiene a *Blender*, el cual es un software de licencia libre y con un apartado de modelado muy competitivo.

En resumen, los 3 programas tienen los elementos necesarios para modelar los elementos hidráulicos antes mencionados. Conjuntamente, los modelos que se hagan en estos programas se pueden migrar con facilidad a motores gráficos como *Unity* o *Unreal Engine 4*. De manera que el factor por el que se utilizó cierto software fue el *expertise*.

Anteriormente, ya se había hecho uso del programa *Maya*, por lo que se consideró idóneo que fuera el software a utilizar para realizar el modelado de la mejor manera.

## 2.3 Realidad Virtual

Las aplicaciones de realidad virtual (VR por sus siglas en inglés) pretenden conceder al usuario la sensación de presencia (definida como la sensación de estar “dentro” de un entorno virtual) el cual se alcanza gracias a factores como la inmersión, participación e interacción. Para proporcionar dicha sensación de presencia, la realidad virtual permite al usuario experimentar un entorno virtual a través de estímulos sensoriales coordinados y simultáneos. Esto es logrado con dispositivos de interacción especiales (p.ej. dispositivos de interacción espacial, *head mounted displays* o HMD, dispositivos hápticos, etc.) [15].



Figura 2.11 HMD y dispositivos de interacción [16].

### 2.3.1 Head-mounted Display (HMD)

El modelo perceptual detrás del HMD es modificar completamente la información visual hacia los ojos del usuario y reemplazarla con una perspectiva global del entorno virtual. Esto se logra montando una o dos pantallas muy cerca enfrente de los ojos del usuario con un sistema de lentes para permitir un mayor enfoque natural. A partir de que las pantallas están muy cerca de los ojos del usuario, cualquier parte de una pantalla es únicamente visible en un ojo, volviendo al sistema autoestereoscópico<sup>3</sup> [17].

---

<sup>3</sup> **Pantallas autoestereoscópicas:** Son aquellas pantallas que no requieren gafas porque la distancia física entre los ojos es suficiente para filtrar las imágenes, es decir, dar la sensación de profundidad [17].

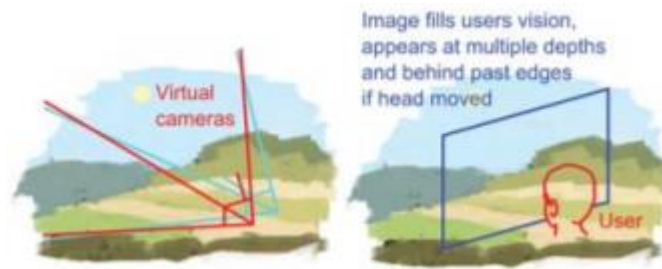


Figura 2.12 Principio de funcionamiento del HMD [17].

### 2.3.2 Field of View (FOV)

El campo de visión o *field of view* es el tamaño angular total de la imagen que puede ser visible para ambos ojos. La visión binocular es un tipo de visión que explica cómo los humanos son capaces de percibir imágenes tridimensionales de sus alrededores [18].

El campo de visión horizontal total de los humanos es de 200°, sin embargo, el campo de visión binocular promedio es de 120°.

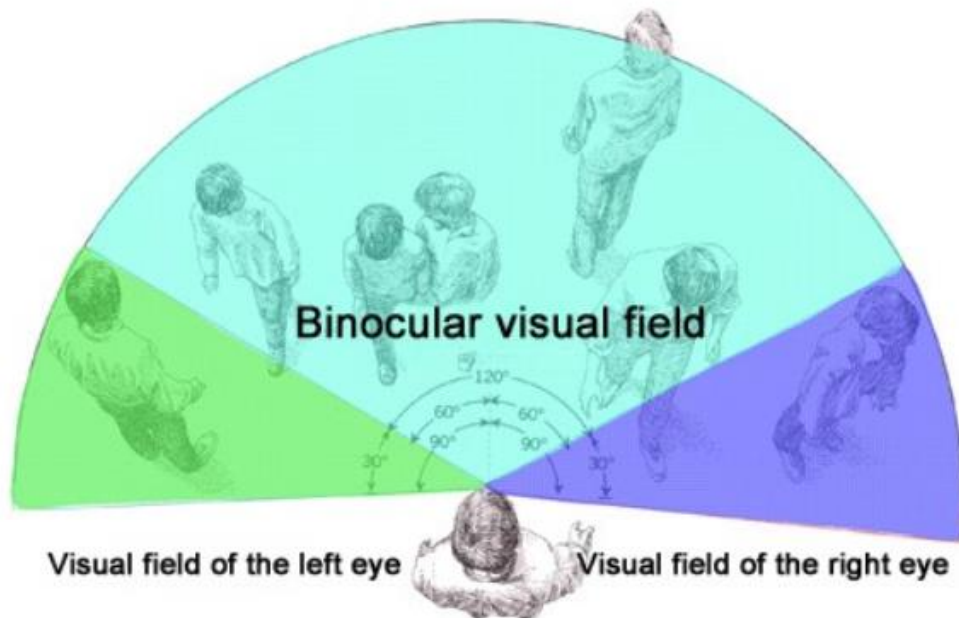


Figura 2.13 Campo de visión horizontal de los humanos [18].

Por otra parte, el campo de visión vertical promedio es de 130°.

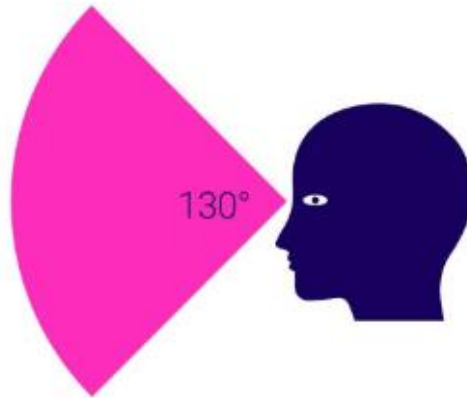


Figura 2.14 Campo de visión vertical de los humanos [19].

En el caso de la realidad virtual, es un aspecto importante a considerar durante el desarrollo de una aplicación, puesto que se usa para proveer a los usuarios de una percepción realista de su ambiente. De manera simple, el campo de visión en realidad virtual se refiere a qué tan ancha será la imagen. El campo de visión se modifica en función del grado de visualización que poseen los visores; actualmente, los visores manejan un campo de visión entre 100° y 210°, lo que las compañías consideran suficiente para la mayoría del contenido en realidad virtual [20].

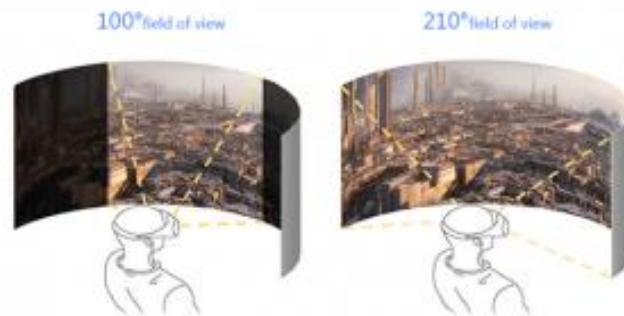


Figura 2.15 Campo de visión en realidad virtual [20].

### 2.3.3 Head tracking

Uno de los objetivos que ha tenido la realidad virtual en los últimos años, es recrear el movimiento humano natural y sobre todo, cómo medir con precisión para que el movimiento de un usuario en el mundo real, corresponda a un cambio idéntico en el mundo virtual.

El seguimiento de cabeza o *head tracking* se refiere a la forma en que la imagen mostrada en el visor, cambiará a medida que el usuario mire hacia arriba, hacia abajo o de un lado a otro [20].

El seguimiento de cabeza funciona mediante tres sensores que son vitales en cualquier visor de realidad virtual: giroscopio<sup>4</sup>, acelerómetro<sup>5</sup> y magnetómetro<sup>6</sup>.

Estos sensores permiten que el dispositivo detecte cómo se mueve el usuario en tres tipos de rotación direccional (también conocido como tres grados de libertad). Registran ciertos movimientos del usuario que se traducen y se ejecutan en tiempo real en el programa de realidad virtual [23]. Los tres grados de libertad con los que cuentan los visores de realidad virtual son:

### **Roll**

*Roll* o balanceo es cuando la cabeza gira de un lado a otro (véase figura 2.16). Este giro es realizado respecto al eje Z.



Figura 2.16 Balanceo [23].

### **Pitch**

*Pitch* o cabeceo es cuando la cabeza gira hacia delante o hacia atrás (véase figura 2.17). Este giro es realizado respecto al eje X.



Figura 2.17 Cabeceo [23].

### **Yaw**

*Yaw* o guiñada se refiere al giro que realiza la cabeza de forma horizontal (véase figura 2.18). Este giro es realizado respecto al eje Y.

---

<sup>4</sup> **Giroscopio:** Instrumento que detecta rotaciones y mide la velocidad angular para determinar qué tan suave o brusco fue el movimiento [21].

<sup>5</sup> **Acelerómetro:** Instrumento que mide la aceleración de un cuerpo.

<sup>6</sup> **Magnetómetro:** Instrumento que mide la fuerza, y en algunos casos, la dirección de los campos magnéticos [22].





Figura 2.18 Guiñada [23].

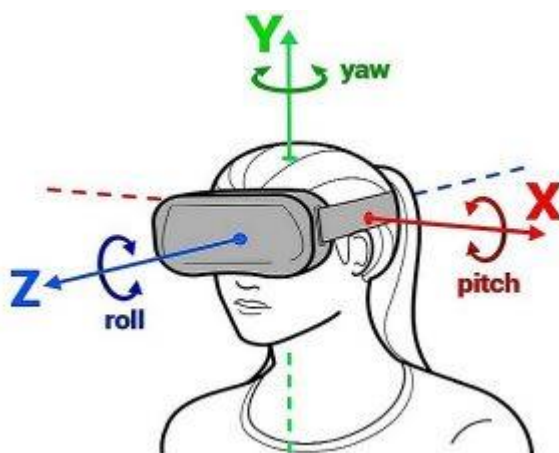


Figura 2.19 Representación gráfica de los tres grados de libertad [20].

### 2.3.4 Motion tracking

El seguimiento de movimiento o *motion tracking* es un elemento muy importante en la realidad virtual; sin esto, el usuario se quedaría paralizado, incapaz de moverse en el mundo virtual. Por lo tanto, realizar movimientos en un mundo virtual, es esencial para una experiencia inmersiva de primera clase.

Existen algunas opciones para seguir el movimiento, sin embargo, éstas se dividen en dos grupos: seguimiento óptico y seguimiento no óptico.

El seguimiento óptico utiliza sensores de imagen para rastrear el movimiento del cuerpo. El seguimiento no óptico utiliza sensores que pueden instalarse en el hardware o pegarse en el cuerpo; los más avanzados hacen uso de ondas de sonido o de campos magnéticos [24].

#### Métodos ópticos

Generalmente usan cámaras y la persona a la cual se le realizará el seguimiento de su movimiento, tiene marcadores ópticos, ya sea en su cuerpo, en el visor o en el controlador. Una cámara permite captar la profundidad cuando ve un marcador y puede



asignarlo a un espacio 3D; se puede generar una mayor precisión cuando se hace uso de más de una cámara, colocándolas en ángulos conocidos, por ejemplo, cada 45 grados se puede colocar una cámara [24].

Los marcadores emiten luz y de esta manera las cámaras son capaces de captarlos. Pueden ser de diferentes colores e incluso, pueden parpadear en sincronización con el sistema de captura. Un claro ejemplo de estos marcadores es el dispositivo *PlayStation Move*, que cuenta con una luz al extremo del control y cambia de color dependiendo de los jugadores que se encuentren en ese momento. El seguimiento se realiza mediante una cámara conectada a la consola *PlayStation*.



Figura 2.20 Mando de movimiento *PlayStation Move* [25].

### Métodos no ópticos

Hacen uso de sensores electromecánicos, tales como giroscopio, acelerómetro y magnetómetro. Los equipos de realidad virtual actuales hacen uso de estos métodos, ya que no requieren de un espacio amplio para su uso, además de ser más económicos y asequibles para la población [24].

Existen diferentes dispositivos que utilizan esta tecnología, y en su mayoría son controladores que cuentan con *joysticks* y botones, sin embargo, algunas compañías han implementado *touchpads* para facilitar el manejo, tal es el caso del controlador *Oculus Go* (véase figura 2.21).



Figura 2.21 Controlador *Oculus Go* [26].

En ocasiones, al utilizar este tipo de dispositivos se pierde la sensación de inmersión, razón por la cual en el mundo virtual se soluciona esto. Se representa de manera gráfica la mano o las manos del usuario (dependiendo del número de controladores que esté manejando en ese momento) y se sigue el movimiento del mundo real en el mundo virtual.

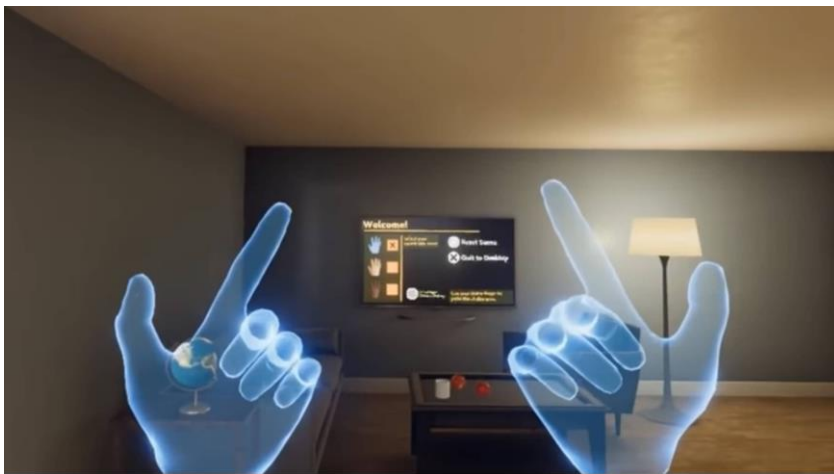


Figura 2.22 Ejemplo de manos virtuales [27].

### 2.3.5 HMD actuales

#### **HTC VIVE**

Uno de los visores con mayor avance tecnológico en los últimos años, ya que permite un seguimiento preciso del movimiento del usuario, incluyendo el desplazamiento de las manos.

El visor cuenta con 32 sensores, los cuales permiten detectar con precisión la posición del usuario y el movimiento de cabeza. Además, cuenta con dos mandos que incluyen 12 sensores cada uno, permitiendo recrear con exactitud los movimientos de la mano en el mundo virtual. Para captar el desplazamiento del usuario, se hace uso de dos estaciones láser, logrando rastrear el movimiento dentro de un área de 4.5 m x 4.5 m [28].



Figura 2.23 Elementos que conforman el *HTC VIVE* [29].

### ***Google Cardboard***

Uno de los visores de realidad virtual más económicos del mercado, las *Google Cardboard* están hechas de cartón y de lentes de distancia focal, además hacen uso de un *smartphone* para poder ver las aplicaciones en realidad virtual [30].

Es necesario descargar un software específico en un *smartphone* para poder emparejarlo con el visor. Una vez hecho esto, se podrá acceder a una gran variedad de aplicaciones, como juegos o videos en 360°.

Una de las grandes ventajas que tiene este visor, es que no es necesario comprarlo, ya que la empresa *Google* liberó las instrucciones para fabricar las *Google Cardboard* en casa.

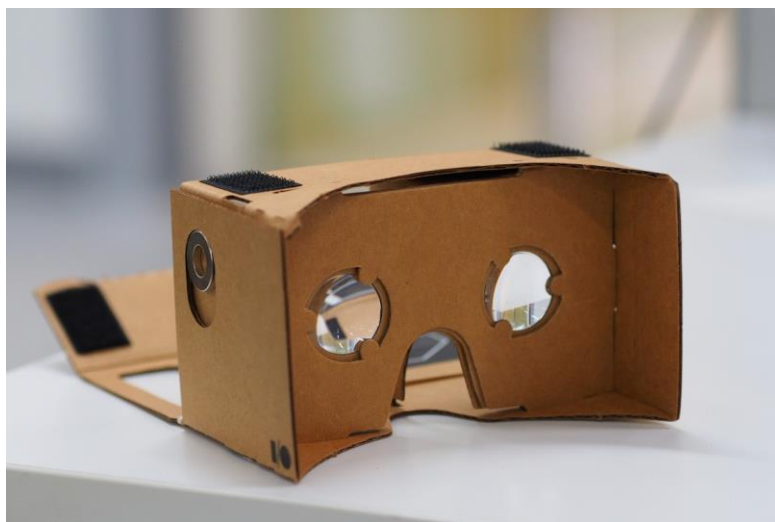


Figura 2.24 *Google Cardboard* [30].

## **PlayStation VR**

Una de las grandes apuestas de la empresa *Sony*, sus visores de realidad virtual ofrecen la mayor comodidad en el mercado, gracias a la colocación estratégica de sus almohadillas. Ofrece un sistema de ajuste muy completo, lo que permite que cualquier persona lo pueda usar, además el visor está diseñado para distribuir el peso y que no se presente una sensación de pesadez, logrando que el uso del visor pueda ser durante un tiempo prolongado (superior a 40 minutos) [31].

Uno de los aspectos a tomar en cuenta, es el uso de una cámara que se conecta a la consola *PlayStation 4*, lo que permite obtener la distancia a la que se encuentra el usuario y, si se está utilizando el sistema *PlayStation Move*, detecta el mando de movimiento; logrando una mayor inmersión.



Figura 2.25 *PlayStation VR* [31].

## **Oculus Go**

Presenta la mayor innovación en el mercado, ya que es el primer visor de realidad virtual portátil, permitiendo su uso en cualquier lugar sin la necesidad de una consola de videojuegos o una computadora, además no necesita el uso de ningún cable para su funcionamiento.

Por otra parte, es el primer visor pensado para las personas que usan lentes de armazón, ya que gracias a su espaciador ergonómico, les permite a los usuarios usar el visor con sus lentes puestos [32].

Una de las principales características con las que cuenta el visor, son los altavoces que tiene integrados, por lo que no hace falta usar auriculares. Ofrecen una gran calidad de audio gracias a su sonido cinematográfico, envolvente y nítido [32].

El controlador es ligero, ergonómico e intuitivo; está conformado por un *touchpad* o panel táctil en la parte superior con el que se puede desplazar dentro de la interfaz, además

puede funcionar como cuatro botones dependiendo de la posición en donde se apriete (arriba, abajo, izquierda o derecha). También cuenta con un par de botones: inicio y atrás. El botón inicio permite ir a la interfaz del *Oculus Go* mientras que el botón atrás muestra la pantalla anterior a la actual [33].



Figura 2.26 *Oculus Go* [32].

Para poner en funcionamiento el visor, es necesario descargar primero la aplicación de *Oculus*, la cual es compatible con *IOS* y *Android*; se debe crear una cuenta de *Oculus* o bien, se puede registrar el usuario mediante una cuenta de *Facebook*. Una vez hecho esto, el teléfono móvil y el visor se emparejan mediante bluetooth, lo cual permitirá configurar el visor y decidir con qué mano se utilizará el controlador. Desde la aplicación se puede acceder a la biblioteca de juegos, videos y aplicaciones del *Oculus Go* o comprar una aplicación mediante la *Oculus Store*. Un detalle importante a tomar en cuenta, es que el teléfono móvil y el visor deben estar conectados a la misma red WiFi, con la finalidad de que se puedan utilizar todas las opciones con las que cuenta la aplicación de *Oculus* [33].

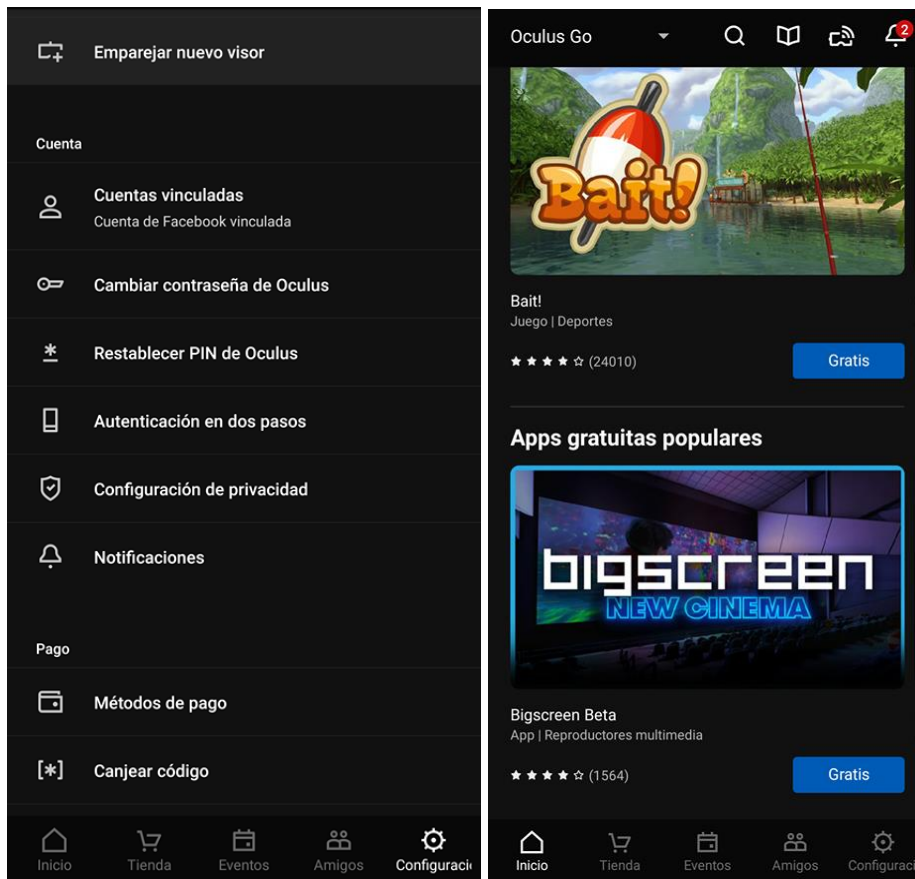


Figura 2.27 Configuración y tienda de la aplicación *Oculus*.

El apartado gráfico es claramente inferior a otros visores que existen en el mercado, es una de las desventajas que tiene al ser portátil, ya que todo el procesamiento de información lo hace el mismo visor y no una computadora, sin embargo, actualmente no existe un visor de realidad virtual portátil igual de potente que el *Oculus Go*, por lo que vale la pena destacar sus características [33]:

- CPU/RAM: *Oculus Go* tiene un microprocesador *SnapDragon 821*, similar a un móvil de gama media-alta, razón por la cual su sistema operativo es una versión modificada de *Android* para integrarse correctamente con el hardware. La capacidad de la memoria RAM es de 3 GB, lo suficiente para ejecutar adecuadamente todas las aplicaciones.
- Almacenamiento: Es la memoria donde se guardan todas las aplicaciones; existen 2 modelos, de 32 GB y 64 GB.
- Peso: 468 gramos.
- Resolución de pantalla: Es el número de *pixeles*<sup>7</sup> que puede ser mostrado en pantalla, en este caso, *Oculus Go* tiene una resolución Full HD o alta definición completa, es decir, 2560 x 1440 *pixeles*. Sin embargo, en realidad hay dos pantallas, una para cada ojo, con una resolución de 1280 x 1440 *pixeles* en cada pantalla.

<sup>7</sup> **Pixeles:** Plural de la palabra pixel. Un pixel es la unidad mínima de color homogéneo que conforma una imagen digital, ya sea una imagen o un video [34].

- Batería: Cuando se utiliza únicamente para ver videos, tiene una duración de 2.5 horas. Por otra parte, cuando se utiliza para juegos, tiene una duración máxima de 2 horas.

#### 2.3.5.1 HMD a utilizar

De entre los HMD que existen se utilizará el *Oculus Go* pues es con el que ya se cuenta actualmente (a partir de este punto se le conocerá al HMD como visor). Esta decisión delimita ciertas características técnicas: el peso que puede tener la aplicación debido a la memoria con la que cuenta el *Oculus Go*, los efectos visuales que es capaz de generar su CPU y la resolución máxima de las texturas que soporta su *display*.

## 2.4 Programas para el desarrollo de entornos virtuales

El proceso de diseño de una aplicación de realidad virtual no es una tarea fácil, especialmente si es hecha directamente en una interfaz de programación de aplicaciones (API por sus siglas en inglés). Esta situación requiere que el desarrollador sea capaz de manejar todos los recursos necesarios para hacer que la aplicación funcione según lo previsto [15].

Para hacer las cosas más fáciles, a principios de los noventas el concepto de “motor gráfico” apareció en la industria de los videojuegos. En términos generales un motor gráfico es un *middleware*, es decir, un grupo de funciones de biblioteca las cuales resumen la mayoría de los detalles de implementación de bajo nivel proveyendo al desarrollador con una interfaz de programación de alto nivel [15].

### 2.4.1 Unity

*Unity* es un motor gráfico el cual proporciona un acervo de componentes y herramientas necesarias para el desarrollo de videojuegos, tales como simulaciones de físicas, efectos de iluminación, entre otros. Goza de un poderoso editor de niveles, existe una amplia comunidad que brinda soporte sobre cualquier tema y posee una documentación muy extensa. Además, es posible adquirir una licencia gratuita la cual te permite migrar tus proyectos a plataformas como *Windows*, *Mac OS* y *Android* [35].





Figura 2.28 Editor de niveles de *Unity* [36].

### 2.4.1.1 Programación en *Unity*

El lenguaje utilizado para programar en *Unity* es *C#*. Al ser un lenguaje de programación muy conocido es fácil buscar y encontrar documentación sobre él. Sin embargo, es importante conocer los conceptos y particularidades con las que se tiene que trabajar en *Unity*.

#### **GameObjects**

Cualquier tipo de contenido en *Unity* empieza con un *GameObject* u objeto del juego. Cualquier objeto que esté presente dentro del juego es un *GameObject*. Personajes, luces, efectos especiales, accesorios, sonido, etc., es decir, todo [37].

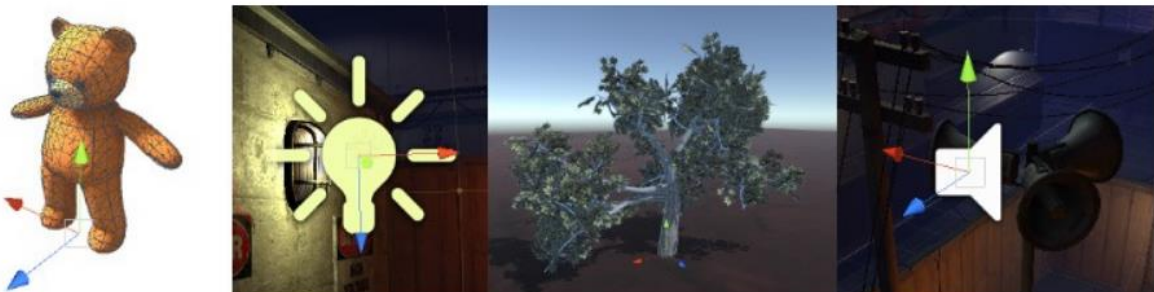


Figura 2.29 Ejemplos de *GameObjects* en *Unity* [37].



## Componentes

Los componentes definen y controlan el comportamiento de los *GameObjects* a los cuales están conectados. Un ejemplo simple sería la creación de una luz que indica conectar un **Componente de luz** a un *GameObject* [37].

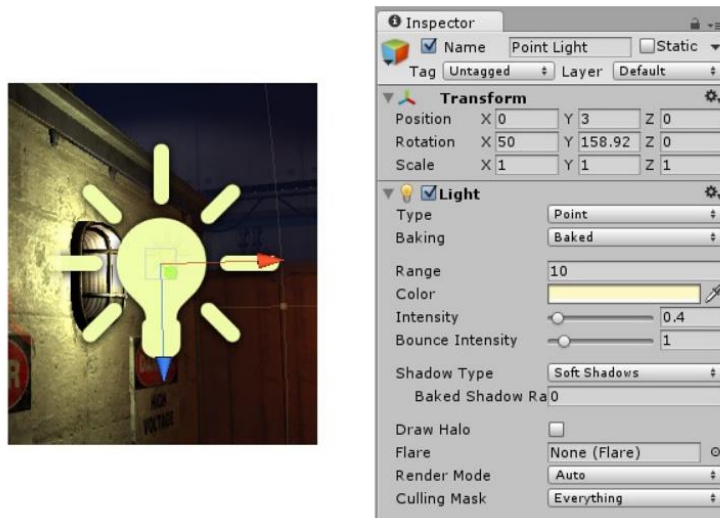


Figura 2.30 Componente de luz en el editor de niveles de *Unity* (izquierda) y la ventana Inspector de dicho componente (derecha) [37].

## Variables

Los **Componentes** pueden tener cualquier número de propiedades editables que pueden retocarse a través de la ventana **Inspector** en el editor o a través de *scripts*<sup>8</sup>. En el ejemplo anterior, algunas propiedades del **Componente de luz** son rango, color e intensidad [37].

Los **Componentes** incorporados en *Unity* son muy versátiles, pero para lograr que se comporten bajo cierta lógica establecida es necesario ir más allá. Utilizando *scripts* (programados en *C#*) es posible implementar una lógica particular y un determinado comportamiento del juego, simplemente aplicándolo a los **GameObjects**. Esto permite hacer muchas cosas: iniciar eventos en el juego, verificar si hay colisiones, aplicar la física del juego, responder a la información de entrada del usuario y mucho más [37].



Figura 2.31 Logotipo de *Unity* [37].

<sup>8</sup> **Script:** Tipo de lenguaje para programar computadoras [38].

## 2.4.2 Unreal Engine 4

Es un motor gráfico gratuito desarrollado por la empresa *Epic Games* que cuenta con un gran conjunto de herramientas para el desarrollo de aplicaciones empresariales, experiencias cinematográficas y juegos de alta calidad para diferentes plataformas. Una de sus grandes características es el hecho de que los desarrolladores pueden probar de manera rápida sus ideas sin realizar una línea de código [39].

*Unreal Engine 4* tiene unas excelentes capacidades gráficas, destacando la función avanzada de la iluminación dinámica y un sistema de partículas capaz de alcanzar hasta un millón de partículas dentro del nivel (entorno virtual) [39].



Figura 2.32 Logotipo de *Unreal Engine* [39].

### 2.4.2.1 Programación en Unreal Engine 4

El lenguaje utilizado para programar en *Unreal Engine 4* es C++, sin embargo, existe la posibilidad de programar por bloques mediante *Blueprints*; esta opción permite crear rápidamente proyectos sin desarrollar una sola línea de código. Una de las ventajas de utilizar los *Blueprints* es el hecho de que se puede visualizar el flujo del juego en cualquier momento. Por otra parte, se pueden modificar de manera sencilla la interfaz que tendrá el usuario dentro del juego, ajustar controladores de entrada y mucho más [39].

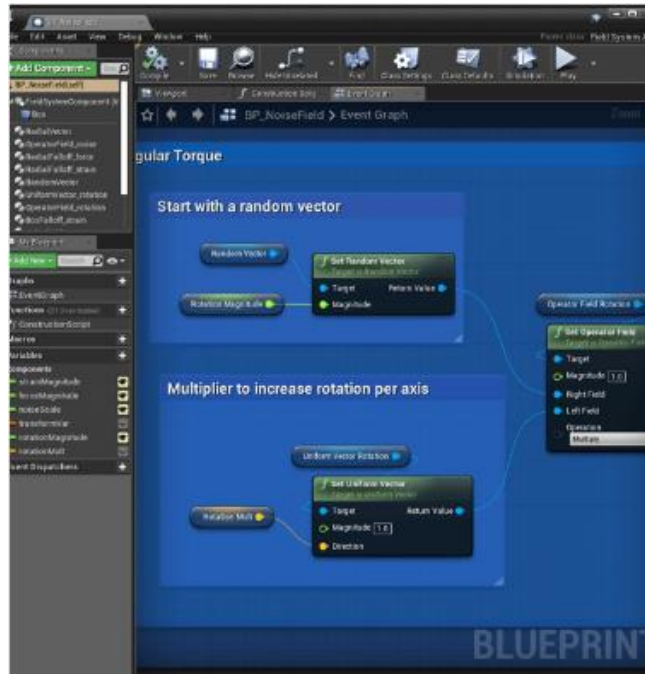


Figura 2.33 Ejemplo de programación por bloques mediante *Blueprints* [39].

La programación no se limita únicamente al lenguaje C++, en los últimos años se han realizado actualizaciones en el motor gráfico para que los usuarios puedan crear scripts usando lenguaje *Python*, derivando en la optimización de proyectos.

*Unreal Engine 4* se caracteriza por ser un motor gráfico que permite el desarrollo de aplicaciones o videojuegos en tecnologías emergentes, tal es el caso de la realidad aumentada y de la realidad virtual. Hablando específicamente de realidad virtual, uno de los grandes avances que ha tenido este motor gráfico es la posibilidad de programar en tiempo real utilizando un visor de realidad virtual y sus respectivos controladores, con esta herramienta “lo que ve el usuario es lo que obtiene”, permitiendo mayor facilidad en el desarrollo de entornos virtuales [39].



Figura 2.34 Ejemplo de desarrollo de un entorno virtual mediante realidad virtual [39].

### 2.4.3 Comparación entre Unity y Unreal Engine

#### **Unity**

##### **Ventajas**

- Es compatible con diversos programas de modelado.
- Ofrece soporte para varios modelos de visores de realidad virtual.
- Mediante su tienda oficial, permite a los desarrolladores acceder a una gran cantidad de contenido de terceros de manera gratuita.
- Al ser un motor gráfico muy popular, existe una amplia documentación y tutoriales publicados por desarrolladores.
- Se caracteriza por ser más sencillo crear juegos para *smartphones* y juegos en 2D.

##### **Desventajas**

- La versión gratuita no contiene todas las herramientas de personalización y soporte.
- La calidad gráfica que se presenta en los videojuegos, es menor comparada con otros motores gráficos.
- No incluye plantillas para creación de un proyecto.

#### **Unreal Engine 4**

##### **Ventajas**

- Tiene compatibilidad con muchas plataformas de modelado.
- Ofrece de inicio soporte para el desarrollo de realidad virtual en varias plataformas.
- El software es completamente gratuito.
- Su editor de materiales es muy avanzado, lo que permite crear diferentes tipos de materiales
- Posee un mercado en el cual se pueden comprar o vender todo tipo de elementos como modelos, materiales, animaciones, etc.

##### **Desventajas**

- Es un poco difícil para programadores expertos adaptarse a este tipo de programación por bloques.
- La gama de herramientas es tan grande que cuesta trabajo convertirse en experto en ellas.

#### 2.4.4 Selección del software para el desarrollo del entorno virtual

Una vez seleccionado el visor es momento de seleccionar el software para el desarrollo del entorno virtual el cual se llamará motor gráfico a partir de este momento. Tanto *Unity* como *Unreal Engine 4* son motores gráficos con muchas herramientas disponibles, compatibles con *Android* y por ende con el *Oculus Go*. Al ser ambos motores gráficos una buena opción, el *expertise* se vuelve nuevamente el factor clave para seleccionar uno de los programas. Es por eso que se opta por utilizar *Unreal Engine 4*.

## 3. DESARROLLO ITERACIÓN 1

En este capítulo se presenta el desarrollo de la primera iteración del software, además de realizar pruebas para validar su funcionamiento.

### 3.1 Modelado de elementos hidráulicos

La selección de los elementos hidráulicos a modelar se hizo con base en el sistema hidráulico que se encuentra en el laboratorio de la Facultad de Química el cual se muestra a continuación:



Figura 3.1 Sistema hidráulico del laboratorio de la Facultad de Química.

A partir de esto, se decidió modelar los elementos más básicos para poder simular la práctica, se modelaron tuberías rectas, bomba de agua y placas de orificios.

### 3.1.1 Tubería recta

El modelo de la tubería recta se basó en hacer una abstracción del objeto, no se buscó que fuera enteramente realista, sino que fuera fácil de reconocer:

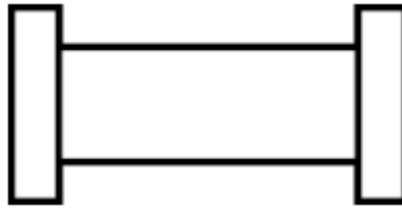


Figura 3.2 Abstracción de la tubería recta.

A partir de esto, el modelo guarda la silueta de la abstracción y es hueco.

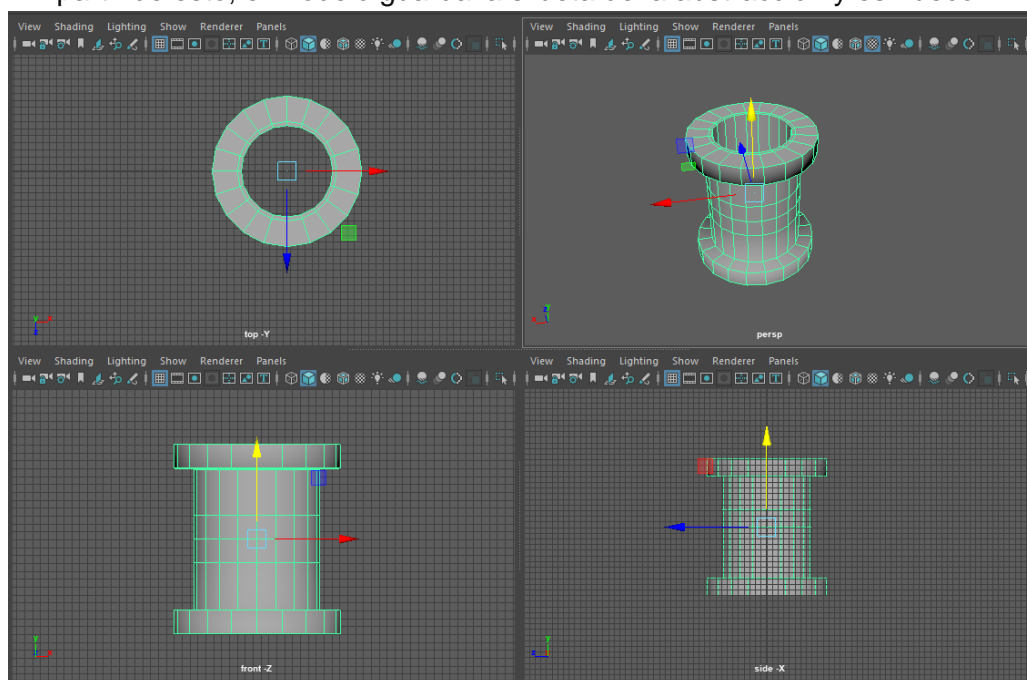


Figura 3.3 Modelo de tubería recta.

El modelo está compuesto por 624 triángulos. Esto es importante pues a mayor número de polígonos, incrementa la carga de procesamiento para el dispositivo de realidad virtual.



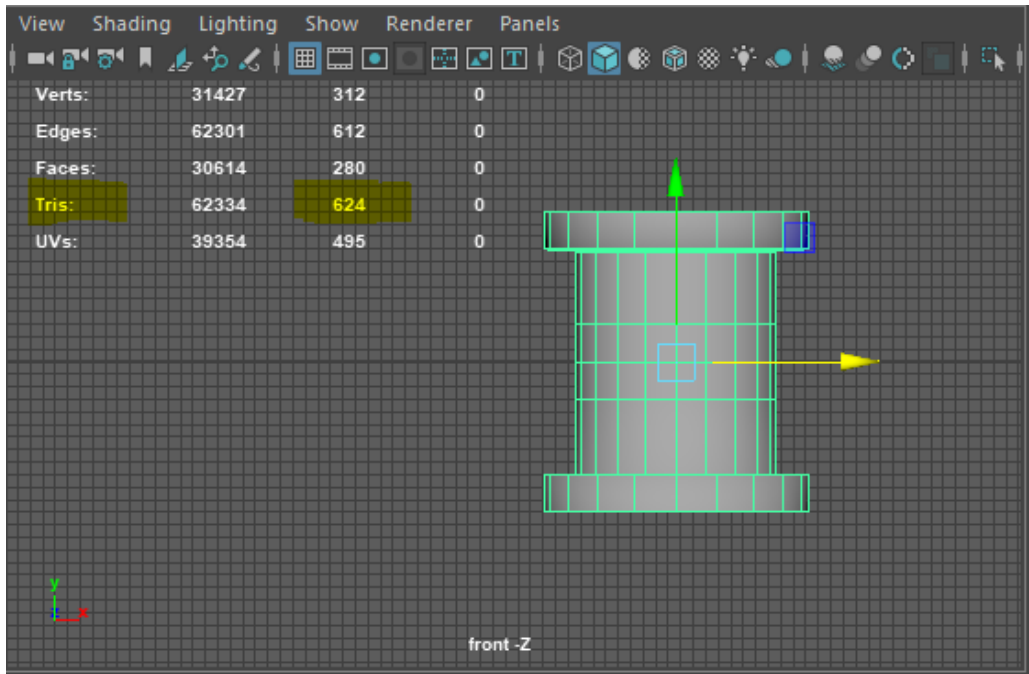


Figura 3.4 Número de triángulos de la tubería recta.

### 3.1.2 Placas de orificio

El siguiente elemento del que se hizo el modelo fueron las placas de orificios. Se escogieron 2 placas de orificio, la placa concéntrica y la segmentada puesto que son las más utilizadas para este tipo de experimentos.

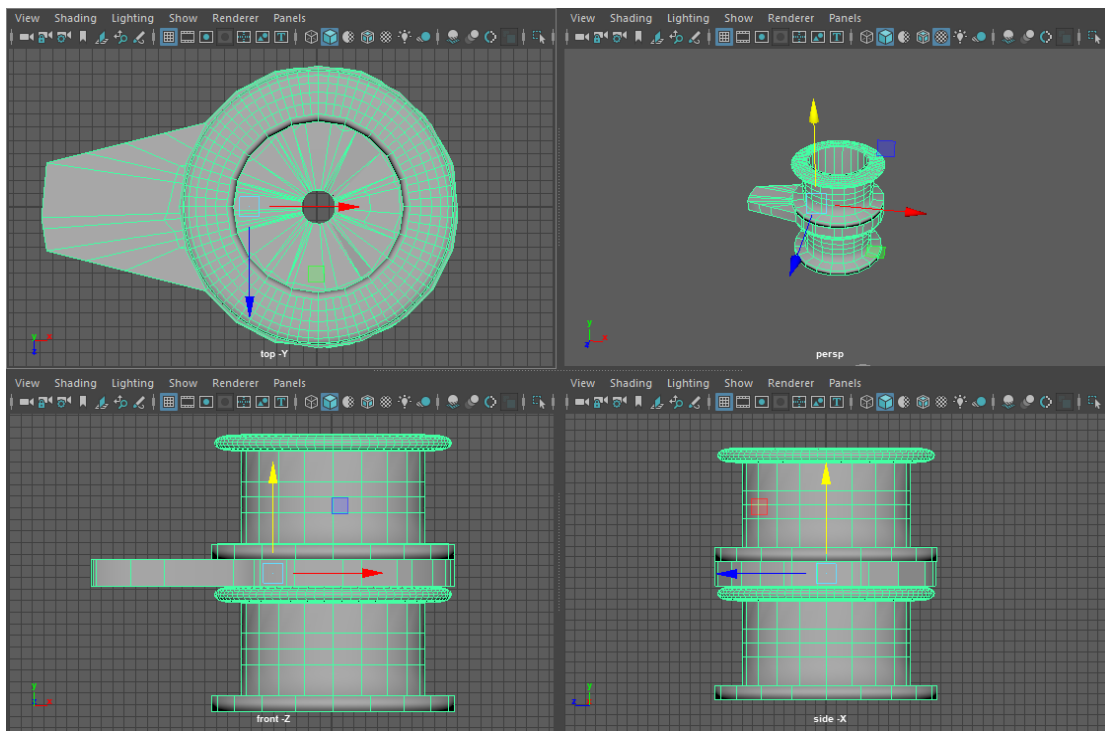


Figura 3.5 Modelo de la placa de orificio concéntrica.



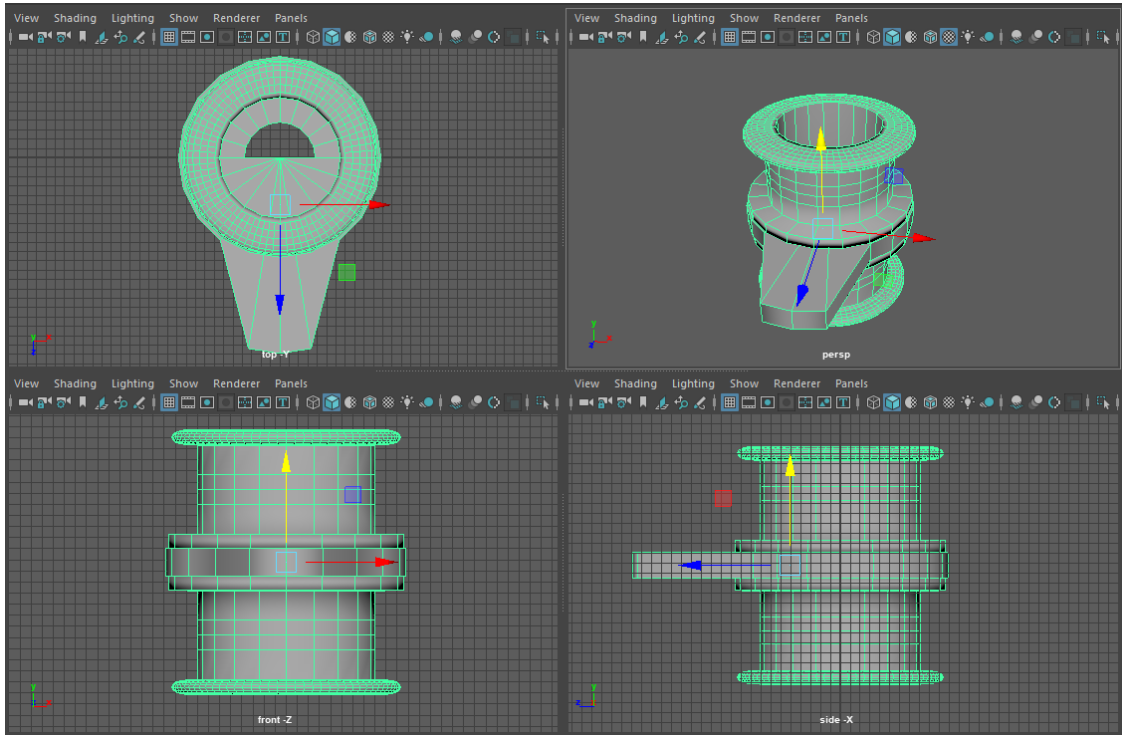


Figura 3.6 Modelo de la placa de orificio segmentada.

El modelo de la placa de orificio concéntrica tiene 5750 triángulos, mientras que el de la placa segmentada tiene 5380 triángulos.

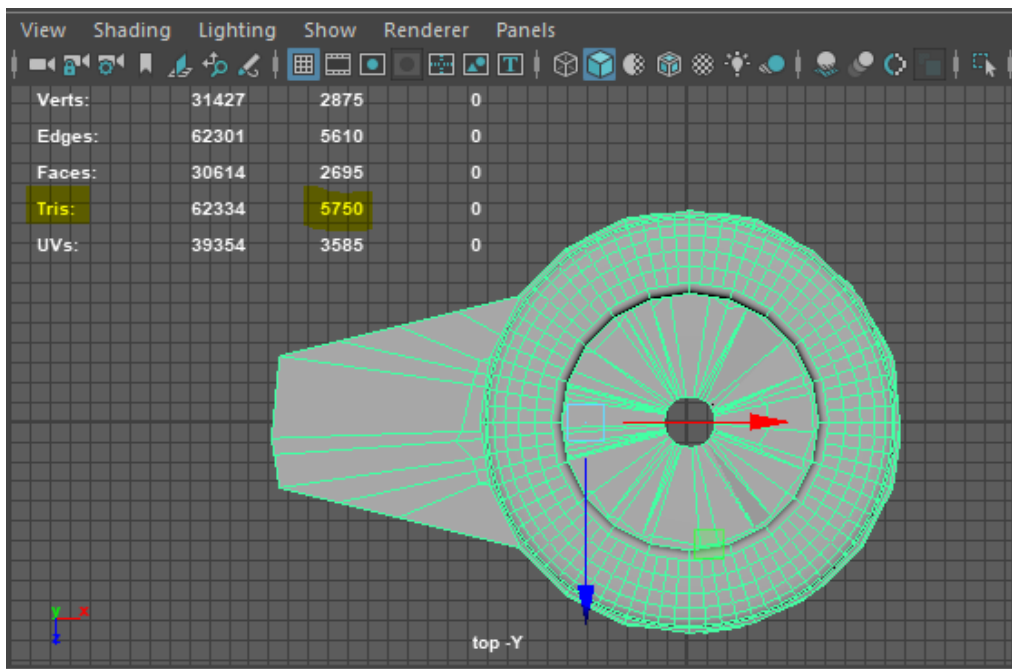


Figura 3.7 Número de triángulos de la placa de orificio concéntrica.

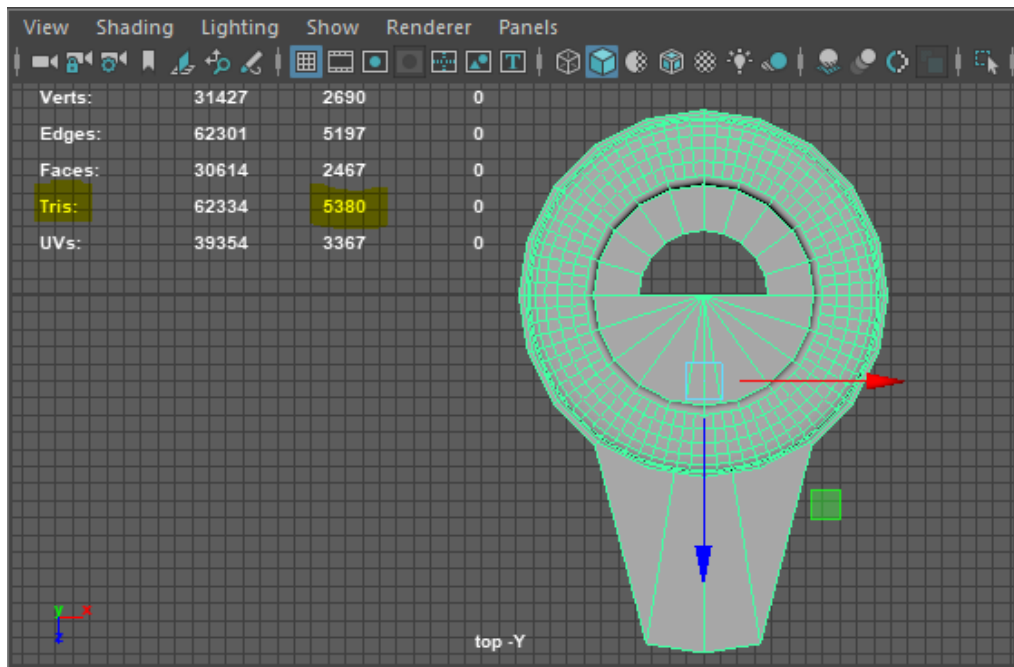


Figura 3.8 Número de triángulos de la placa de orificio segmentada.

### 3.1.3 Bomba de agua

Por último, la bomba de agua se modeló con base en una bomba hidráulica genérica, aunque de manera abstracta pero inteligible:



Figura 3.9 Motobomba *munich* centrífuga [40].

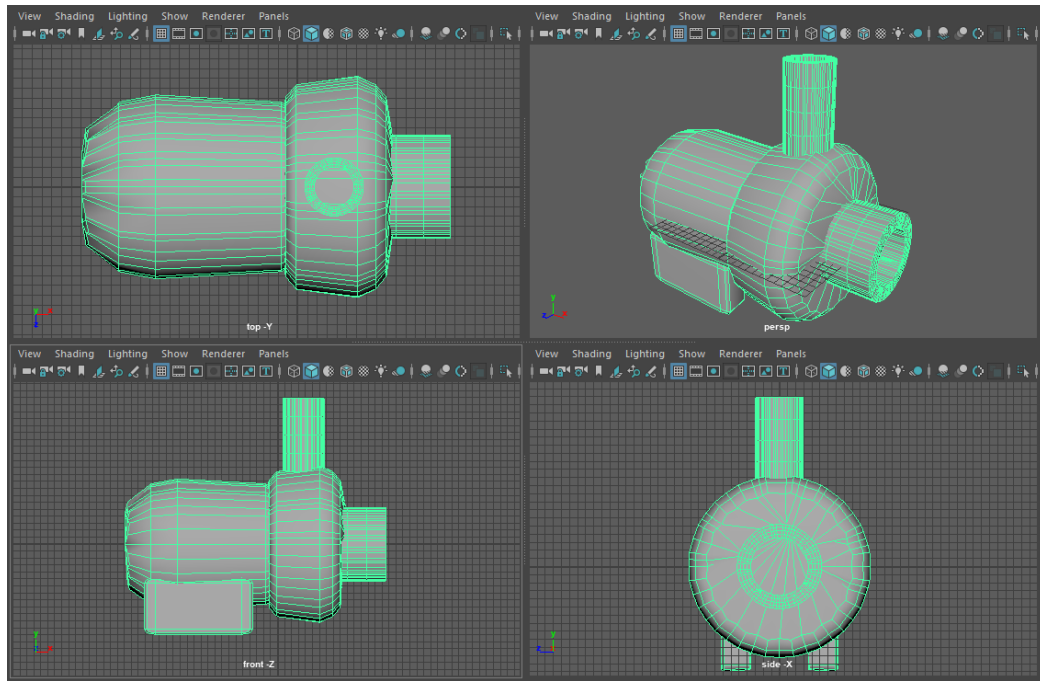


Figura 3.10 Modelo de la bomba de agua.

El modelo de la bomba cuenta con 5404 triángulos.

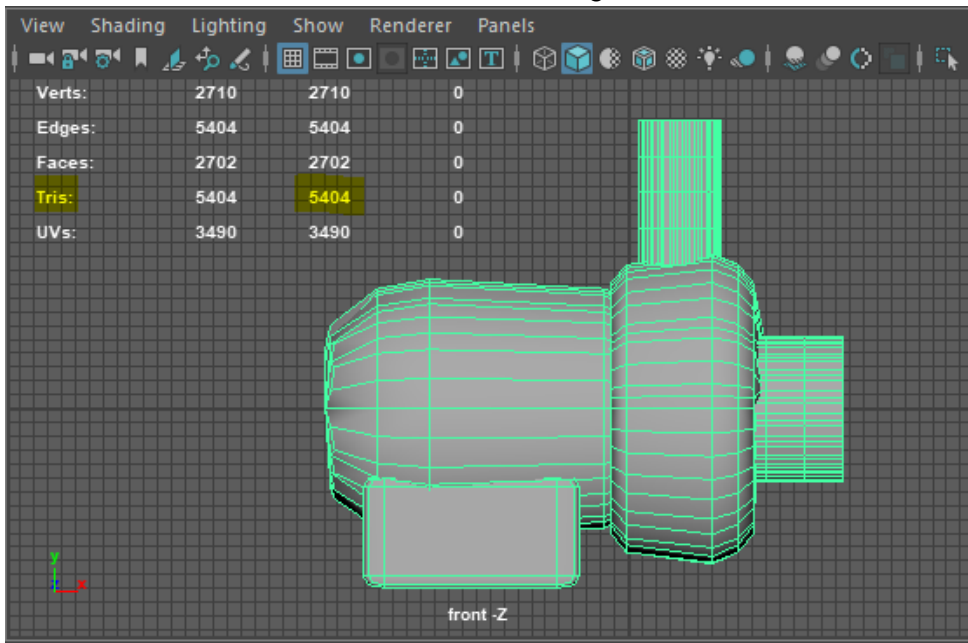


Figura 3.11 Número de triángulos de la bomba de agua.

## 3.2 Componentes de Unreal Engine 4

### 3.2.1 Terminología de Unreal Engine 4

Cuando se trabaja en *Unreal Engine 4* se utilizan términos con los que no se está familiarizado, es por esta razón que es importante comprender el significado de cada término para saber con exactitud las herramientas con las que se cuenta durante el desarrollo de un videojuego.

#### Objetos

Son los bloques de construcción básicos de *Unreal Engine* y contienen muchas de las funciones esenciales para que los activos o elementos dentro del juego se ejecuten correctamente [41].

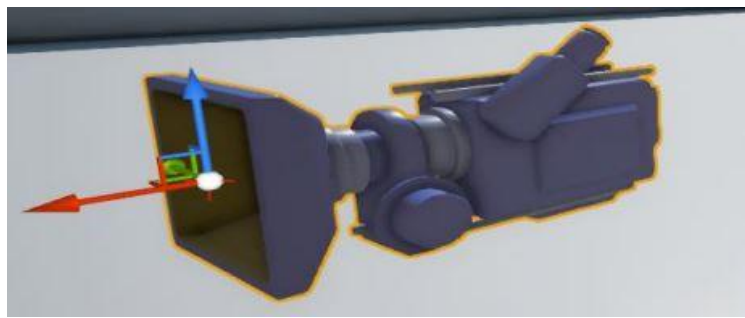


Figura 3.12 Ejemplo de Objeto en *Unreal Engine* [41].

#### Clases

Una clase define el comportamiento y las propiedades de un **Actor** u **Objeto** utilizado durante el desarrollo de un juego. Las clases son jerárquicas, lo que significa que una clase hereda información de las **Clases Padre** y pasa esa información a las **Clases Hijo** [41].

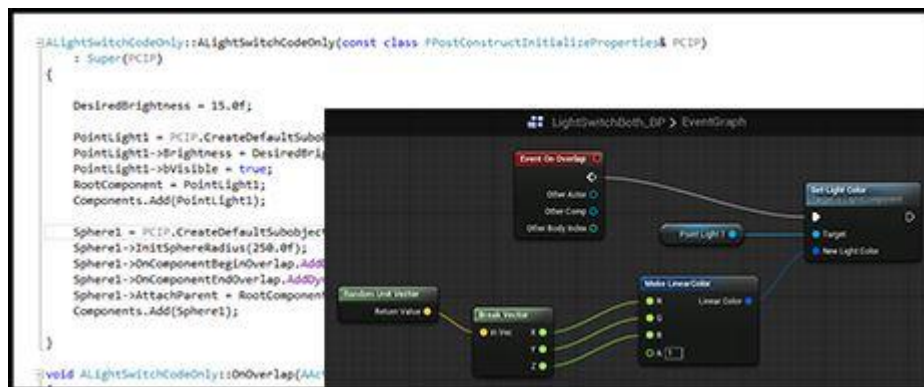


Figura 3.13 Clase creada en lenguaje C++ y en Blueprints [41].

## Actores

Un Actor es cualquier objeto que se puede colocar en un **Nivel**. Los Actores soportan transformaciones 3D tales como rotación, traslación y escalamiento. Tienen la característica de que pueden ser creados y destruidos a través del código del juego [41].

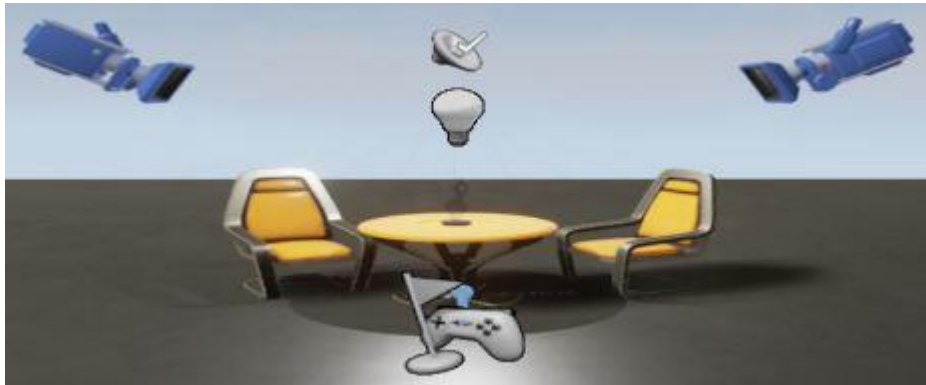


Figura 3.14 Ejemplos de Actores [41].

## Componentes

Un Componente es una pieza funcional que se puede agregar a un **Actor**. Los Componentes no pueden existir por sí mismos, sin embargo, cuando se agregan a un **Actor**, este tendrá acceso a cada una de las funciones proporcionadas por el Componente [41].

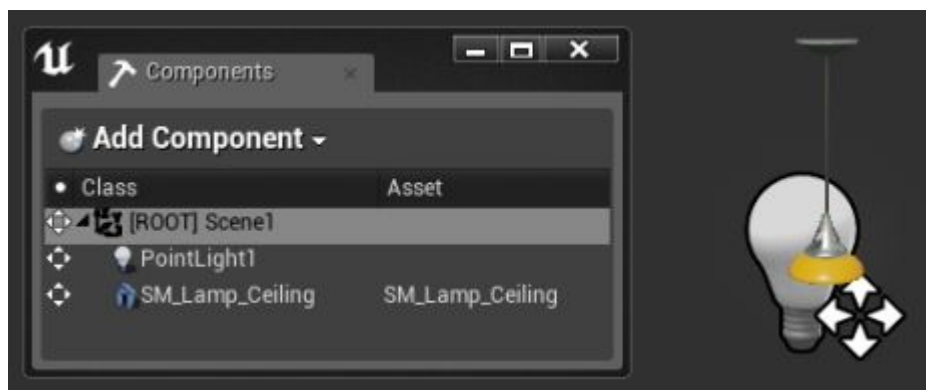


Figura 3.15 Componente de luz puntual [41].

## Peones

Los Peones son una subclase de **Actor** que sirven como personaje dentro del juego. Pueden ser controlados por el usuario o mediante inteligencia artificial, sin embargo, también existe la posibilidad de no ser controlados [41].

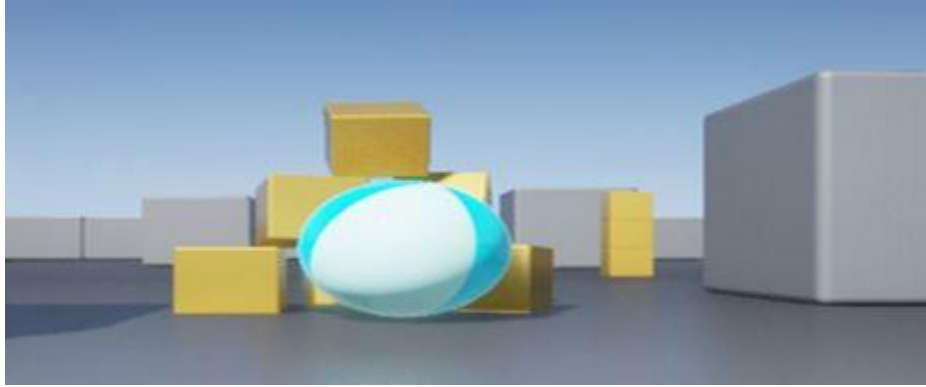


Figura 3.16 Ejemplos de peones no controlados [41].

## Personajes

Un Personaje es una subclase de **Peón** que tiene el objetivo de ser utilizado como un personaje controlado por el jugador. Incluye configuraciones que permiten que tenga movimiento bípedo y colisión [41].



Figura 3.17 Movimiento bípedo de una Personaje [41].

## Niveles

Un Nivel es un área de juego definida por el usuario. Los Niveles se crean, visualizan y modifican principalmente colocando, transformando y editando las propiedades de los **Actores** que contiene. En ocasiones, estos Niveles son referenciados como **Mapas**.



Figura 3.18 Ejemplo de un Nivel creado en *Unreal Engine* [41].

### 3.2.2 Interfaz de Unreal Engine 4

Cuando se ejecuta Unreal Engine 4 y se selecciona el tipo de proyecto a realizar, por defecto se muestra el Editor de Niveles. En términos generales, es donde se define el área de juego agregando diferentes tipos de Actores, sistemas de partículas, etc. El Editor de Niveles está conformado de la siguiente manera (véase figura 2.41) [42]:

1. **Barra de pestañas y barra de menús:** La barra de pestañas tiene una pestaña en la parte superior que proporciona el nombre del Nivel, además se pueden acoplar otro tipo de pestañas para una navegación más rápida y sencilla, similar a un navegador web. La barra de menús proporciona acceso a herramientas y comandos en general, que se utilizan al trabajar con Niveles.
2. **Barra de herramientas:** Es un grupo de comandos que proporciona acceso a herramientas y operaciones de uso común durante el desarrollo del juego.
3. **Modos:** Contiene una selección de varios elementos básicos para la construcción de niveles, como figuras geométricas, luces, Peones, Personajes, efectos visuales, etc.
4. **Buscador de contenido:** En esta parte se encuentran todos los archivos realizados del proyecto. Se pueden crear carpetas para una mayor organización.
5. **Ventana gráfica:** Es la ventana en donde se construye el Nivel. Se pueden realizar movimientos libremente dentro de ella y ver en diferentes perspectivas el Nivel. Una de las grandes ventajas con las que cuenta la ventana gráfica, es el hecho de poder arrastrar los Actores e incluirlos de forma muy rápida en el Nivel.
6. **World Outliner:** Muestra de forma jerárquica todos los Actores que se encuentran en el Nivel, lo que permite encontrarlos y modificarlos de manera más sencilla.
7. **Detalles:** Contiene la información, utilidades y funciones específicas del Actor seleccionado en la ventana gráfica. Permite editar las propiedades del Actor de forma muy rápida.





Figura 3.19 Interfaz de *Unreal Engine 4* [42].

### 3.2.3 Blueprints

El sistema de *Blueprints* en *Unreal Engine* es un sistema de *scripting* implementado con el concepto de utilizar una interfaz basada en nodos para crear elementos de juego. Coloquialmente, los objetos definidos mediante *Blueprint* se denominan simplemente como “*Blueprints*” [43].

Este sistema es sumamente flexible y potente dado que brinda a los diseñadores la capacidad de usar virtualmente toda la gama de conceptos y herramientas que generalmente solo están disponibles para los programadores. Mediante la conexión de Nodos, Eventos, Funciones y Variables a través de cables, es posible crear elementos de juego complejos [43].

Por medio del uso de *Blueprints*, los diseñadores pueden crear prototipos, implementar o modificar virtualmente cualquier elemento de juego, entre los que destacan [43]:

- Modos de juego: Se crean las reglas del juego, las condiciones de victoria, se crea el multijugador, etc.
- Jugadores: Modificar o personalizar personajes controladores por el usuario, en donde se incluye la velocidad de movimiento, velocidad de giro, altura, etc.
- Cámaras: Crear diferentes perspectivas mediante el uso de varias cámaras y cambiar dinámicamente la cámara durante el juego.
- Entradas: Crear eventos de entrada para distintos controladores.
- Interfaz dentro del juego: Crear interfaces en 2D o 3D y que se muestren en pantalla una vez iniciado el juego.





Figura 3.20 Ejemplo de las opciones del *Blueprint* tipo Personaje [43].

### 3.2.4 Iluminación

Para poder iluminar un Nivel es necesario utilizar Actores de luz los cuales funcionan como fuentes de luz y tienen propiedades que determinan las características de la luz tales como: Qué tan brillante es la luz o de qué color es la luz [44].

También existen diferentes tipos de Actores de luz los cuales dependiendo de sus características ayudan a simular la emisión de luz provocada por el sol, un bombillo o una lámpara led [44].



Figura 3.21 Actores de luz en *Unreal Engine 4* [45].

### 3.2.5 Materiales

Un Material es un *asset*<sup>9</sup> el cual puede ser aplicado a una malla para controlar cómo se ve en el Nivel. Probablemente es más fácil pensar en un Material como una pintura que es aplicada a un objeto. Pero incluso eso puede ser engañoso, ya que un Material literalmente define el tipo de superficie del cual el objeto parece estar hecho. Se puede definir el color, qué tan brillante es, si se puede ver a través de él y mucho más [44].

<sup>9</sup> **Asset:** Cualquier cosa útil o valiosa [46].

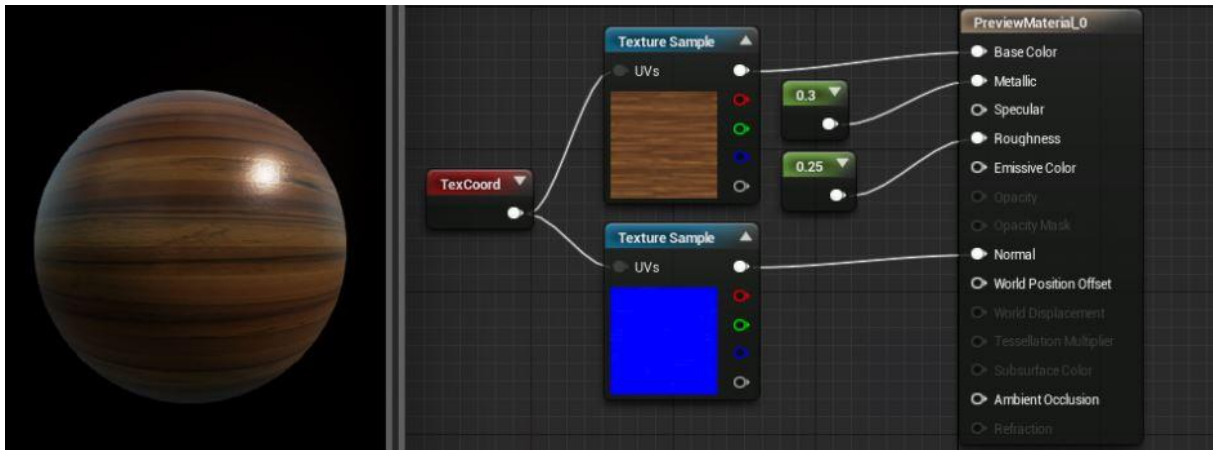


Figura 3.22 Ejemplo de un Material y el Editor de Material en *Unreal Engine 4* [47].

### 3.2.6 Sistema de partículas

*Unreal Engine 4* contiene un poderoso y robusto sistema de partículas permitiendo crear increíbles efectos visuales, desde humo, fuego, agua y más. Los Sistemas de partículas son editados vía *Cascade* (es el nombre que se utiliza para llamar al editor de Sistemas de partículas). *Cascade* ofrece retroalimentación en tiempo real, edición de efectos mediante módulos, permitiendo una creación y edición rápida y fácil [48].

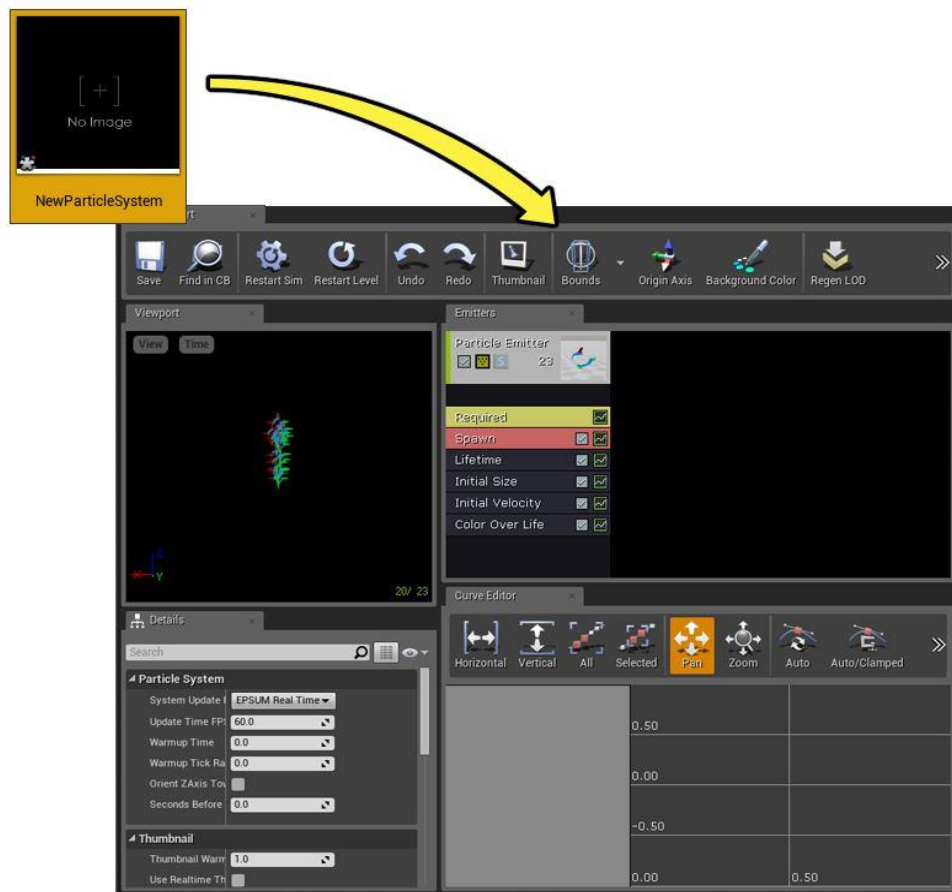


Figura 3.23 Editor de Sistemas de partículas *Cascade* [49].

### 3.2.7 Widgets

Un *Widget* es aquel pequeño programa que compone lo que se ve en la pantalla de una computadora y permite tomar acciones particulares. Un *Widget* puede ser un botón en el que puedes hacer click en él, una barra de desplazamiento, una barra de estado (Head-Up Display o HUD), casillas de verificación, entre otros [50].

Por medio del editor gráfico de *Unreal Engine* es posible crear elementos de interfaz de usuario como menús o cualquier otro gráfico relacionado con la interfaz que se desea presentar a los usuarios [51].

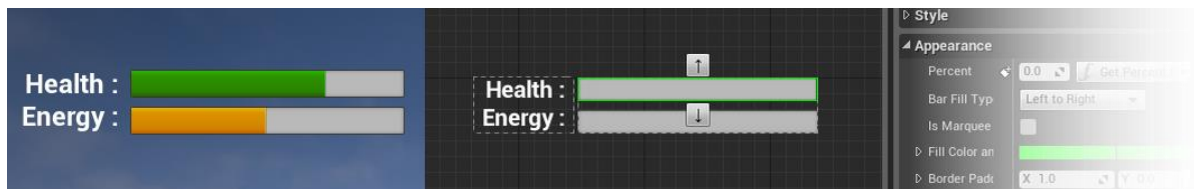


Figura 3.24 Ejemplo de barra de estado (HUD) [51].

## 3.3 Desarrollo del entorno virtual

El entorno virtual es el lugar en donde el usuario va a ser capaz de moverse y construir los sistemas hidráulicos que desee. El único límite para el jugador será el espacio, el cual estará definido en unidades del motor gráfico.

### 3.3.1 Diseño del entorno virtual

Para el diseño del entorno virtual se tomaron en cuenta varios factores con base en la experiencia adquirida en otros proyectos y en pruebas con usuarios, en total 8.

Para dichas pruebas se diseñó un Nivel el cual solamente era un cuarto grande y blanco. Además, se realizaron una serie de preguntas para saber la opinión de los usuarios sobre el aspecto del entorno, esto después de haberles explicado la finalidad del mismo. Las preguntas son las siguientes:

- 1.- ¿Qué te gusta del lugar?
- 2.- ¿Qué te parecen los colores?
- 3.- ¿Qué te parece el aspecto del techo y el piso?
- 4.- Si pudieras cambiar algo, ¿Qué sería?
- 5.- ¿En algún punto te sentiste mareado?

Las respuestas que más se repitieron fueron:

- "El cuarto se ve muy vacío"
- "El entorno se ve muy blanco"
- "La cuadrícula marea"

- “El color del techo no me gustó”
- “Me desagrada que tenga semejanza con los laboratorios de la escuela”

A partir de esto se comenzó a diseñar la composición del entorno, los colores y elementos que contendrá.

- El entorno virtual: El espacio del entorno debe ser grande, pero con un límite, para que permita al usuario construir los sistemas hidráulicos donde quiera y al mismo tiempo no sobrecargue el CPU del *Oculus Go*. Los elementos que posee fueron diseñados para que los usuarios tuvieran la sensación de estar en un lugar en donde sería lógico el uso de tuberías (una fábrica) y al mismo se asemejara a los laboratorios.

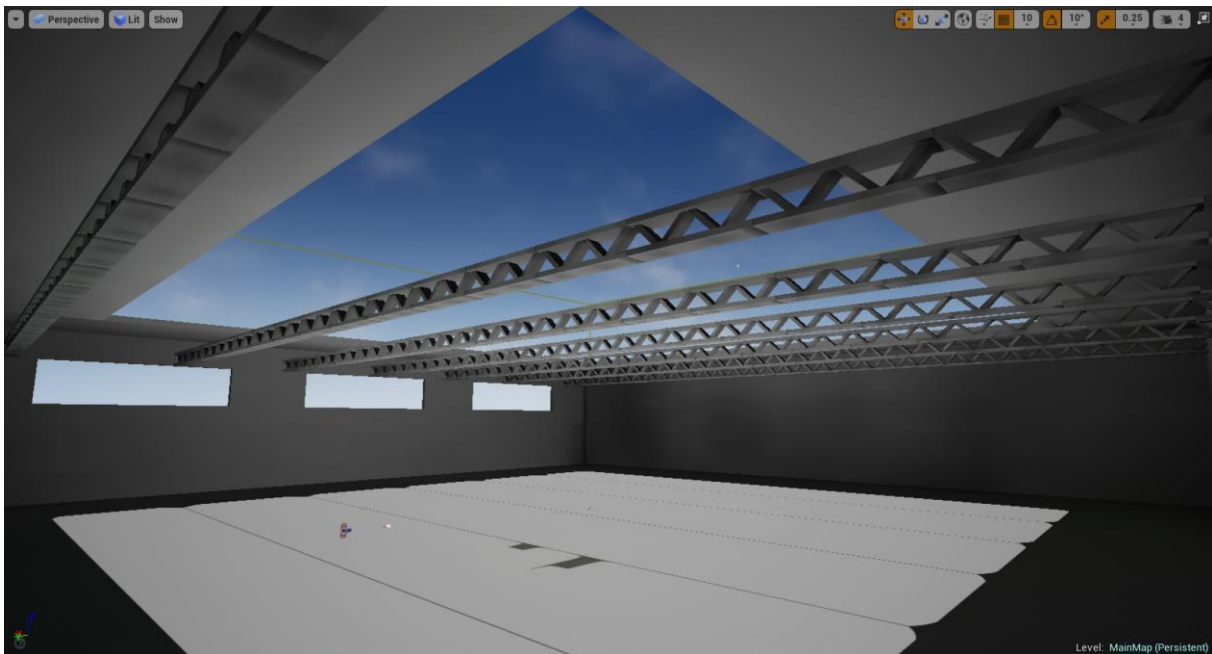


Figura 3.25 Entorno virtual.

- Texturas: Con ayuda de un editor de imágenes se dibujaron las texturas las cuales son imágenes en 2D.

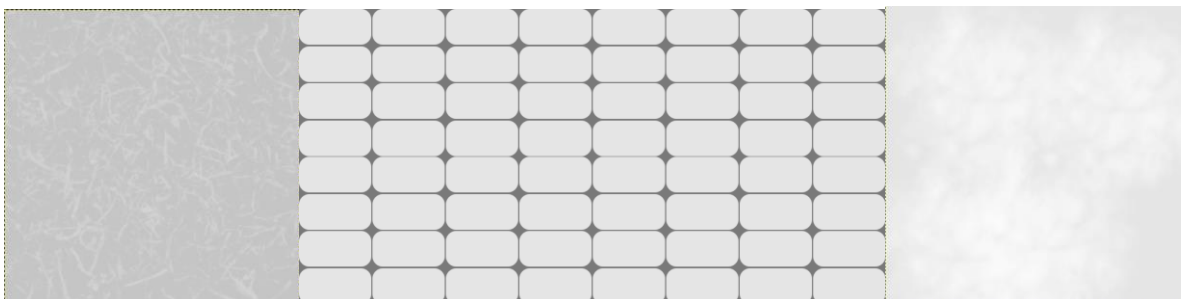


Figura 3.26 Texturas del techo, piso y pared (de izquierda a derecha).

- Materiales: Los materiales que se programaron son el material para las paredes, vigas, piso, repisas, tuberías de muestra y techo. Las texturas son imágenes en blanco y negro, con ayuda del editor de materiales se les dio cierto color y otras características. Cada material requiere de ciertos parámetros visuales para que

adquieran un determinado aspecto. Es importante resaltar que se hizo una instancia del material original, la cual es un nuevo material que hereda las propiedades del material padre y cuyos parámetros pueden ser modificados sin entrar al editor de materiales, esto se hizo para ajustar ciertos parámetros que ayudan a mejorar el renderizado de los mismos. A continuación, se presenta el material de la pared como ejemplo:

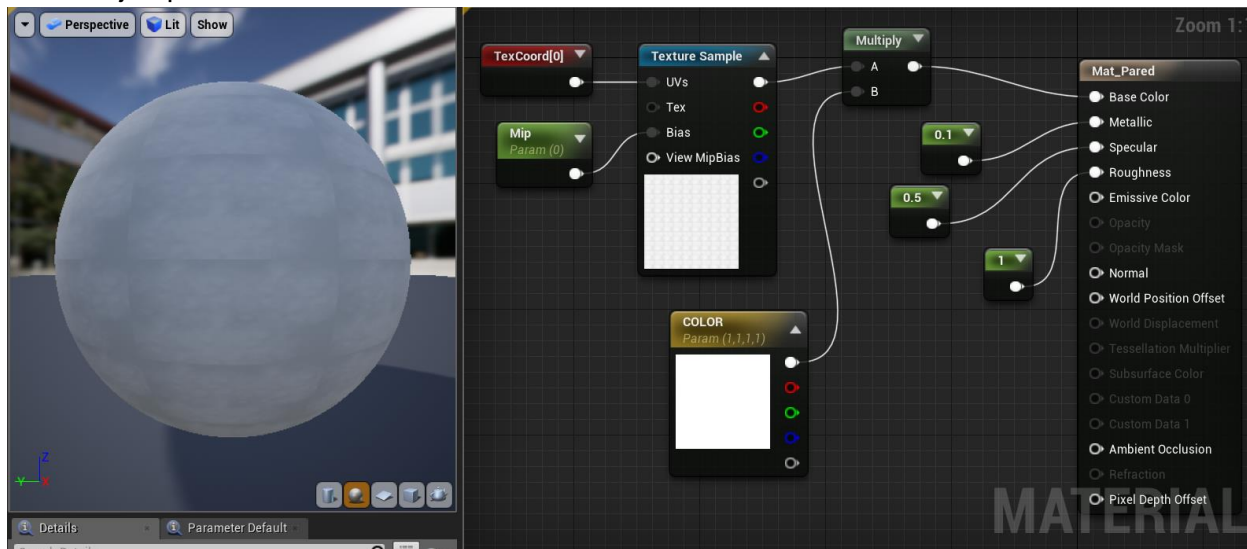


Figura 3.27 Programación del material original para la pared y su previsualización.

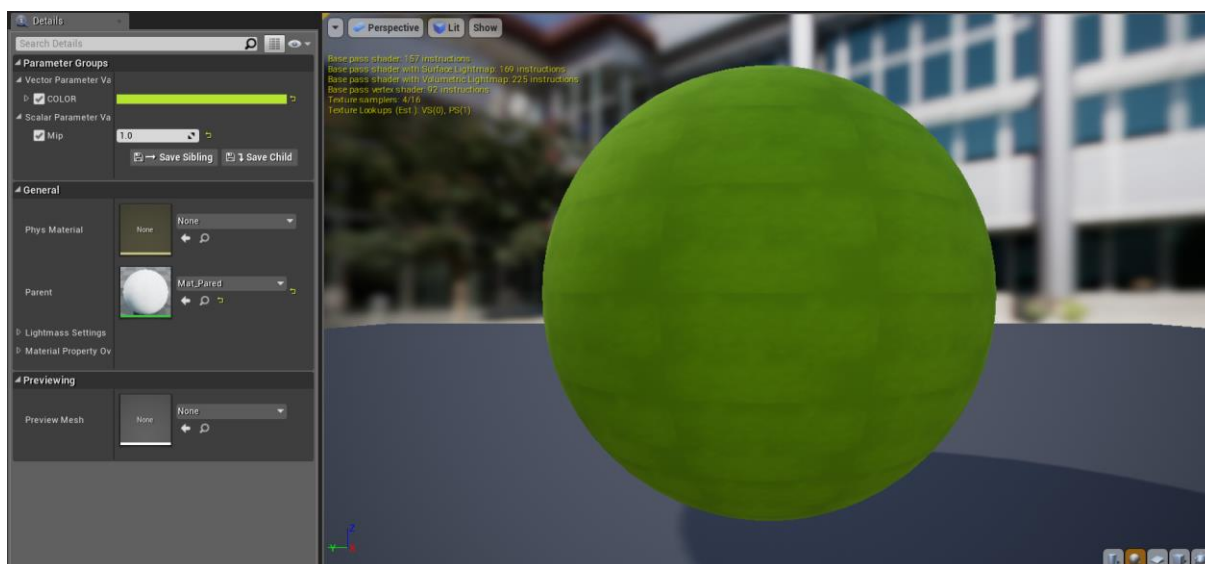


Figura 3.28 Instancia del material de la pared y su previsualización.



De forma similar se crearon los demás materiales quedando de la siguiente manera:

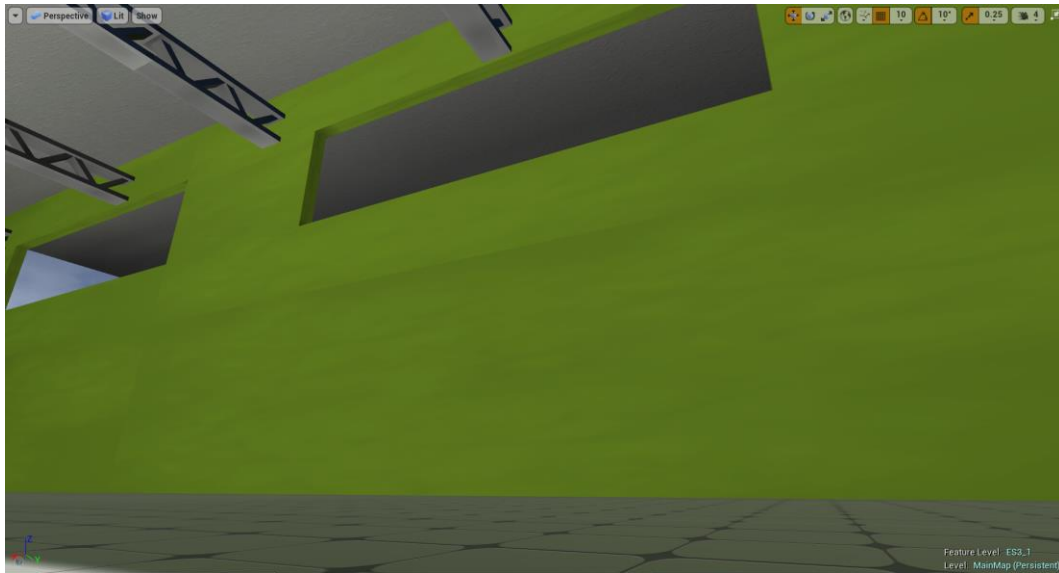


Figura 3.29 Material aplicado a la pared.

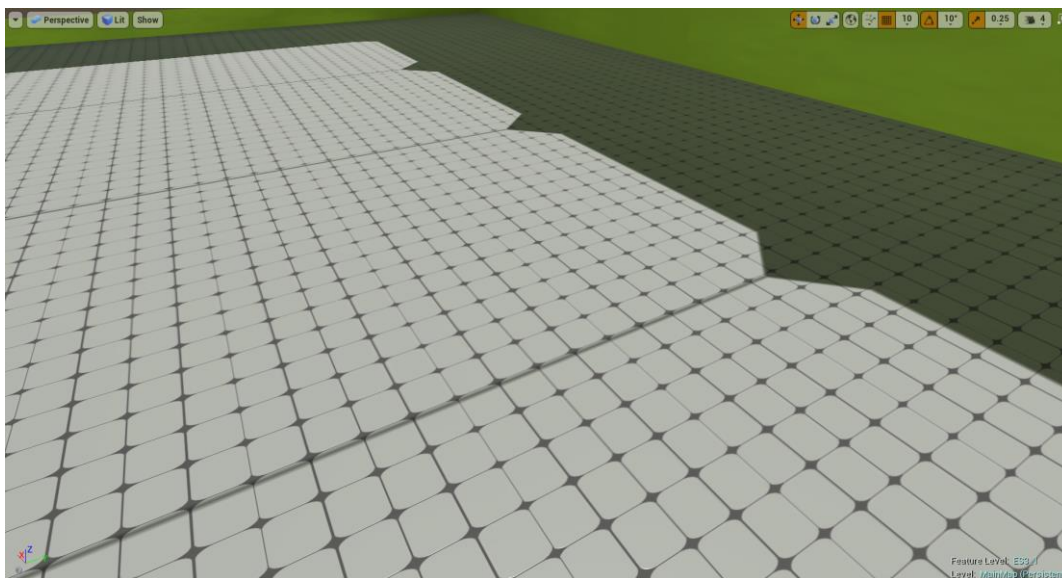


Figura 3.30 Material aplicado al piso.

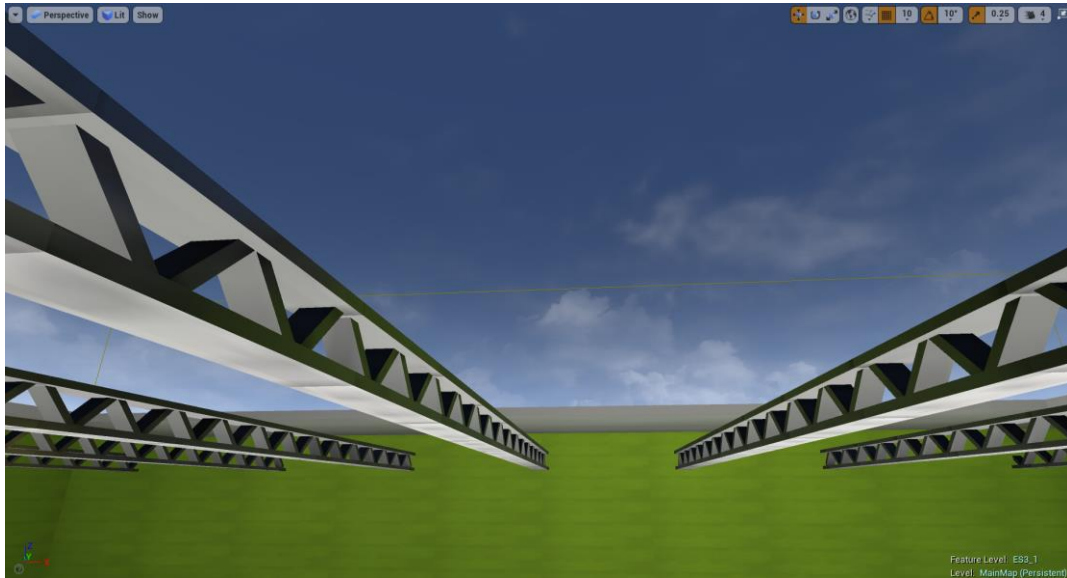


Figura 3.31 Material aplicado a las vigas.



Figura 3.32 Material aplicado al techo.

- Instrucciones: Para evitar que el entorno pareciera vacío, se pusieron tuberías, placas de orificios y una bomba de agua como muestra en unas repisas. De esta manera, el usuario es capaz de ver dichos elementos de manera lateral y frontal; de igual forma se encuentra una breve descripción de cada elemento junto con el ícono que lo representa.



Figura 3.33 Muestra de la placa de orificio segmentada junto con su descripción.

También, en otra de las paredes se colocó una textura de 85.75x25.75 m en la que se encuentran descritas las instrucciones de interacción dentro del entorno virtual.

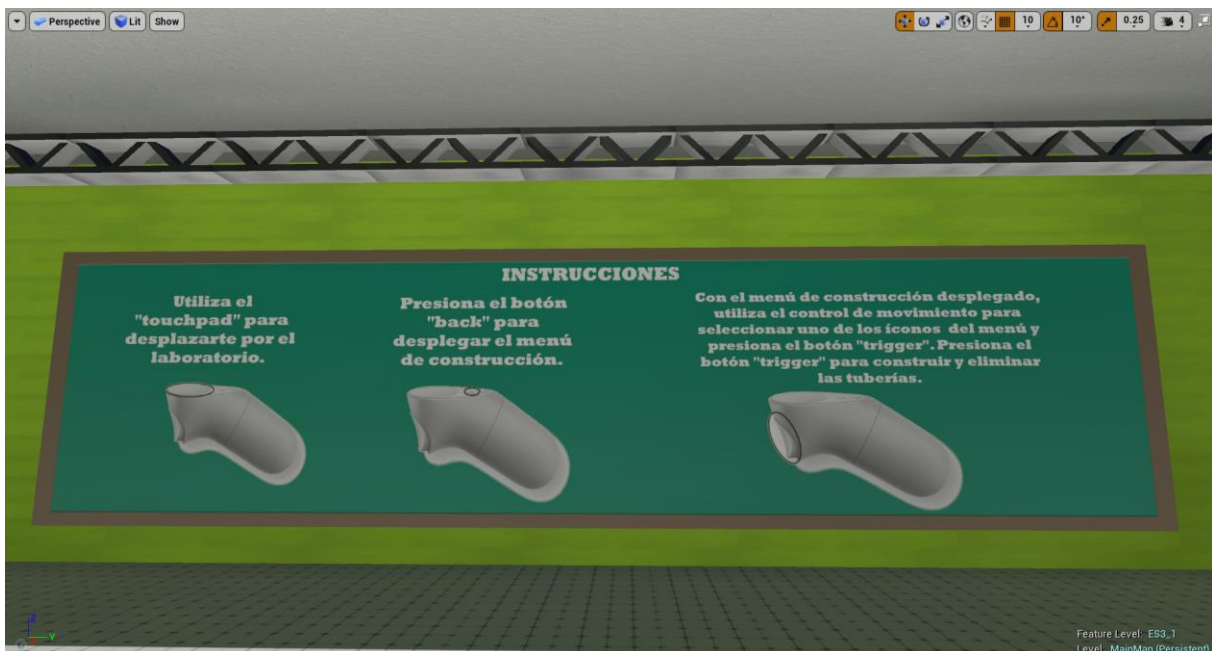


Figura 3.34 Instrucciones para interactuar en el entorno virtual.



Quedando como diseño final lo siguiente:



Figura 3.35 Primer diseño del entorno virtual.

### 3.4 Partícula del agua

Gracias al sistema de partículas con el que cuenta *Unreal Engine 4* se realizó una simulación muy sencilla del comportamiento del agua. De igual manera, fue necesario el diseño de una textura y de un material para las partículas. Dadas las limitantes de procesamiento del *Oculus Go* solo se logró introducir parámetros básicos para el comportamiento de las partículas. Se obtuvo lo siguiente:

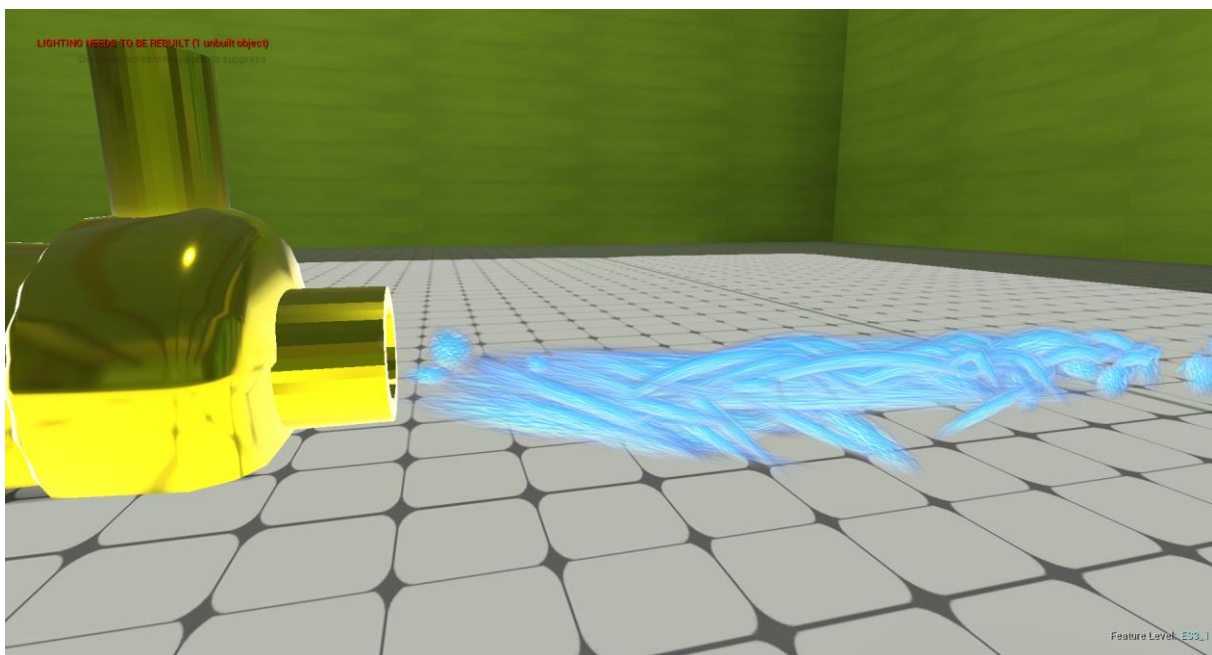


Figura 3.36 Diseño final de la partícula de agua.

## 3.5 Sistema de construcción

El sistema de construcción es la parte fundamental del software, debido a que el usuario tendrá la libertad de crear el sistema hidráulico como lo desee.

### 3.5.1 Interacción del software

Los entornos virtuales de tipo *sandbox* son entornos de juego construidos en los que los jugadores cuentan con una gran libertad de acción y exploración. A pesar de que el juego está determinado por unos principios generales que establecen sus límites y posibilidades, los usuarios tienen una notable capacidad de elección para definir sus objetivos, así como para generar de forma activa la secuencia narrativa que desean crear en el proceso de juego. A diferencia de la estructura habitual de los entornos virtuales, en los que existe una “gran historia” que enmarcan los sucesos narrativos, en los *sandbox* la narrativa es construida por el propio jugador. De igual modo, el ambiente suele ser flexible, los jugadores deciden qué modificar, añadir o eliminar, superando incluso los límites dispuestos en el juego y con ello configuran nuevas metas que no fueron definidas por el creador [52].

Indiscutiblemente, estas características estimulan la creatividad del usuario, lo que resulta benéfico para fines educativos. No obstante, esa esencia abierta de este tipo de juegos depende del tipo de usuario. Todos los juegos requieren de un “jugador” o “jugadora” y para el caso especial del *sandbox* requiere de usuarios que disfruten con la libertad que se les brinda de construir su propio relato, que obtengan un beneficio de una libertad casi completa de crear y destruir [52].

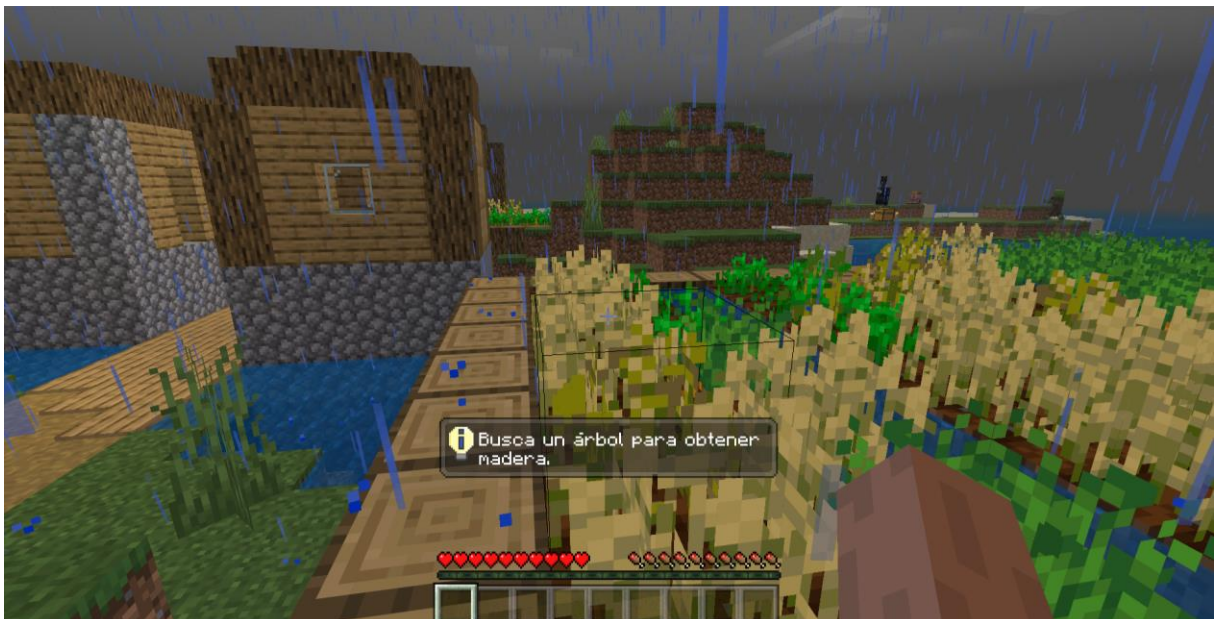


Figura 3.37 *Minecraft* el videojuego tipo *sandbox* más popular.

### 3.5.2 Desarrollo de *Blueprints*

Durante el desarrollo del sistema de construcción, se utilizaron diferentes *Blueprints* divididos en varias categorías: Actores, Enumeraciones, Estructuras y Personaje.



Figura 3.38 Categorías de *Blueprints*.

#### 3.5.2.1 Enumeraciones

Una enumeración es una lista de valores identificados. Para este caso se hace uso de una enumeración de nombre: “e\_EstadosDeJugador”.

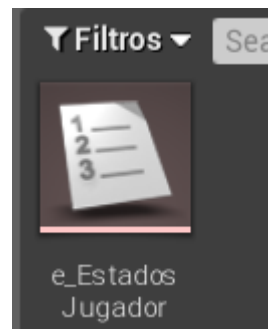


Figura 3.39 Enumeración utilizada.

Una vez iniciado el juego, el usuario tendrá la posibilidad de desplazarse y observar todo lo contenido en el Nivel o construir un sistema hidráulico. Por lo tanto, se definen dos estados diferentes en los cuales puede encontrarse el usuario: “Modo Libre” y “Modo Construcción”; estos son valores que se encuentran dentro de la enumeración “e\_EstadosDeJugador”. En “Modo Libre” no se podrá acceder a la interfaz de construcción, por otra parte, en “Modo Construcción” se accederá a la interfaz de construcción y a todas sus funciones, permitiendo la construcción de sistemas hidráulicos.

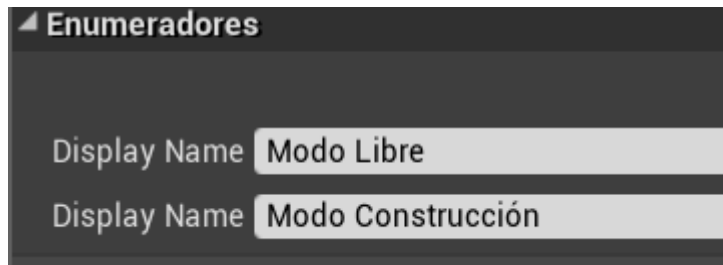


Figura 3.40 Valores de la enumeración “e\_EstadosDeJugador”.

### 3.5.2.2 Estructuras

Una estructura es una colección de diferentes tipos de datos que están relacionados y unidos para facilitar el acceso [53]. Se definieron dos estructuras cuyos nombres son: “es\_DatosDeConstrucción” y “es\_DatosDeRecurso”.

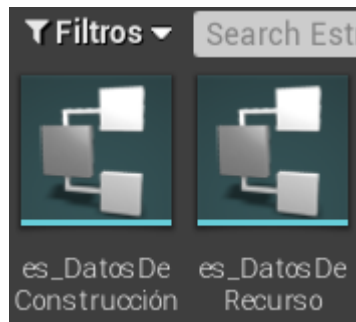


Figura 3.41 Estructuras realizadas.

Los datos que conforman una estructura se conocen como variables, las cuales pueden ser de diferente tipo: vector, entero, flotante, texto, etc.

La estructura “es\_DatosDeConstrucción” consiste en establecer los datos con los que contarán los elementos hidráulicos. Cuenta con dos variables: “Nombre” de tipo texto y “Tecla de acceso rápido” de tipo clave o tecla. Por medio de la variable “Nombre” se identifica el nombre que tendrá cada elemento hidráulico, por otra parte la variable “Tecla de acceso rápido” permite acceder rápidamente al elemento hidráulico que se desea. Cabe señalar, que la variable “Tecla de acceso rápido” tiene la restricción de utilizarse por medio de un controlador, sin embargo, al emplear el visor *Oculus Go*, su controlador está limitado en cuanto a botones se refiere, por lo que esta variable únicamente se utilizó durante el desarrollo del juego por medio del teclado de la computadora con la finalidad de detectar errores y no está implementada en el juego final.



Figura 3.42 Variables de la estructura “es\_DatosDeConstrucción”.

Como su nombre lo indica, la estructura “es\_DatosDeRecurso” proporciona los datos con los que contará el recurso a utilizar. Cuando se habla de recurso, se refiere al material con el que estarán construidas la tubería recta y las placas de orificios. Esta estructura cuenta con tres variables: “Nombre” de tipo texto, “Color de construcción fantasma” de tipo color lineal y “Material de construcción” de tipo material.

La variable “Nombre” establece el nombre del recurso. Por otro lado, por medio de la variable “Color de construcción fantasma” se elige el color que tendrá la previsualización del elemento hidráulico. La previsualización consiste en observar dónde se colocará el elemento antes de su construcción. Por último, la variable “Material de construcción” es el material con el cual contarán la tubería recta y las placas de orificio una vez construidas.

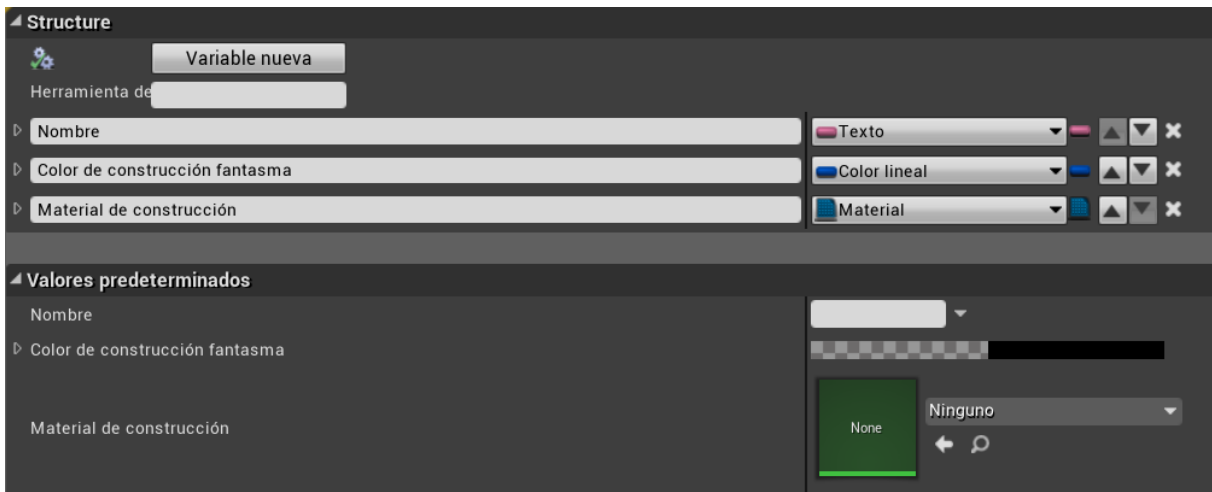


Figura 3.43 Variables de la estructura “es\_DatosDeRecurso”.

### 3.5.2.3 Actores

En esta categoría se localizan las Clases de tipo Actor y Componente de Actor, las cuales permiten modificar el comportamiento y propiedades de los elementos hidráulicos cuando estos se encuentran en previsualización y también una vez construidos. Las Clases se dividen a su vez en tres categorías: Componentes de Actor, Elementos Hidráulicos y Recursos.





Figura 3.44 Categorías de Actores.

### Componentes de Actor

Los Componentes de Actor son piezas funcionales que se añaden a un Actor para que este pueda acceder a todas las funciones con las que cuenta el Componente. Se cuenta con dos Componentes de Actor: “BpC\_DirectorDeConstrucción” y “BpC\_Recursos\_Jugador”. Estos Componentes se añaden al *Blueprint* “PersonajeVR2”, de esta manera el usuario es capaz de acceder a las funciones.



Figura 3.45 Componentes de Actor.

Por medio del Componente “BpC\_Recursos\_Jugador” se obtiene el recurso con el que contará el elemento hidráulico. El Componente dispone de un evento, una función y tres variables.

Un evento es un nodo que ejecuta un código dentro de un *Blueprint*. Permite que el *Blueprint* realice una serie de acciones en respuesta a ciertos acontecimientos que ocurren dentro del juego [54].

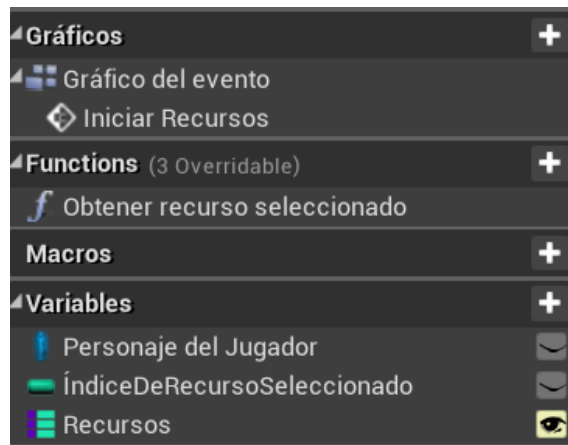


Figura 3.46 Contenido del Componente de Actor “BpC\_Recursos\_Jugador”.

El Componente “Bpc\_DirectorDeConstrucción” es el encargado de realizar la previsualización del elemento hidráulico y manda la “orden” de construir el elemento actual al *Blueprint* “Bp\_Tubería\_Maestra”. Es la Clase que cuenta con mayor cantidad de eventos, funciones y variables; que en conjunto hacen posible el correcto funcionamiento del sistema de construcción.

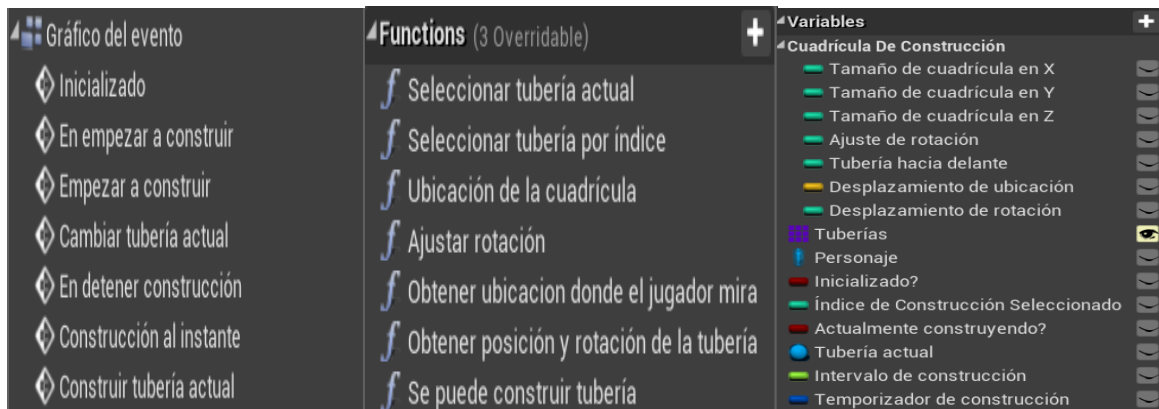


Figura 3.47 Contenido del Componente de Actor “BpC\_DirectorDeConstrucción”.

## Elementos hidráulicos

Esta categoría está conformada por Clases de tipo Actor que contienen cada uno de los elementos hidráulicos previamente modelados. Existe la particularidad de que se tiene una Clase Padre de nombre “Bp\_Tubería\_Maestra” y cuatro Clases Hijo de nombres: “T\_Recta”, “T\_Concéntrica”, “T\_Segmentada” y “Bomba”.

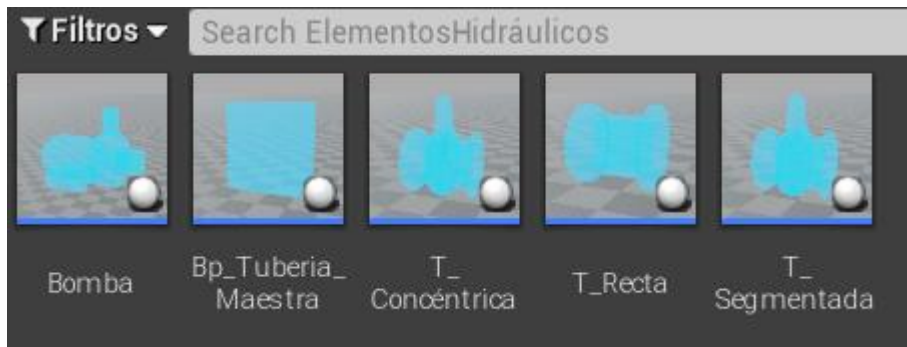


Figura 3.48 Blueprints de los elementos hidráulicos.

A través del Blueprint “Bp\_Tubería\_Maestra” se establecen los Componentes indispensables para agregar los modelos realizados de los elementos hidráulicos y definir límites en el sistema de construcción. Estos límites se establecen por medio de una caja de colisión, en donde una colisión se puede definir como el choque entre dos cuerpos. Asimismo, se agrega la estructura “es\_DatosDeConstrucción” mediante una variable, lo que permite identificar cada elemento por medio de su nombre.

Como se mencionó anteriormente, una Clase Padre le transmite todos sus componentes, variables y propiedades a las Clases Hijo, y es por esta razón que se decidió utilizar este método para facilitar la realización del *Blueprint* de cada elemento hidráulico. En el caso particular del *Blueprint* “Bomba”, se agregó un componente extra, el cual es un sistema de partículas que va a permitir observar un fluido a la salida de la bomba.

## Recursos

Cuando se habla de un recurso, es en términos generales un Blueprint el cual contiene el color de previsualización del elemento hidráulico y el material que tendrá una vez construido; estas propiedades se modifican por medio de una variable de tipo estructura “es\_DatosDeRecurso”. Se cuenta con dos recursos: “Bp\_Recurso\_Maestro” y “Recurso\_Vidrio”.



Figura 3.49 Recursos.

De la misma manera en la que se realizaron los *Blueprints* de los elementos hidráulicos, se hace uso de una Clase Padre y de una Clase Hijo. En este caso “Bp\_Recurso\_Maestro” es la Clase Padre; aquí se añade la variable necesaria para modificar las propiedades del recurso a utilizar. “Recurso\_Vidrio” es la Clase Hijo que



hereda la información de la Clase Padre y, como su nombre lo indica, el material a utilizar es vidrio, mientras que el color de previsualización será azul.

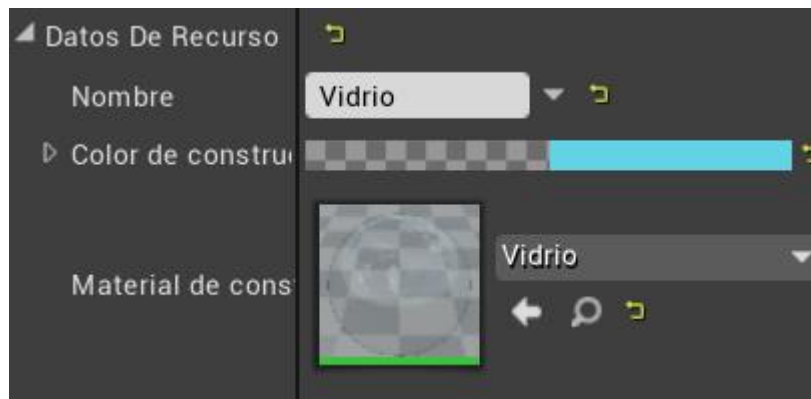


Figura 3.50 Propiedades de del recurso “Bp\_Recurso\_Vidrio”.

#### 3.5.2.4 Personaje

La categoría Personaje está conformada por una Clase tipo Peón de nombre “PersonajePawnVR” y una Clase tipo Personaje de nombre “PersonajeVR2”. En conjunto permiten realizar el movimiento del personaje del usuario dentro del Nivel, además de mostrar el controlador *Oculus Go* y utilizar las funciones que ofrece el mismo.



Figura 3.51 Contenido de la categoría Personaje.

El Blueprint “PersonajePawnVR” es el encargado de realizar el movimiento del controlador en el entorno virtual imitando el movimiento que se realice en el mundo real, además de mostrar un haz de luz proveniente del controlador que permite la interacción con la interfaz del usuario. Por otra parte, el Blueprint “PersonajeVR2” se encarga de efectuar el movimiento bípedo del personaje controlado por el usuario, así como de la configuración de los botones del controlador para realizar diferentes acciones.

### 3.5.3 Desarrollo de la interfaz del usuario

De la misma manera que para el diseño del entorno virtual, se realizaron 8 pruebas con usuarios con el fin de saber cuál sería la mejor disposición de la interfaz gráfica y qué

tan intuitiva resulta. La interfaz contenía íconos de prueba dispuestos de manera circular, el usuario era capaz de interactuar con ellos por medio del controlador una vez que cambiase al “Modo Construcción” y tenía la posibilidad de colocar los diferentes elementos hidráulicos.

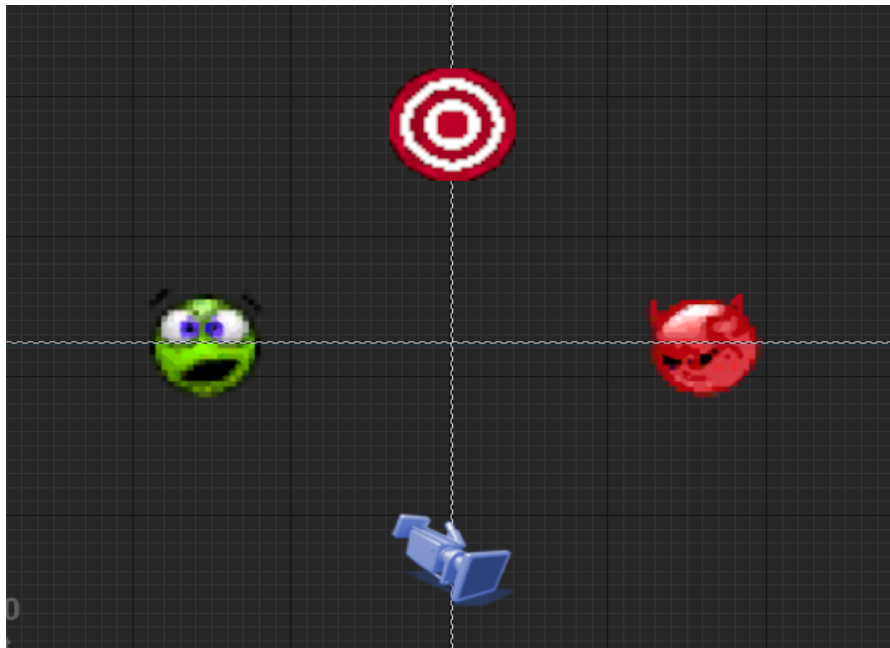


Figura 3.52 Interfaz de prueba con usuarios.

Al explicarles en qué consistía esta prueba, los usuarios realizaron varios comentarios, en donde los que más se repitieron fueron los siguientes:

- “No es tan intuitivo”.
- “Las imágenes están pequeñas”.
- “No me gusta que estén colocadas en forma circular”.
- “Preferiría que hubiese un texto con el nombre de lo que estoy seleccionando”.

A partir de esto, se diseñó la interfaz de construcción con la que contará el usuario y las opciones que incluirá.

Cuando el usuario se encuentra en el estado “Modo Construcción” se muestra una interfaz de construcción intuitiva, con la finalidad de que se seleccionen fácilmente los elementos hidráulicos y tener la posibilidad de eliminar aquellos elementos que se construyeron en un lugar no deseado. Esto se realizó por medio de un *Widget*, en donde se agregaron imágenes previamente realizadas en un editor de imágenes y que hacen alusión a cada elemento hidráulico respectivamente.

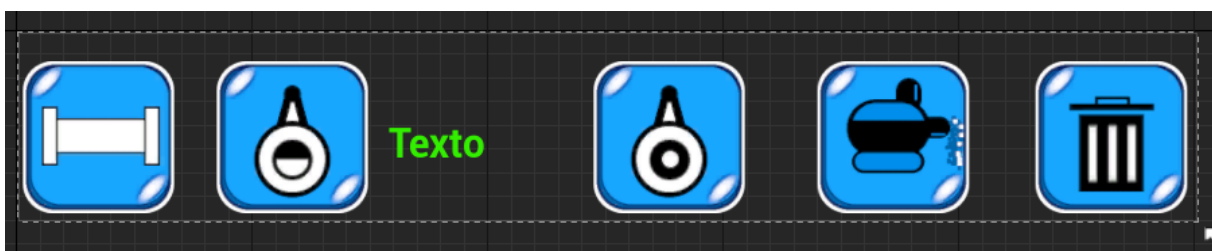


Figura 3.53 Interfaz de construcción.

La interacción con la interfaz se realiza mediante el haz de luz proveniente del controlador. Cuando el haz se encuentra sobre una imagen, esta cambia de color y se muestra el nombre del elemento hidráulico. Para el caso de la imagen que asemeja un bote de basura, se relaciona con la opción de eliminar un elemento hidráulico previamente construido y muestra en la interfaz la palabra “Eliminar”.

### 3.5.4 Funcionamiento

Las pruebas con usuarios que se realizaron para el desarrollo del entorno virtual y la interfaz del usuario, sirvieron para detectar errores que se presentaban en el desplazamiento y rotación del personaje, así como en la construcción de los elementos hidráulicos. Algunos de los errores fueron:

- El controlador virtual se descalibraba.
- La dirección del desplazamiento era incorrecta cuando el personaje rotaba.
- No se visualiza qué elemento hidráulico es el que se va a eliminar.
- No se realiza correctamente la interacción con la interfaz.
- Los botones utilizados para el desplazamiento no responden bien.

Después de que se identificaron estos errores, se procedió a corregirlos y lograr un correcto y fácil funcionamiento del juego.

El manejo del sistema de construcción es sencillo e intuitivo, ya que únicamente se utilizan dos botones del controlador de *Oculus Go* para hacerlo posible. Una vez iniciado el juego, el personaje se encontrará en “Modo Libre”, esto quiere decir que solamente podrá desplazarse en el entorno virtual y observar lo que hay a su alrededor. El desplazamiento se realiza mediante el *touchpad* incluido en el controlador y la dirección del movimiento dependerá de dónde se esté haciendo contacto en el *touchpad*, por ejemplo, si el usuario está realizando contacto en la parte superior del *touchpad*, el movimiento del personaje será hacia delante.

Para cambiar a “Modo Construcción” se debe presionar el botón *back* del controlador, esto provocará que se muestre la interfaz de construcción y se active el haz de luz proveniente del controlador virtual. A continuación, se podrá realizar la interacción con la interfaz mediante el haz de luz, simplemente colocándolo encima de la imagen relacionada con el elemento hidráulico que se desea construir o si, en dado caso, se desea eliminar un elemento, se colocará encima de la imagen que asemeja un bote de basura. A su vez, se mostrará un texto relacionado a la opción a la cual hace referencia la imagen.

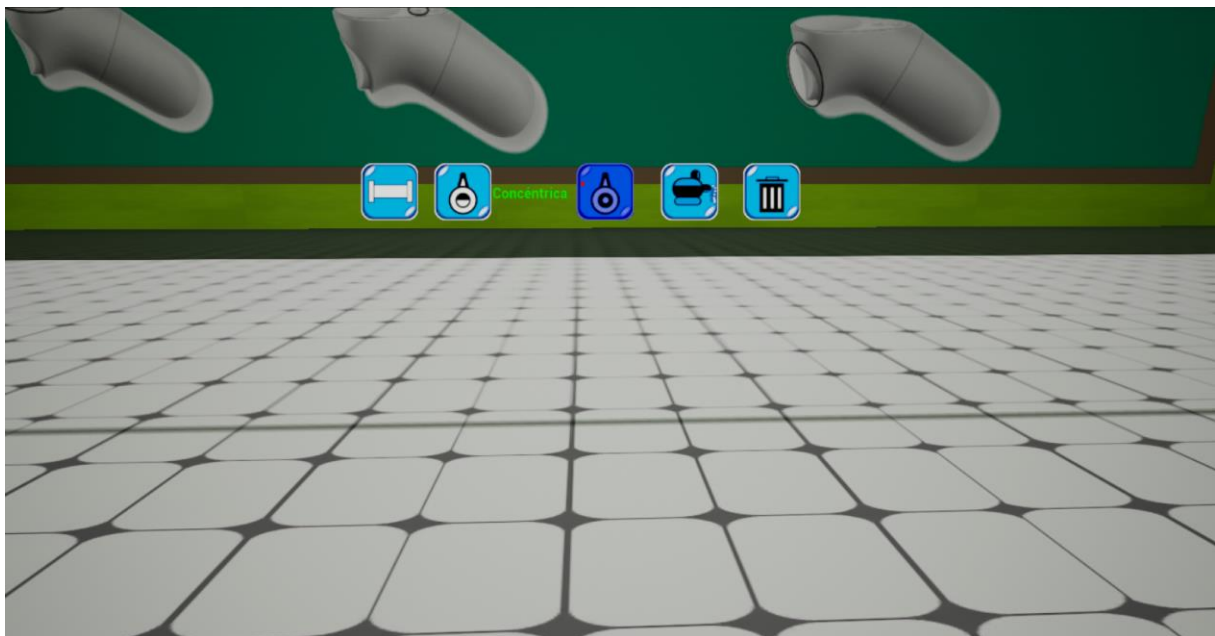


Figura 3.54 Interacción entre el controlador y la interfaz de construcción.

Una vez que el haz de luz se encuentra sobre una imagen, se puede seleccionar esa opción por medio del gatillo del controlador, esto causará una previsualización del elemento hidráulico. En caso de seleccionar la opción de eliminar, esta contará con una previsualización del elemento hidráulico a eliminar; la cual consiste en un parpadeo continuo del elemento que se está observando en ese momento.

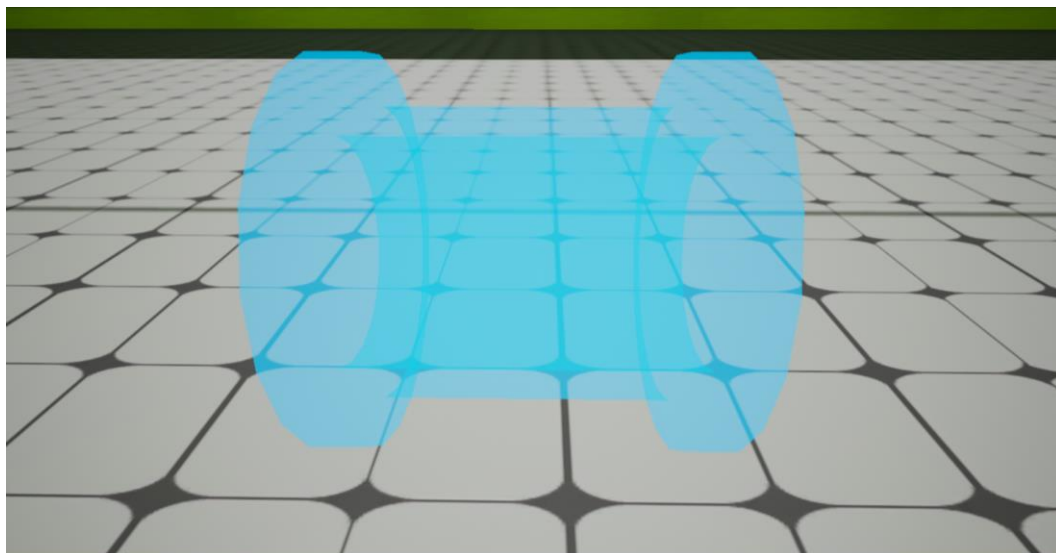


Figura 3.55 Previsualización de una tubería recta.

La función de previsualización consiste en poder observar en dónde se va a construir el elemento hidráulico. El lugar de construcción lo cambiará el usuario mediante el movimiento de su cabeza, ya que el elemento hidráulico seguirá la rotación del visor en relación a su guiñada. Además, la previsualización del elemento mantendrá una distancia respecto al personaje del usuario, por lo que si el usuario decide desplazarse mientras se encuentra la previsualización, el elemento seguirá el movimiento del personaje.

Cuando se tiene identificado el lugar donde se quiere construir el elemento hidráulico, se debe presionar nuevamente el gatillo del controlador para construir el elemento con su respectivo material. Cabe destacar, que si el haz de luz se encuentra sobre una imagen en el momento que se quiera construir, no permitirá la construcción del elemento, debido a que se está realizando la interacción con la interfaz de construcción.

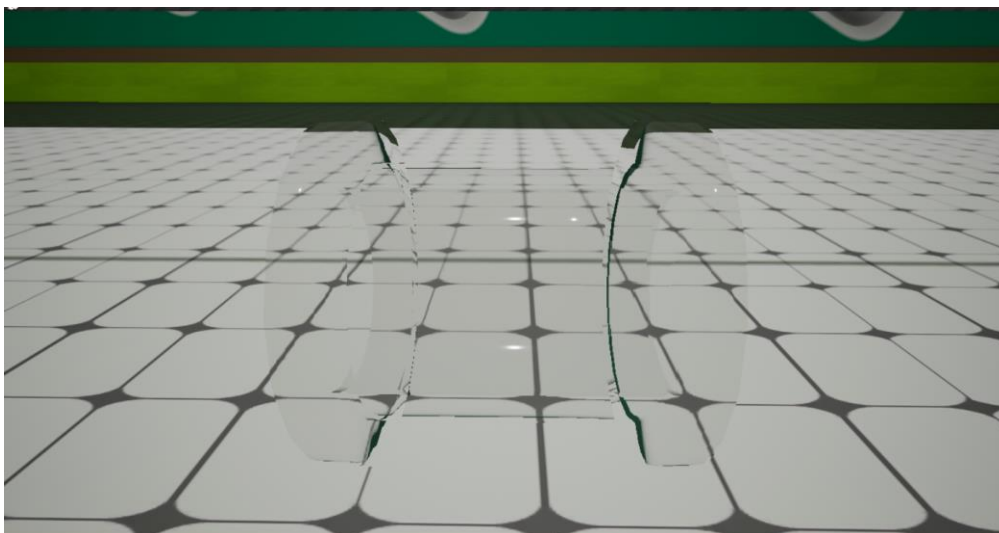


Figura 3.56 Construcción de una tubería recta.

A continuación, se muestra el diagrama de flujo del programa en el que se pueden apreciar las mecánicas de interacción:

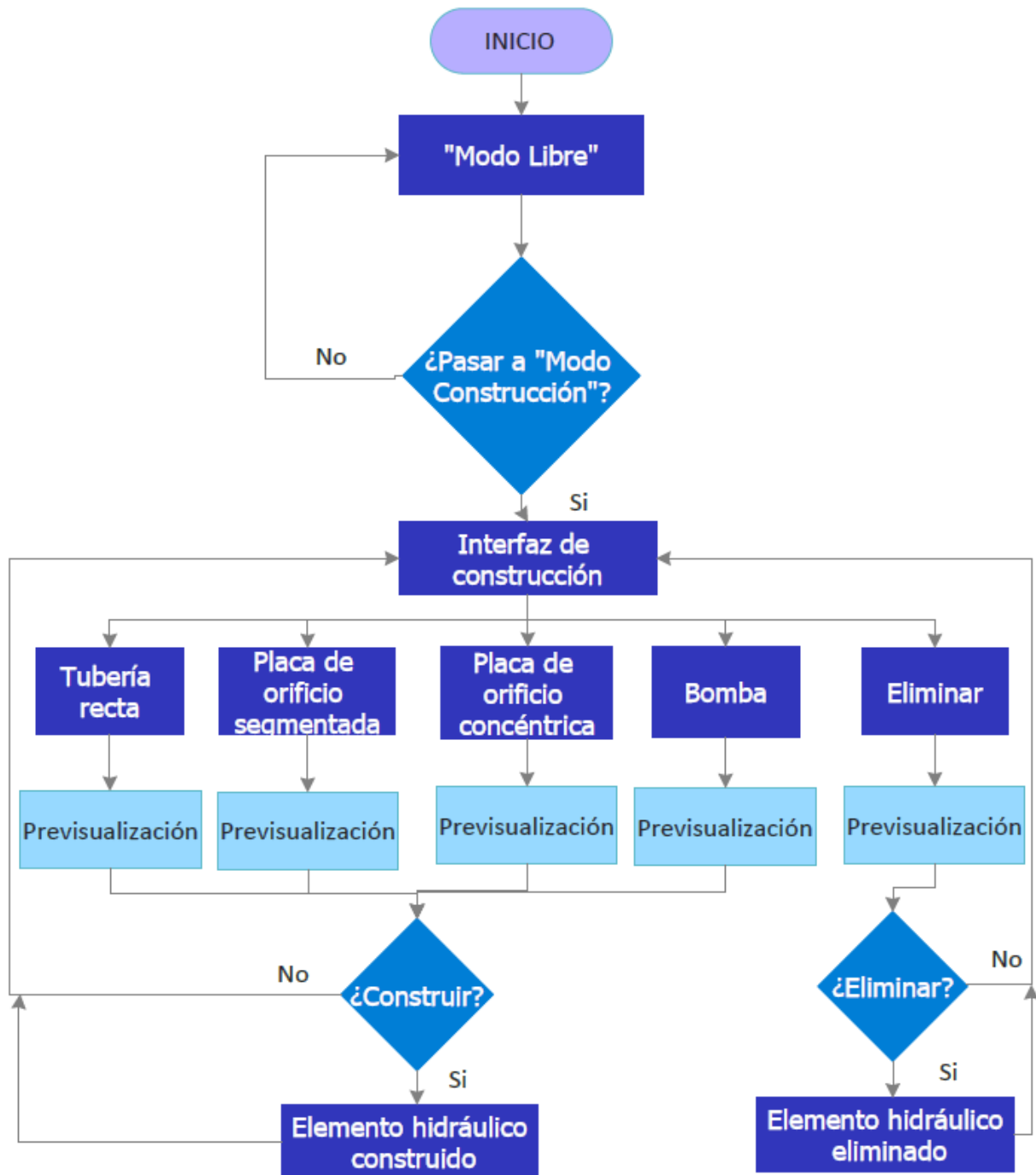


Figura 3.57 Diagrama de flujo que representa el funcionamiento del sistema de construcción.

## 3.6 Pruebas Iteración 1

Una vez realizada la primera versión del software, se elaboraron 10 pruebas de validación (cada una con una duración de 5 minutos) con alumnos de la Facultad de Ingeniería, con la finalidad de comprobar que el software es funcional y cumple con el objetivo del proyecto, es decir, que el usuario es capaz de construir sistemas hidráulicos y además funciona como un nuevo medio para que los alumnos reafirmen los conceptos adquiridos en clase.

Para realizar estas pruebas fue necesario el desarrollo de un protocolo de pruebas el cual se muestra a continuación:

- **Preparación de recursos**

- 1.- Encender el *Oculus Go*.
- 2.- Verificar que la batería del *Oculus Go* y la de su control estén por encima del 70%.
- 3.- Calibrar la referencia de posición del *Oculus Go*.
- 3.- Comenzar la transmisión del *Oculus Go* al celular.
- 4.- Ejecutar el software.
- 5.- Limpiar con un paño seco los lentes y la almohadilla del *Oculus Go*.

- **Recepción del usuario**

- 1.- Explicar en qué consiste el software.
- 2.- Proporcionar el *Oculus Go* al usuario para que se lo coloque.
- 3.- En caso necesario, ajustar las correas del *Oculus Go* a la medida del usuario.
- 4.- Dar el control al usuario.

- **Desarrollo de la prueba**

- 1.- Observar lo que el usuario esté realizando en el entorno virtual a través del celular.
- 2.- Estar en constante comunicación con el usuario con el fin de aclarar dudas.
- 3.- Evaluar el estado físico del usuario.

- **Término de la prueba**

- 1.- Recibir el control por parte del usuario.
- 2.- Ayudar al usuario a quitarse el *Oculus Go*.
- 3.- Limpiar nuevamente con un paño los lentes y la almohadilla del *Oculus Go*.
- 4.- Realización de encuesta.

La encuesta que se les proporcionó a los usuarios es la siguiente:

Instrucciones:

Dependiendo de tu experiencia, indica del 1 al 5 tu nivel de satisfacción (donde 1 representa un nivel muy bajo y 5 un nivel muy alto).

Preguntas	1	2	3	4	5
Manejo del control					
1.- Configuración de botones (¿Qué es lo que hace cada botón?)					
2.- Seguimiento de movimiento (¿El control virtual sigue el movimiento del control real?)					
3.- Apariencia del haz de luz					
4.- Desplazamiento (¿La velocidad de desplazamiento es adecuada?)					
Interfaz de usuario					
5.- Claridad de la interfaz (¿Los iconos son entendibles?)					
6.- Interacción entre la interfaz y el haz de luz (¿Es fácil seleccionar algo en la interfaz?)					
7.- Identificación de función (¿Es fácil observar la función que se está seleccionando?)					
Sistema de construcción					
8.- Previsualización de elementos hidráulicos (¿Se distingue claramente en dónde se va a construir el elemento?)					
9.- Facilidad para colocar elemento (Seguimiento de la previsualización del elemento respecto al visor)					
10.- Apariencia de los elementos hidráulicos (Forma y color de los elementos construidos)					
11.- Previsualización del elemento a eliminar (¿Es claro qué elemento se va a eliminar?)					
Entorno virtual					
12.- Instrucciones claras (¿Es fácil leer las instrucciones que aparecen en el entorno?)					



13.- Similitud respecto a un laboratorio					
14.- Apariencia (¿Los colores y la iluminación te resultan agradables?)					
15.- Tamaño del espacio					
Fluido de trabajo					
16.- Apariencia (¿Es entendible que el fluido es agua?)					
17.- El agua interactúa correctamente con los elementos hidráulicos.					

Tabla 1 Encuesta proporcionada a los usuarios.

### 3.7 Resultados Iteración 1

A continuación, se muestran algunas imágenes de las pruebas de la iteración 1:



Figura 3.58 Transmisión del Oculus Go al celular en la etapa de preparación de recursos.



Figura 3.59 Transmisión en el desarrollo de la prueba.



Figura 3.60 Desarrollo de pruebas de iteración 1.

Las pruebas de validación con usuarios arrojaron datos muy interesantes, los cuales ayudaron tanto para realizar mejoras como para validar algunos de los aspectos que fueron evaluados.

En primer lugar, en la sección en la que se evaluaron los controles, se encontró que una buena parte de los usuarios no están contentos con la velocidad con la que se desplazan dentro del entorno virtual, esto se puede ver en la figura 3.61.

### Desplazamiento (¿La velocidad de desplazamiento es adecuada?)

10 respuestas

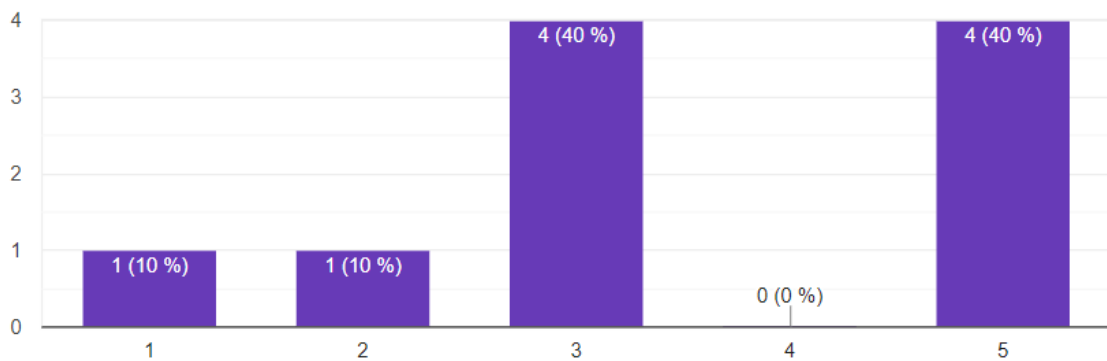


Figura 3.61 Gráfico de resultados de la pregunta 4 de la encuesta.

Por otro lado, uno de los aspectos que se validaron con gran satisfacción fue la facilidad con la que los usuarios pueden interpretar la interfaz gráfica. Como se puede ver en la figura 3.62, el 80% de usuarios validan la inteligibilidad de la interfaz gráfica.

### Claridad de la interfaz (¿Los iconos son entendibles?)

10 respuestas

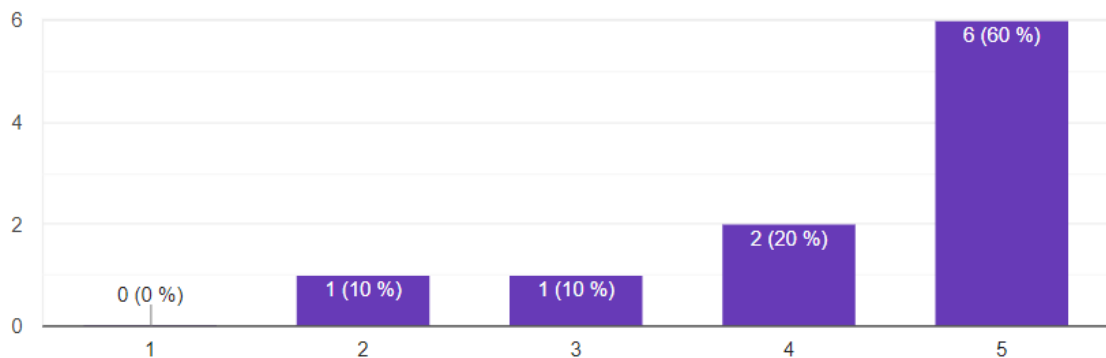


Figura 3.62 Gráfico de resultados de la pregunta 5 de la encuesta.

Además de lo anterior, todos los usuarios aprueban la facilidad con la que pueden interactuar con la propia interfaz (véase figura 3.63), lo que en suma confirma el cumplimiento de uno de los objetivos específicos previamente planteados. Dicho objetivo es el de “Diseñar una interfaz intuitiva al usuario”.

## Interacción entre la interfaz y el haz de luz (¿Es fácil seleccionar algo en la interfaz?)

10 respuestas

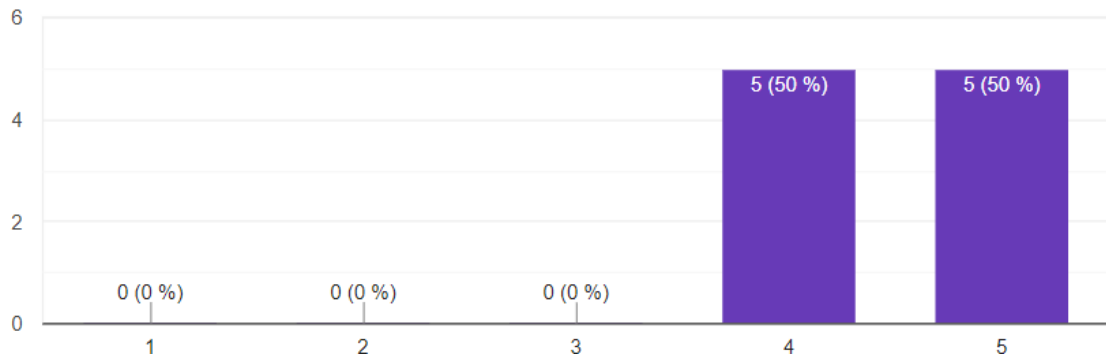


Figura 3.63 Gráfico de resultados de la pregunta 6 de la encuesta.

Simultáneamente, se evaluó el sistema de construcción. Lo que más llamó la atención fue que la mayoría de los usuarios (véase figura 3.64) están contentos con la previsualización que provee el software, pues se considera que es este tipo de retroalimentación la que beneficia a los usuarios en la soltura con la que estos construyen. Dicho lo anterior, se afianzó el cumplimiento de otro de los objetivos específicos, “Diseñar y programar un sistema de construcción en un entorno virtual”.

## Previsualización de elementos hidráulicos (¿Se distingue claramente en dónde se va a construir el elemento?)

10 respuestas

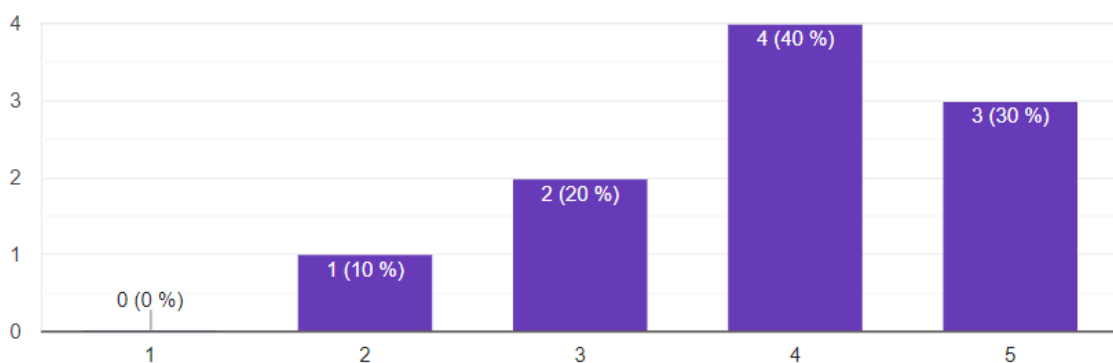


Figura 3.64 Gráfico de resultados de la pregunta 8 de la encuesta.

Conjuntamente, se observó que los usuarios son capaces de comprender las instrucciones que se encuentran dentro del entorno virtual. No obstante, de manera personal los usuarios manifestaron que en cierta distancia, las instrucciones se volvían un poco borrosas y no se leían con tanta nitidez.

## Instrucciones claras (¿Es fácil leer las instrucciones que aparecen en el entorno?)

10 respuestas

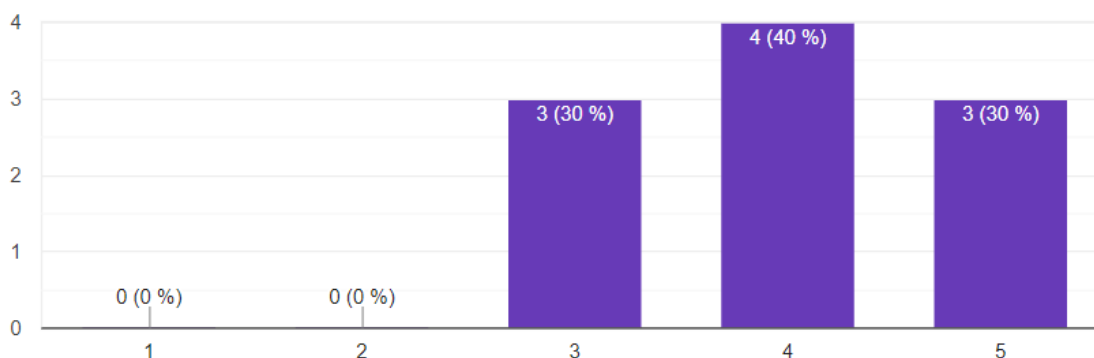


Figura 3.65 Gráfico de resultados de la pregunta 12 de la encuesta.

De la misma manera, los usuarios encuentran similitud entre el entorno virtual y un laboratorio (ver figura 3.66). A pesar de que los usuarios entrevistados pertenecen a la Facultad de Ingeniería, los laboratorios de dicha facultad se asemejan al laboratorio en el que se realiza la práctica de la que se habla en la problemática. Por lo que se considera que es un gran avance en el diseño del sistema, aunque se tendría que esperar por las pruebas de la segunda iteración, las cuales se realizarán con alumnos de la Facultad de Química, para confirmar el cumplimiento de otro de los objetivos específicos (Diseñar un entorno virtual asemejando a un laboratorio).

## Similitud respecto a un laboratorio

10 respuestas

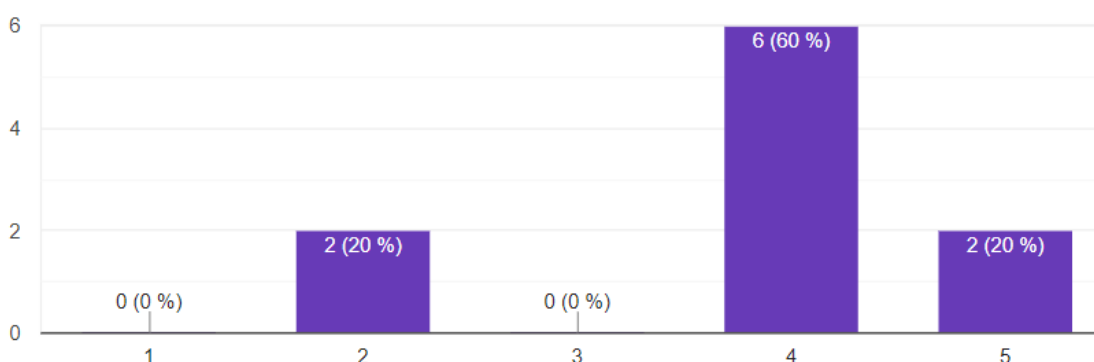


Figura 3.66 Gráfico de resultados de la pregunta 13 de la encuesta.

Por otra parte, se encontró que la partícula de agua diseñada cumple con la función de ser utilizada como fluido de trabajo (se parece y se comporta como agua, ver figuras 3.67 y 3.68). Esto influye directamente en el objetivo principal, aunque no esté de manera explícita. Si el fluido se asemeja al comportamiento del agua, entonces permitirá la correcta experimentación y observación dentro del entorno virtual.

### Apariencia (¿Es entendible que el fluido es agua?)

10 respuestas

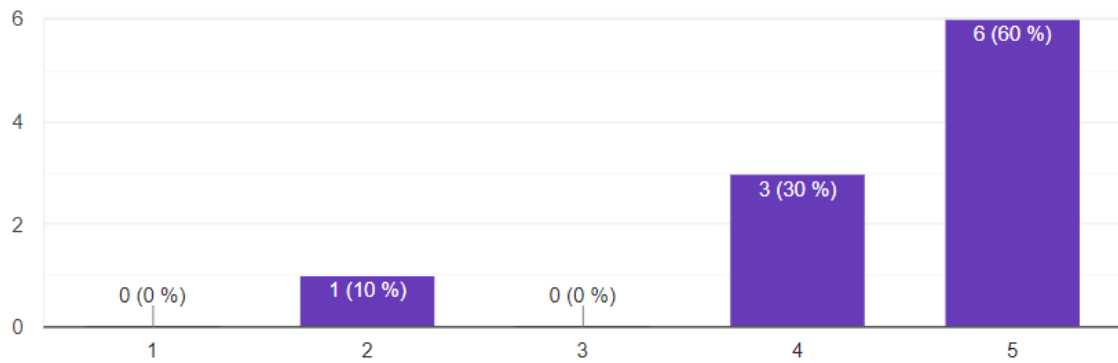


Figura 3.67 Gráfico de resultados de la pregunta 16 de la encuesta.

### El agua interactúa correctamente con los elementos hidráulicos.

10 respuestas

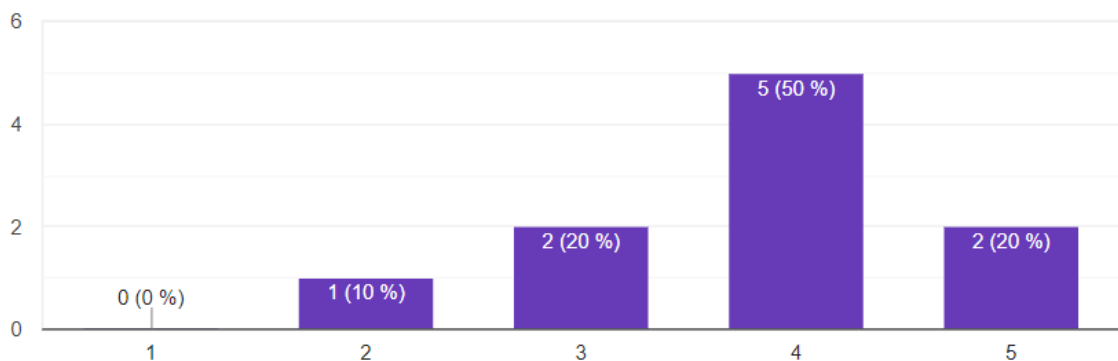


Figura 3.68 Gráfico de resultados de la pregunta 17 de la encuesta.

Por último, en los diálogos que se tuvo con los usuarios durante la realización de las pruebas, expresaron que les incomodaba un poco la iluminación y que el color blanco predomina en el entorno, provocando un poco de mareo.

## 4. DESARROLLO ITERACIÓN 2

En este capítulo se presentan los cambios realizados en el software a raíz de los resultados obtenidos en la iteración 1, así como las nuevas pruebas de validación realizadas a alumnos de la Facultad de Química.

### 4.1 Cambios efectuados

Uno de los primeros cambios que se realizaron en el software fue el hecho de modificar la velocidad de desplazamiento del personaje. En un principio la velocidad se encontraba en un valor de 4 m/s; al estar inconformes los usuarios, se ajustó a 5 m/s. Este valor no puede ser muy alto debido a que tiende a marear a las personas cuando la velocidad es elevada, por lo que el cambio puede parecer pequeño, sin embargo, se aprecia el aumento en la velocidad de desplazamiento del personaje.

Para garantizar una sensación de rapidez en el personaje y que los usuarios no tengan que recorrer grandes distancias para observar las descripciones de los elementos en las repisas, se redujo el tamaño del Nivel; en un principio el tamaño era de 200x200 m, lo cual se disminuyó a 140x140 m, además se cambiaron de lugar las repisas.



Figura 4.1 Segundo diseño del entorno virtual.

Con base en los últimos comentarios de los usuarios, se redujo la iluminación del entorno a 1.7 lux y al material del piso se le redujo la intensidad del color blanco.

Así mismo, se resaltaron los botones en la parte de las instrucciones para que fuese aún más claro identificarlos.



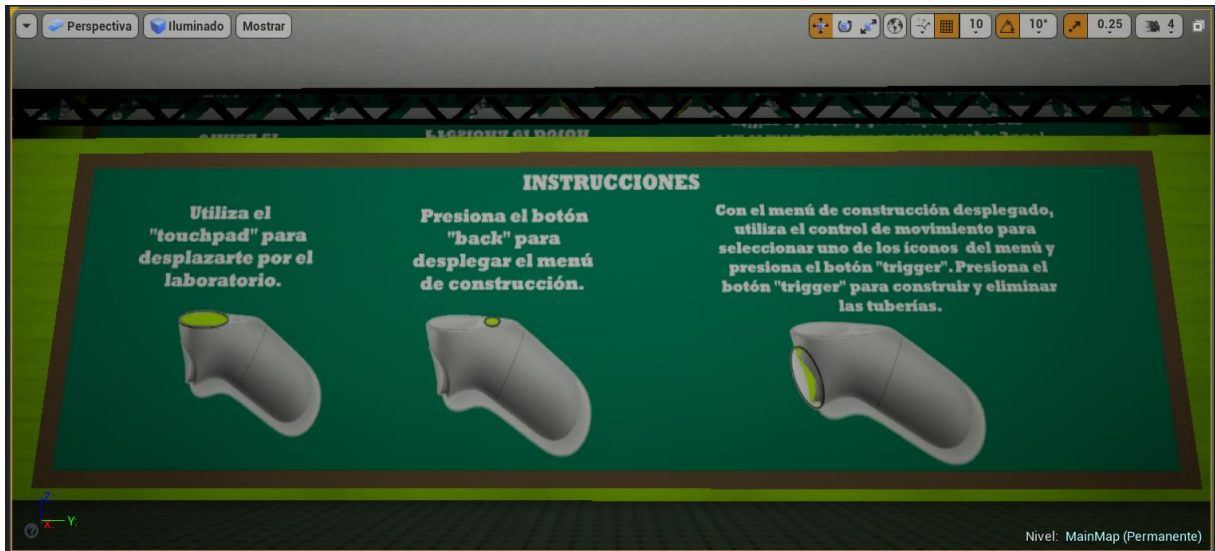


Figura 4.2 Botones resaltados en las instrucciones.

La última modificación que se hizo fue agregar una pantalla de carga al inicio del programa, para añadir información sobre las instituciones que colaboraron en el proyecto.



Figura 4.3 Pantalla de carga.



## 4.2 Pruebas Iteración 2

Una vez realizados los cambios para la segunda versión del software, nuevamente se elaboraron pruebas de validación, pero en este caso se llevaron a cabo con 9 alumnos de la Facultad de Química. Las pruebas tenían la finalidad de comprobar si los cambios realizados previamente, fueron una mejora en el software y por otra parte, constatar que por medio de esta tecnología se pueden reafirmar conceptos vistos en clase.

Se decidió realizar estas pruebas con alumnos de la carrera Química Farmacéutico Biológica, debido a que ellos realizan una práctica de laboratorio en donde hacen uso de un sistema hidráulico; este incluye bomba, tuberías rectas y placas de orificios. De esta manera, su opinión es de suma importancia, ya que permitirá validar si el software cumple con su objetivo.

Durante la realización de las pruebas, se siguió el mismo protocolo de pruebas que en la primera iteración, incluyendo la misma encuesta, permitiendo comparar los resultados entre la iteración 1 y 2. La diferencia radica en que cada una de las pruebas tuvo una duración de 7 minutos, además de que se usaron dos *Oculus Go* para que los alumnos se acercaran en parejas y esto permitiría que las pruebas se realizaran en el menor tiempo posible.

## 4.2 Resultados Iteración 2

A continuación se muestran algunas imágenes de las pruebas de la iteración 2:



Figura 4.4 Desarrollo de pruebas con dos *Oculus Go*.



Figura 4.5 Realización de pruebas con alumnos de la Facultad de Química.

Las pruebas realizadas con alumnos de la Facultad de Química presentaron datos positivos que hacen dar cuenta que los cambios efectuados para la segunda iteración, fueron una mejora en el software.

Uno de los aspectos con el que los usuarios anteriormente no estaban conformes, era la velocidad de desplazamiento del personaje. Al haber hecho un ajuste en la misma, la mayoría de usuarios sintió adecuada la velocidad, expresando que se pueden desplazar rápidamente por el entorno sin presentar un efecto de mareo.

### Desplazamiento (¿La velocidad de desplazamiento es adecuada?)

9 respuestas

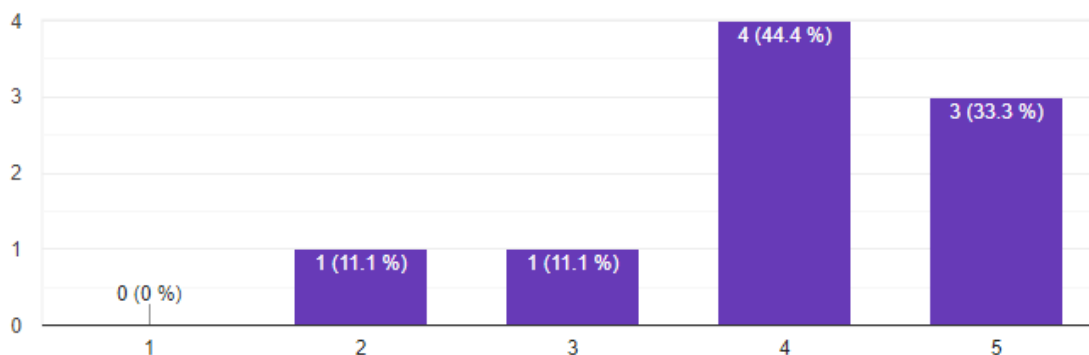


Figura 4.6 Gráfico de resultados de iteración 2 de la pregunta 4 de la encuesta.

Por otra parte, los usuarios manifestaron que las instrucciones eran claras y que al ver resaltados los botones en el control, esto les facilitó identificar a qué botón se hacía referencia.

## Instrucciones claras (¿Es fácil leer las instrucciones que aparecen en el entorno?)

9 respuestas

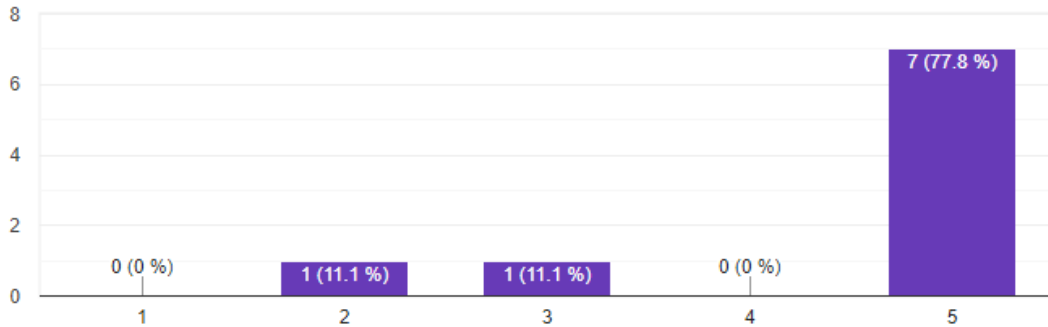


Figura 4.7 Gráfico de resultados de iteración 2 de la pregunta 12 de la encuesta.

Simultáneamente, como se mencionó en los resultados de la primera iteración, se evaluó la similitud entre el entorno virtual y un laboratorio. Los datos que se obtuvieron por parte de los alumnos de la Facultad de Química (véase figura 4.8) indican que todos asociaron el entorno virtual a un laboratorio. Por lo que esto permite confirmar el cumplimiento de otro de los objetivos específicos, “Diseñar un entorno virtual asemejando a un laboratorio”.

## Similitud respecto a un laboratorio

9 respuestas

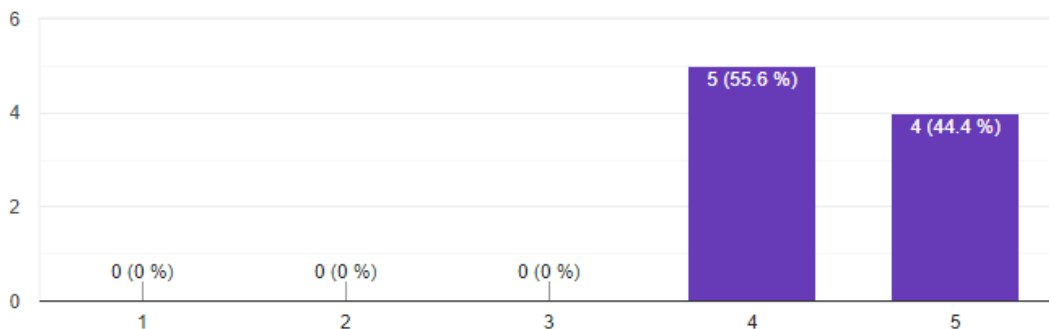


Figura 4.8 Gráfico de resultados de iteración 2 de la pregunta 13 de la encuesta.

Aunado a esto, los resultados permitieron confirmar de la misma manera que en la primera iteración, que los otros objetivos específicos planteados, se cumplieron con gran satisfacción. Los usuarios manifestaron que la interfaz de construcción es muy intuitiva y que se entiende claramente a lo que hacen referencia los íconos. El sistema de construcción es sencillo de utilizar a su parecer, debido a que prácticamente con un sólo botón se selecciona, construye o elimina un elemento hidráulico. Uno de los aspectos que más les llamó la atención y consideran de gran utilidad, es el hecho de que con el movimiento de cabeza se indica en dónde se va a construir el elemento.

## 5. CONCLUSIONES

Los objetivos que se plantearon en un principio se lograron con la ayuda de 2 principales factores. En primer lugar, gracias al aprendizaje adquirido en las clases de diseño y programación. A pesar de no haber cursado materias sobre diseño de ambientes virtuales o sobre programación con *Blueprints*, con el conocimiento adquirido en materias como técnicas de programación y CAD-CAM se logró aprender a utilizar los programas con los que se desarrolló el proyecto (*Maya y Unreal Engine 4*).

En segundo lugar, resultó de gran ayuda el involucrar a los usuarios de manera directa en la fase de diseño. Gracias a la experiencia adquirida con otros proyectos en los que se utilizó la metodología de *design thinking*, se consideró primordial involucrar al usuario, pues se ha constatado que se obtienen mejores resultados para el desarrollo de proyectos.

Con la ayuda de las pruebas que se hicieron durante el diseño e implementación del proyecto, se descubrieron factores importantes que a simple vista no eran notorios en un principio. Por ejemplo, se ajustó la iluminación y los colores de manera que permitieran una experiencia agradable y llamativa para los usuarios. También, se logró diseñar una interfaz de usuario que cumpliera con el objetivo de ser intuitiva. De la misma manera, la apariencia y comportamiento de los modelos y la partícula de agua resultaron exitosos.

Otra de las cosas que permitieron las pruebas de usuarios, fueron saber cuáles eran los factores que provocan que los usuarios se mareen; tal fue el caso del piso cuadrulado y la alta saturación del color blanco, que en un principio no se imaginó que provocaría esto. También, se supo el tiempo que necesitaban para familiarizarse con los controles, puesto que había alumnos que ya habían utilizado sistemas de realidad virtual y otros no.

Afortunadamente, cuando se hicieron las pruebas con los alumnos de ambas facultades, existía un entusiasmo muy grande por parte de ellos para utilizar el software. Esto fue un resultado muy positivo, aunque inesperado, pues se creyó que por el hecho de ser una tecnología que aún no ha sido probada por tantas personas y por ser una aplicación orientada al aprendizaje podría ocasionar que los alumnos no estuvieran tan interesados. Lo que demuestra que se puede aprovechar el interés por este tipo de tecnologías para hacer llegar a los estudiantes nuevas herramientas que beneficien la educación.

Otra de las cosas que resultaron positivas, fue que los estudiantes hacían comentarios con el fin de mejorar ciertas cosas; hicieron recomendaciones sobre aspecto, funciones, elementos y herramientas extras. Es importante señalar, que se deben hacer mejoras continuas pues las necesidades e interacciones varían dependiendo de los usuarios.

## 6. TRABAJO A FUTURO

Gracias a las iteraciones desarrolladas, los comentarios de los alumnos a los que se les realizaron pruebas y a las conclusiones previamente descritas, se pudieron detallar a continuación los puntos en los que se puede desarrollar el proyecto con mayor profundidad:

- Debido a las limitantes técnicas del *Oculus Go* y a la facilidad con la que *Unreal Engine 4* puede migrar entre plataformas, se podría utilizar un visor de realidad virtual con mayor capacidad gráfica y que pueda utilizar 2 controles. De esta manera se mejoraría el aspecto gráfico y la interacción con el programa, por ejemplo, la colocación más precisa de los elementos hidráulicos y la personalización de los mismos.
- Desarrollar un sistema de partículas que asemejara agua de forma más realista; esto incluye mejorar su comportamiento cuando interactúe con los elementos hidráulicos.
- Uno de los principales comentarios que manifestaron los usuarios una vez que probaron el software, fue el hecho de contar con solamente cuatro elementos hidráulicos, por lo que es un aspecto a mejorar en el futuro. Se tiene pensado implementar codos, unión en forma de T, unión en forma de Y y válvula de bola.
- Resultaría de mucha utilidad contar con una herramienta dentro del juego con la que el usuario fuese capaz de medir la longitud del sistema hidráulico que construyeron.
- Otra de las funciones que podrían ser agregadas, sería una herramienta que permitiera personalizar el tamaño de las tuberías que se construyen, tanto el diámetro como el largo de las mismas.

## 7. REFERENCIAS

### SISTEMAS HIDRÁULICOS

- [1] R. Castillo, "Montaje y reparación de sistemas hidráulicos, bienes de equipo y máquinas industriales", ProQuest Ebook Central. [Online]. Available: <https://ebookcentral.proquest.com/lib/bibliodgbsp/reader.action?docID=3212477>. [Accessed: 10-Oct-2019].
- [2] O. M. Gamboa, "Evaluación de métodos para la medición de dureza en materiales cerámicos y refractarios de ingeniería", 2004. [Online]. Available: <http://cdigital.dgb.uanl.mx/te/1020146153.PDF>. [Accessed: 10-Oct-2019].
- [3] "COMO FUNCIONA UN SISTEMA HIDRÁULICO | Partes y aplicaciones." [Online]. Available: <https://comofunciona.co.com/un-sistema-hidraulico/>. [Accessed: 10-Oct-2019].
- [4] "Principio de Pascal | Física Termodinamica." [Online]. Available: <https://hernanleon1002.wordpress.com/fisica-de-fluidos-y-termodinamica/primer-corte/marco-teorico/principio-de-pascal/>. [Accessed: 12-Oct-2019]
- [5] "Sistemas hidráulicos". [Online]. Available: <http://cursos.aiu.edu/Sistemas%20Hidraulicas%20y%20Neumaticos/PDF/Tema%201.pdf>. [Accessed: 12-Oct-2019]
- [6] "BOMBA HIDRÁULICA | Que es, tipos, partes y como funciona." [Online]. Available: <https://como-funciona.co/una-bomba-hidraulica/>. [Accessed: 12-Oct-2019].
- [7] Instrument Engineers Handbook: Process Measurement and Analysis, Fourth Edition, VOLUME 1
- [8] J. O. Vilariño, "Vena Contracta Image," no. 2600, 1997.
- [9] "PLACA DE ORIFICIO | Que es, como funciona y aplicaciones." [Online]. Available: <https://como-funciona.co/una-placa-de-orificio/>. [Accessed: 12-Oct-2019].
- [10] "Medidores de flujo diferenciales. Placas de orificio." [Online]. Available: <http://www.sapiensman.com/tecnoficio/docs/doc16.php>. [Accessed: 12-Oct-2019].

### PROGRAMAS DE MODELADO

- [11] "About — blender.org." [Online]. Available: <https://www.blender.org/about/>. [Accessed: 12-Oct-2019].
- [12] "RENDER - RENDERING - RENDERIZADO | TERMINOLOGIA | TERMINOLOGIA." [Online]. Available: <http://www.3dcadportal.com/rendering.html>. [Accessed: 12-Oct-2019].
- [13] "Nuevas características de 3ds Max 2020 | Modelado y renderización 3D | Autodesk." [Online]. Available: <https://www.autodesk.mx/products/3ds-max/features>. [Accessed: 12-Oct-2019].

[14] R. Nemeč, "Using and Citation of 3D Modeling Software for 3D Printers," vol. 11, no. 2102, pp. 160–170, 2017.

## **REALIDAD VIRTUAL**

[15] S. M. Malfatti, S. R. Dos Santos, L. M. Fraga, C. M. Justel, P. F. F. Rosa, and J. C. De Oliveira, "The Design of a Graphics Engine for the Development of Virtual Reality Applications," *Rev. Informática Teórica e Apl.*, vol. 15, no. 3, pp. 25–46, 2009.

[16] "Realidad Virtual - Brainsigns." [Online]. Available: <https://www.brainsigns.com/es/science/s2/technologies/virtual-reality>. [Accessed: 12-Oct-2019].

[17] E. Peek, B. W??nsche, and C. Lutteroth, "Virtual reality capabilities of graphics engines," 21st Int. Conf. Cent. Eur. Comput. Graph. Vis. Comput. Vision, WSCG 2013 - Commun. Pap. Proc., pp. 39–48, 2013.

[18] "How Display Technologies work in AR/VR - Alishba Imran - Medium." [Online]. Available: <https://medium.com/@alishbai734/how-display-technologies-work-in-ar-vr-6448445fc9ca>. [Accessed: 12-Oct-2019].

[19] "Fundamentals of display technologies for Augmented and Virtual Reality - By Akshay Kore." [Online]. Available: <https://hackernoon.com/fundamentals-of-display-technologies-for-augmented-and-virtual-reality-c88e4b9b0895>. [Accessed: 12-Oct-2019].

[20] "What is Virtual Reality (VR)? Ultimate Guide to Virtual Reality (VR) Technology." [Online]. Available: [https://www.realitytechnologies.com/virtual-reality/#targetText=The most common types of,Emitting Diode \(OLED\) screen](https://www.realitytechnologies.com/virtual-reality/#targetText=The most common types of,Emitting Diode (OLED) screen). [Accessed: 12-Oct-2019].

[21] "GIROSCOPIO - Que es, como funciona y para que sirve." [Online]. Available: <https://como-funciona.co/un-giroscopio/>. [Accessed: 12-Oct-2019].

[22] "Un Magnetómetro | Campo magnético | Corriente eléctrica." [Online]. Available: <https://es.scribd.com/document/315737786/Un-Magnetometro>. [Accessed: 12-Oct-2019].

[23] "A quick guide to Degrees of Freedom in Virtual Reality - Kei Studios." [Online]. Available: <https://kei-studios.com/quick-guide-degrees-of-freedom-virtual-reality-vr/>. [Accessed: 12-Oct-2019].

[24] "Virtual Reality Motion Tracking Technology Has All the Moves - Virtual Reality Society." [Online]. Available: <https://www.vrs.org.uk/virtual-reality-gear/motion-tracking/>. [Accessed: 12-Oct-2019].

[25] "Mando de movimiento PlayStation Move | Más formas de jugar | PlayStation." [Online]. Available: <https://www.playstation.com/es-es/explore/accessories/playstation-move-motion-controller/>. [Accessed: 12-Oct-2019].

[26] "Oculus Go | Accesorios y piezas." [Online]. Available: <https://www.oculus.com/go/accessories/>. [Accessed: 12-Oct-2019].

[27] "▷ Realidad Virtual vs Realidad Aumentada ¿Cuál es mejor? » Adictec." [Online]. Available: <https://adictec.com/realidad-virtual-vs-realidad-aumentada-cual-es-mejor/>. [Accessed: 12-Oct-2019].

[28] J. Maturana, "HTC Vive, análisis: esto sí que es realidad virtual interactiva" Xataka, 2016.

[29] "VIVETM | VIVE Virtual Reality System." [Online]. Available: <https://www.vive.com/us/product/vive-virtual-reality-system/>. [Accessed: 12-Oct-2019].

[30] "Realidad virtual con las Google Cardboard." [Online]. Available: [https://codigo21.educacion.navarra.es/autoaprendizaje/realidad-virtual-con-las-google-cardboard/#targetText=Las Google Cardboard son unas,un teléfono móvil tipo smartphone](https://codigo21.educacion.navarra.es/autoaprendizaje/realidad-virtual-con-las-google-cardboard/#targetText=Las Google Cardboard son unas,un teléfono móvil tipo smartphone.). [Accessed: 12-Oct-2019].

[31] "PlayStation VR | El dispositivo de juego de RV para PS4 | PlayStation." [Online]. Available: <https://www.playstation.com/es-es/explore/playstation-vr/>. [Accessed: 12-Oct-2019].

[32] "Oculus Go | Funciones." [Online]. Available: <https://www.oculus.com/go/features/>. [Accessed: 12-Oct-2019].

[33] "Gafas Oculus Go. Opinión - Características - Juegos - Precio." [Online]. Available: <https://www.gafasoculus.com/gafas-oculus-go/#tab-con-1>. [Accessed: 12-Oct-2019].

[34] "¿Qué es un pixel? - Red Gráfica Latinoamérica." [Online]. Available: <http://redgrafica.com/Que-es-un-pixel.> [Accessed: 12-Oct-2019].

## **PROGRAMAS PARA ENTORNOS VIRTUALES**

[35] T. F. De Grado, "Guía de desarrollo de un videojuego con Unity en realidad virtual," 2018.

[36] "Powerful 2d - 3d software for modeling, animation, rendering, & simulation | App builder - Unity." [Online]. Available: [https://unity3d.com/es/unity?\\_ga=2.214828368.672691867.1569950035-363352294.1569950035](https://unity3d.com/es/unity?_ga=2.214828368.672691867.1569950035-363352294.1569950035). [Accessed: 12-Oct-2019].

[37] "Programming in Unity: For programmers new to Unity - Unity." [Online]. Available: <https://unity3d.com/es/programming-in-unity>. [Accessed: 12-Oct-2019].

[38] "SCRIPT | meaning in the Cambridge English Dictionary." [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/script?q=scripting>. [Accessed: 12-Oct-2019].



[39] "Unreal Engine | Features." [Online]. Available: <https://www.unrealengine.com/en-US/features#targetText=Unreal Engine 4 is a,stand out from the crowd.> [Accessed: 12-Oct-2019].

[40] "Bombas | The Home Depot México." [Online]. Available: [http://www.homedepot.com.mx/plomeria/bombas-y-equipohidroneumatico/bombas?pageGroup=Category&DM\\_PersistentCookieCreated=true&storeId=10351&storeId=10351&pageName=CategoryPage&pageView=grid&catalogId=10101&beginIndex=0&top\\_category=10016&langId=-5&categoryId=11233&parent\\_category\\_rn=10586&ddkey=http%3AHDMPPhysicalStoreDataCmd#facet:&productBeginIndex:0&facetLimit:&orderBy:&pageView:grid&minPrice:&maxPrice:&pageSize:&](http://www.homedepot.com.mx/plomeria/bombas-y-equipohidroneumatico/bombas?pageGroup=Category&DM_PersistentCookieCreated=true&storeId=10351&storeId=10351&pageName=CategoryPage&pageView=grid&catalogId=10101&beginIndex=0&top_category=10016&langId=-5&categoryId=11233&parent_category_rn=10586&ddkey=http%3AHDMPPhysicalStoreDataCmd#facet:&productBeginIndex:0&facetLimit:&orderBy:&pageView:grid&minPrice:&maxPrice:&pageSize:&) [Accessed: 24-Oct-2019].

[41] "Unreal Engine 4 Terminology | Unreal Engine Documentation." [Online]. Available: <https://docs.unrealengine.com/en-US/GettingStarted/Terminology/index.html>. [Accessed: 12-Oct-2019].

[42] "Level Editor | Unreal Engine Documentation." [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/UI/LevelEditor/index.html>. [Accessed: 12-Oct-2019].

[43] "Introduction to Blueprints | Unreal Engine Documentation." [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/Blueprints/GettingStarted/index.html>. [Accessed: 12-Oct-2019].

[44] "Get Started with UE4 | Unreal Engine Documentation." [Online]. Available: <https://docs.unrealengine.com/en-US/GettingStarted/index.html>. [Accessed: 12-Oct-2019].

[45] "Lighting Basics | Unreal Engine Documentation." [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/Basics/index.html>. [Accessed: 12-Oct-2019].

[46] "Traducir asset del inglés al español: Diccionario Cambridge." [Online]. Available: <https://dictionary.cambridge.org/es/diccionario/ingles-espanol/asset>. [Accessed: 12-Oct-2019].

[47] "Essential Material Concepts | Unreal Engine Documentation." [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/Rendering/Materials/IntroductionToMaterials/index.html>. [Accessed: 12-Oct-2019].

[48] "Cascade Particle Systems | Unreal Engine Documentation." [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/Rendering/ParticleSystems/index.html>. [Accessed: 12-Oct-2019].

[49] "Particle System User Guide | Unreal Engine Documentation." [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/Rendering/ParticleSystems/UserGuide/index.html>. [Accessed: 12-Oct-2019].

[50] “Significado de WIDGET en el Diccionario Cambridge inglés.” [Online]. Available: <https://dictionary.cambridge.org/es/diccionario/ingles/widget>. [Accessed: 12-Oct-2019].

[51] “UMG UI Designer User Guide | Unreal Engine Documentation.” [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/UMG/UserGuide/index.html>. [Accessed: 12-Oct-2019].

## **GÉNERO DE UN VIDEOJUEGO**

[52] F. Gértrudix Barrio, Manuel; Gértrudix Barrio, “Aprender Jugando. Mundos Inmersivos Abiertos Como Espacios De Aprendizaje De Los Y Las Jóvenes,” Rev. Estud. Juv., no. 101, pp. 123–137, 2013.

## ***UNREAL ENGINE 4***

[53] “Struct Variables in Blueprints | Unreal Engine Documentation.” [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/Blueprints/UserGuide/Variables/Structs/index.html>. [Accessed: 24-Oct-2019].

[54] “Events | Unreal Engine Documentation.” [Online]. Available: <https://docs.unrealengine.com/en-US/Engine/Blueprints/UserGuide/Events/index.html>. [Accessed: 24-Oct-2019].

# APÉNDICE I : MODELOS DE ELEMENTOS HIDRÁULICOS

Se puede encontrar en este apéndice los modelos desarrollados de los diferentes elementos hidráulicos en Maya 2019 para la realización del software, estos se encuentran en formato fbx. Se localizan en el repositorio con la siguiente ruta:

**<https://github.com/ferchelsea/Modelos-Caudalis/tree/master>**