



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Prueba de calentamiento de
materiales en Realidad
Virtual**

MATERIAL DIDÁCTICO

Que para obtener el título de

Ingeniera Mecatrónica

P R E S E N T A

Brenda Alejandra Aguilar Pérez

ASESOR DE MATERIAL DIDÁCTICO

M.A. Luis Yair Bautista Blanco



Ciudad Universitaria, Cd. Mx., 2024

Agradecimientos

A mi familia por todo el apoyo que me han brindado para permitirme llegar hasta este punto y por motivarme para seguir adelante en los momentos difíciles.

A mi asesor por todo su apoyo y comprensión durante la elaboración de este trabajo.

A mis amigas por su compañía y apoyo a lo largo de toda la carrera.

Índice general

Introducción	5
1 Planteamiento del problema.....	6
1.1 Justificación	6
1.2 Objetivo.....	7
1.2.1 Objetivo general	7
1.2.2 Objetivos específicos.....	7
2 Antecedentes	9
2.1 Realidad Virtual	9
2.1.1 Realidad Virtual como herramienta en la educación.....	10
2.2 Conductividad Térmica.....	12
2.2.1 Conductividad en metales.....	13
2.3 Prueba de calentamiento	13
2.3.1 Descripción del experimento	13
2.3.2 Resultados esperados.....	13
3 Descripción del laboratorio virtual	15
3.1 Herramientas utilizadas para la elaboración del entorno virtual.....	15
3.1.1 Unity	15
3.1.2 VrChat	16
3.1.3 Udon	17
3.2 Elementos básicos	18
3.2.1 Pizarrón.....	19
3.2.2 Experimento.....	20
3.2.3 Panel de control	20
3.3 Interacciones básicas.....	21
3.3.1 Sensor de Posición.....	21

3.3.2	Cambio de material.....	25
3.3.3	Cambio de temperatura ambiental.....	26
3.3.4	Cambio de temperatura de la resistencia.....	28
3.4	Identificación de los puntos a mejorar.....	28
4	Desarrollo.....	29
4.1	Cambio de material.....	29
4.2	Interfaz del pizarrón.....	30
4.3	Interacciones con el ambiente.....	30
4.4	Gráfica Distancia – Temperatura.....	31
4.4.1	Descripción de los elementos que componen la gráfica.....	32
4.4.2	Interacción para añadir un punto a la gráfica.....	34
4.4.3	Interacción para unir los puntos de la gráfica.....	37
4.4.4	Interacción para eliminar la gráfica.....	41
5	Resultados.....	43
5.1	Pruebas realizadas.....	43
5.2	Análisis de resultados.....	44
6	Conclusiones.....	46
7	Trabajo Futuro.....	48
8	Referencias.....	49

Introducción

El siguiente trabajo aborda el tema del uso de la realidad virtual (RV) como una herramienta didáctica que permite mejorar la calidad de la educación, así como las ventajas que ofrece su uso. La realidad virtual utiliza distintas tecnologías inmersivas para generar escenarios virtuales con los cuales se puede interactuar, actualmente el mayor uso de esta tecnología se encuentra en la industria del entretenimiento, pero cada vez es más utilizada en otras áreas como la medicina, la industria y la educación.

En el campo de la educación la RV ha demostrado ofrecer mejores resultados en comparación a métodos tradicionales ya que la inmersión que experimentan los usuarios genera un mayor interés y mejor desempeño. El desarrollo de este trabajo consiste en la optimización de un laboratorio virtual donde se realiza una prueba de calentamiento de materiales, dicha prueba es realizada en la Facultad de Química. Este proyecto fue realizado con la intención de ofrecer a los estudiantes una herramienta que les permita experimentar en un entorno virtual de manera similar a como lo harían en la vida real, esto con el objetivo de ofrecer una alternativa para cuando no es posible acceder a los laboratorios de manera física o que puedan reforzar los conocimientos adquiridos durante la elaboración de la práctica.

El contenido de este trabajo se encuentra dividido en siete capítulos. El primer capítulo consiste en el planteamiento del problema, así como la justificación y los objetivos que llevaron a la realización de este trabajo. En el capítulo dos se abordan los antecedentes necesarios para la comprensión del trabajo realizado, se explica que es la realidad virtual y como es utilizada en el campo de la educación, así como del concepto de conductividad térmica para finalizar con una descripción de la prueba realizada. En el capítulo tres se describen las herramientas utilizadas para la elaboración de este proyecto y las características de la versión inicial del laboratorio virtual. En el capítulo cuatro se desarrolla y describen los cambios y mejoras realizadas a la versión inicial que permiten que el laboratorio sea funcional. En capítulo cinco se realiza un análisis de los resultados obtenidos con los cambios realizados. El capítulo seis contiene las conclusiones de la elaboración de este trabajo. Finalmente, el capítulo siete contiene los aspectos que tienen campo de mejora y que pueden desarrollar en un futuro.

1 Planteamiento del problema

La elaboración de prácticas juega un papel muy importante dentro de la ingeniería, permite a los estudiantes familiarizarse con el uso del equipo necesario para realizar un experimento, así como visualizar el comportamiento real de algún fenómeno lo que permite comprender de mejor manera la teoría que lo respalda y obtener valores reales que reflejen como el entorno afecta dicho fenómeno.

En algunas ocasiones es imposible acceder a un laboratorio físico, una alternativa es el uso de un entorno en realidad virtual que permita realizar las prácticas de manera muy cercana a como se haría en el mundo real. El uso de un entorno virtual ofrece varias ventajas a los estudiantes;

- Les permite familiarizarse con el uso del equipo antes de la elaboración de una práctica lo que puede reducir el riesgo de cometer errores que puedan afectar la integridad de los alumnos o del equipo utilizado.
- Permite a los estudiantes realizar las prácticas de manera atemporal
- Permite elaborar las practicas cuando no se cuenta con el equipo necesario

El material didáctico elaborado consiste en la optimización de un laboratorio en realidad virtual para realizar una prueba de calentamiento en metales.

1.1 Justificación

El uso de la realidad virtual permite que los usuarios interactúen en un entorno virtual de una manera muy similar a como la harían en el mundo real, la inmersión que se presenta aumenta el interés por parte del usuario lo que genera mejores resultados en el aprendizaje.

El uso de un laboratorio virtual ofrece un gran número de ventajas a los estudiantes: les permite capacitarse antes de utilizar el equipo real lo que reduce las probabilidades de sufrir un accidente o dañar el equipo, le permite realizar las practicas fuera del horario de clase en caso de no poder asistir de manera presencial o para reforzar lo aprendido durante la práctica, además de otorgar la posibilidad de realizar las practicas cuando no se cuenta con el materia o equipo necesario para realizarla de manera presencial o cuando no se cuenta con acceso al laboratorio.

Debido a las ventajas que ofrece el uso de un laboratorio de realidad virtual se busca optimizar un proyecto de titulación elaborado por estudiantes de la Facultad de Ingeniería, este proyecto consiste en un laboratorio que permite realizar una prueba de calentamiento,

lo cual permite observar el comportamiento de la conductividad térmica en distintos materiales.

El desarrollo de este trabajo surge a partir de distintas iteraciones realizadas por otros compañeros de la facultad, los cambios realizados durante la elaboración de este trabajo corresponden a la tercera versión del laboratorio virtual.

La versión recibida del laboratorio a optimizar no permite realizar de manera correcta la prueba de calentamiento, por lo cual se busca realizar los cambios necesarios para que los resultados de este experimento reflejen el comportamiento real y de esta manera el laboratorio pueda ser utilizado por la comunidad estudiantil para la elaboración de prácticas. Además, se busca agregar interacciones que mejoren la sensación de inmersión para los usuarios, así como agregar una gráfica que facilite la lectura de los resultados obtenidos.

1.2 Objetivo

1.2.1 Objetivo general

Optimizar la versión inicial de un laboratorio en realidad virtual para realizar una prueba de calentamiento de materiales

1.2.2 Objetivos específicos

- Modificar el código de manera que los resultados obtenidos para cada material correspondan al valor teórico esperado

En la versión inicial del laboratorio cuando se realiza un cambio de material en la barra los resultados obtenidos no reflejan este cambio, esto se debe a que el código que permite cambiar dicho material no modifica el valor de la constante de conductividad térmica correspondiente, al modificar esta parte del código es posible obtener valores cercanos a los esperados teóricamente lo cual permite que este laboratorio pueda ser utilizado para realizar la práctica correspondiente.

- Agregar el nombre del material utilizado al pizarrón donde se muestran los resultados

Esto permite identificar de manera más sencilla los resultados obtenidos al momento de realizar el experimento, en la versión inicial no es posible visualizar el material de la barra con el que se está trabajando, al agregar este elemento el usuario puede identificar con mayor facilidad los resultados.

- Agregar interacciones con el ambiente virtual

Estas interacciones permiten aumentar la sensación de inmersión dentro del laboratorio virtual, lo que genera que una mejor experiencia para los usuarios al permitirles interactuar con ciertos objetos de la misma manera que lo harían en el mundo físico.

- Agregar una gráfica que permita visualizar el comportamiento distancia-temperatura de cada material

La gráfica les permite a los usuarios observar e identificar de manera más sencilla la relación que existe entre la temperatura y la distancia, además de permitir que se pueda comparar la diferencia que existe en el comportamiento de los distintos materiales.

2 Antecedentes

2.1 Realidad Virtual

La Realidad Virtual (RV) es el conjunto de tecnologías inmersivas que permiten simular un determinado ambiente en un entorno virtual de manera que mediante un ordenador el usuario pueda interactuar con dicho ambiente. [1] La RV permite la generación de un entorno virtual en 3D a través del ordenador el cual el usuario puede visualizar haciendo uso de lentes o visores de realidad virtual, en los visores se proyectan imágenes estereoscópicas que le permiten al usuario observar el entorno virtual como si se encontrara dentro de este. Con el uso de controles de movimiento el usuario puede interactuar en este entorno o con otros usuarios. [2]



Imagen 1. Equipo de Realidad Virtual Oculus Quest 2 (Visores y controles) [3]

Durante el siglo XX se crearon distintas herramientas tecnológicas que sirvieron como base para el desarrollo de la realidad virtual. En 1929, Edward Link creó un simulador de vuelo mecánico para el uso del ejército llamado “*Link Trainer*”. A mediados de los años 50 Morton Heilig presenta una tecnología inmersiva llamada *Sensorama*, esta se encontraba basada en técnicas cinematográfica y ofrecía experiencias de teatro multisensorial. En 1963 aparecen “*The Teleyoggles*” creadas por Hugo Gernsback, se trataba de una pequeña pantalla portátil que se ajustaba en la cabeza como si fueran unas gafas, es considerada el antecedente de los visores de realidad virtual. En 1965 Ivan Sutherland publicó el artículo “*The Ultimate Display*” en el cuál hablaba por primera vez del concepto de realidad virtual sin llegar a utilizar este término. En 1968 Sutherland junto a David Evans crearon “*The Sword of Damocles*”, se trata del primer precursor de los sistemas de realidad virtual modernos. Este sistema permitía crear mundos virtuales con imágenes en 3D, las imágenes eran observadas por el usuario a través de dos pantallas y el brazo que sostenía el casco seguía los movimientos de la cabeza. [4]



Imagen 2. The Sword of Damocles [4]

Los avances tecnológicos han acelerado el desarrollo de la realidad virtual. Durante los años 1980 y 1990 compañías de videojuegos como Sega y Nintendo desarrollaron sistemas de videojuegos que incluían realidad virtual, fueron recibidos con poco éxito y tenían limitaciones técnicas debido a la tecnología de la época. La realidad virtual siguió evolucionando durante el siglo XXI, fue hasta la década del 2010 que compañías como *Oculus* y *HTC Vive* presentaron sistemas de realidad virtual de alta tecnología, contando con una gran potencia gráfica, rastreo de movimiento y el uso de controles externos para interactuar con el entorno virtual. Actualmente los sistemas comerciales de realidad virtual más populares son los *Oculus* producidos por Meta y los *Apple's Vision Pro*. La industria de los videojuegos ha sido la principal impulsora de la RV, pero sus usos no se limitan a esta. Se ha logrado adaptar esta tecnología a distintos ámbitos como la medicina, la industria, la ingeniería, la arquitectura, el entrenamiento militar y la educación. [5]

2.1.1 Realidad Virtual como herramienta en la educación

La tecnología es una herramienta fundamental en la educación, el uso de estas herramientas permite facilitar el acceso a la información y mejorar el proceso de aprendizaje. [6] Por ejemplo, la inesperada pandemia por Covid-19 demostró la necesidad de utilizar diversas tecnologías como apoyo en la educación, al tener que migrar a clases a distancia se

vio la necesidad de contar con un ambiente virtual propicio para que los estudiantes continuaran con sus clases, principalmente mediante videoconferencias, además se mostró la necesidad de contar con equipos de cómputo para realizar las actividades escolares y se hizo uso de simuladores en sustitución a las prácticas presenciales. [7]

Además de la situación extraordinaria ocasionada por la pandemia, existen diversas razones por las cuales sería imposible acceder a un aula física o laboratorio, por ejemplo, falta de recursos para contar con el equipo necesario en un laboratorio, costos altos para dar mantenimiento a las aulas, comunidades que no cuentan con un fácil acceso a las aulas, entre otras.



Imagen 3. Uso de realidad virtual como material didáctico [8]

La Realidad Virtual es un recurso que puede ser utilizado para facilitar la migración de la educación a un ambiente a distancia, ofrece una capacidad de interacción más cercana a la que se tendría con métodos tradicionales, debido a que con esta tecnología los usuarios pueden interactuar con el entorno virtual de manera muy similar a como lo harían en el mundo real, así como interactuar con otros usuarios. Además, permite aprender de maneras que anteriormente no eran posibles, por ejemplo, se puede contar con modelos 3D que permitan explicar de mejor manera un fenómeno ofreciéndole a los alumnos la capacidad de

interactuar con ellos, otra ventaja en el caso de los laboratorios es que se les permite a los estudiantes utilizar el equipo sin miedo a dañarlo o sufrir un accidente. [9]

La empresa *zSpace* utiliza una plataforma de aprendizaje en realidad virtual, la cual le permite a los estudiantes aprender de manera interactiva distintos conceptos de ciencia y tecnología. Al poder realizar experimentos virtuales y manipular distintos objetos en el entorno virtual los estudiantes comprenden de mejor manera los conceptos estudiados. La tecnología ha permitido mejorar la experiencia de aprendizaje para obtener mejores resultados. [8]

La realidad virtual es una herramienta que está transformando los modelos educativos, generando mejoras en el proceso enseñanza-aprendizaje. [7]

2.2 Conductividad Térmica

La transferencia de calor ocurre cuando dos elementos se encuentran a distintas temperaturas, está siempre ocurre del elemento de mayor y temperatura al de menor y el proceso termina cuando se logra el equilibrio térmico, es decir, cuando ambos elementos se encuentran a la misma temperatura. Existen tres formas distintas en las que ocurre la transferencia de calor; conducción, convección y radiación. En la prueba a replicar en el laboratorio virtual la transferencia de calor ocurre mediante la conducción. [10]

La conducción ocurre cuando dos elementos a distinta temperatura se encuentran en contacto físicamente, este fenómeno puede ocurrir en sólido, líquidos y gases. La *conductividad térmica* (k) de un material es una medida de la capacidad de un material para conducir calor. [11] Si el valor de la conductividad es alto dignifica que el material es un buen conductor, si el valor es bajo el material es un mal conductor o es un aislante.

El laboratorio permite realizar una prueba de calentamiento de materiales, el experimento consiste en una barra de metal, esta barra se encuentra 0 [°C] en un extremo y cuanta con una resistencia térmica cuya temperatura es ajustable en el otro extremo. La prueba consiste en medir la temperatura en distintos puntos de la barra de manera que se pueda analizar el comportamiento de la conductividad térmica en distintos materiales. La temperatura en un punto específico de la barra se calcula utilizando la siguiente fórmula.

$$\theta = \frac{\theta_r \operatorname{senh}[L - x] + \theta_h \operatorname{senh}[mx]}{\operatorname{senh}[mL]}$$

$\theta = \text{Temperatura de barra} - \text{Temperatura ambiente}$

$\theta_r = \text{Temperatura de la resistencia} - \text{Temperatura ambiente}$

$\theta_r = \text{Temperatura del hielo} - \text{Temperatura ambiente}$

$x = \text{Posición de la barra}$

$$m = \sqrt{\frac{hP}{kA}}$$

$A = \text{Área transversal de la barra}$

$P = \text{Perímetro de la barra}$

$k = \text{Conductividad térmica del material}$

$h = \text{Coeficiente de transferencia de energía}$

2.2.1 Conductividad en metales

Los metales cuentan con un elevado valor de conductividad térmica lo cual indica que son excelentes conductores de calor, esto se debe a que sus átomos cuentan con electrones libre en las capas externas, lo cual permite que se muevan con mayor facilidad y transporten la energía térmica. [12]

2.3 Prueba de calentamiento

2.3.1 Descripción del experimento

El experimento en el cual se basa el laboratorio virtual realizado consiste en calentar un extremo de una barra de aluminio con vapor de una caldera para encontrar la posición de la barra donde las diferencias entre los calores de los mecanismos conductivos y convectivos son despreciables. Para lograr esto se realizan mediciones a la temperatura de la barra de metal en puntos determinados cada 10 [min]. Con los datos obtenidos se realizan graficas posición-temperatura, utilizando el tiempo como parámetro. A partir de estas gráficas se obtiene el modelo matemático que representa la relación que existe entre estos elementos.

2.3.2 Resultados esperados

Con la realización de este experimento se busca obtener el comportamiento de distintos materiales al aumentar su temperatura en un extremo, así como identificar como afecta el coeficiente de conductividad térmica de cada material y observar la relación distancia – temperatura.

3 Descripción del laboratorio virtual

En esta sección se busca describir los elementos y características principales de la versión inicial del laboratorio virtual realizada por estudiantes de la Facultad de Ingeniería.

3.1 Herramientas utilizadas para la elaboración del entorno virtual

3.1.1 Unity

Unity es un motor gráfico en 2D y 3D que se utiliza principalmente para el desarrollo de videojuegos, Unity permite crear juegos multiplataforma sin necesidad de cambiar de programa. Unity utiliza el lenguaje de programación C# para el funcionamiento y la lógica del juego a realizar. [13] Unity ofrece soporte para Realidad Virtual. [14]

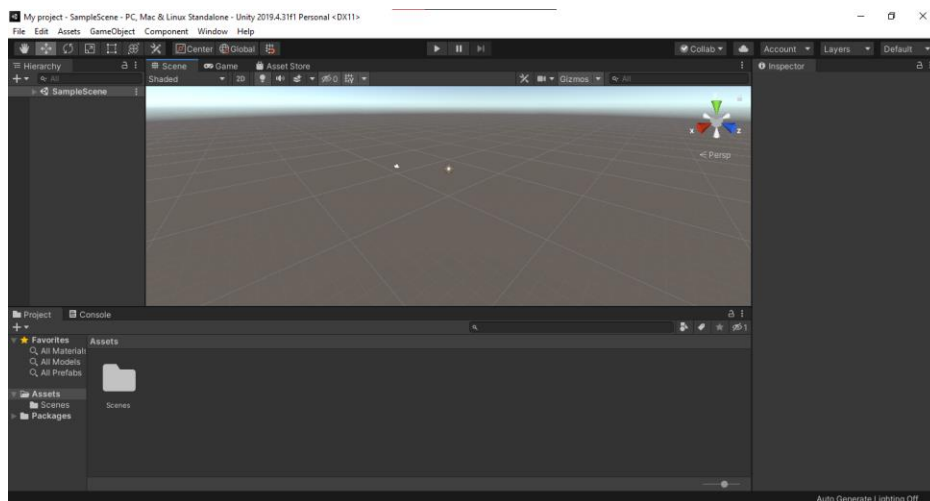


Imagen 4. Interfaz de Unity

Los elementos básicos que se deben comprender para utilizar Unity son: [15]

- **GameObject:** Son los elementos más importantes de Unity, todos los objetos que se encuentran en el juego son GameObject, por si solos no tienen ninguna función, es necesario agregar distintos componentes para otorgarle las características deseadas y se comporten de la manera adecuada.
- **Escenas:** Una escena es el entorno donde se desarrolla el juego, es donde se encuentran todos los elementos de Unity. Este entorno será visible para los usuarios y contendrá los elementos con los que se puede interactuar.
- **Scripts:** Los scripts son el código que permite que los eventos del juego ocurran de la manera deseada, los scripts permiten controlar el comportamiento de un objeto.

En Unity se utiliza el lenguaje C# para elaborar los scripts lo cual es una ventaja ya que se trata de un lenguaje intuitivo y fácil de aprender.

- Modelos: Son archivos que contienen la información sobre la apariencia de objetos 3D, como personajes u objetos ambientales.

3.1.2 VrChat

VrChat es un juego en línea que le permite a los usuarios interactuar entre ellos en un entorno digital en realidad virtual. En VrChat los usuarios pueden crear sus propios avatares y explorar distintos mundos en los cuales pueden realizar distintas actividades. La ventaja de VrChat es que no es necesario contar con un equipo de realidad virtual para poder utilizarla, es posible acceder desde cualquier PC. [16]



Imagen 5. Mundo y Avatares de VrChat [17]

VrChat fue desarrollado en Unity y cuenta con un SDK¹ que permite a los usuarios crear avatares y mundos desde Unity para posteriormente exportarlos a VRChat. Para la elaboración de este proyecto se utilizó la herramienta de creación de mundos de VRChat. [18]

Con el SDK de VrChat se puede crear un mundo de realidad virtual a partir de una escena en Unity, para lograrlo solo es necesario agregar el componente “VRC Scene Descriptor” a la escena, es importante agregar un punto de Spawn a la escena, que es donde nuestro personaje aparecerá al entrar al mundo. [19]

¹ Kit de desarrollo de software

3.1.3 Udon

VrChat utiliza su propio lenguaje de programación llamado “*VRChat Udon*” el cual está diseñado para utilizar una interfaz gráfica de programación llamada “*VRChat Udon Node Graph*” la cual utiliza nodos y “alambres” (o noodles) para conectar salidas y entradas, así como las interacciones deseadas. El propósito de esta interfaz gráfica es facilitar la programación de los comportamientos deseados para personas que no están familiarizadas con los lenguajes de programación. [20]

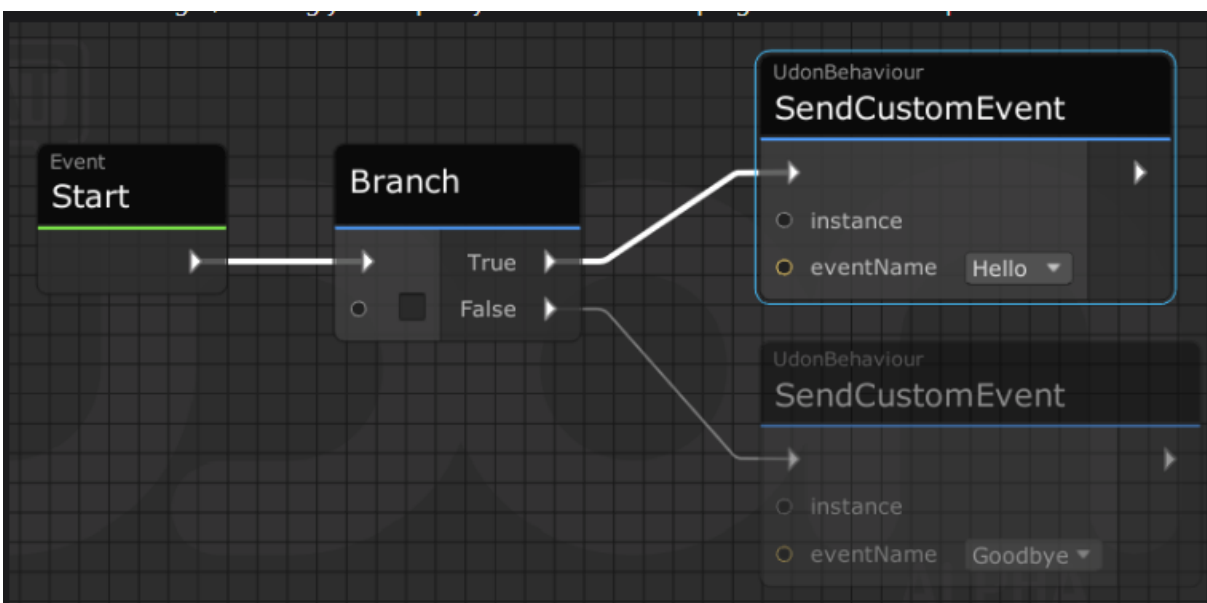


Imagen 6. Ejemplo del uso de nodos y noodles

Cuando el comportamiento que se quiere programar es muy complejo el uso de *Udon Graph* puede dificultar la programación de dicho comportamiento, ya que al existir demasiados nodos se puede confundir la información deseada. Este problema se puede solucionar utilizando el SDK “*Udon Sharp*” el cual permite programar utilizando una sintaxis muy similar a la utilizada en C#, esto facilita la creación de ciertos eventos o funciones y permite distinguir con mayor facilidad las variables y los eventos que se están utilizando.

Además, *Udon Sharp* permite agregar librerías que permiten hacer más eficiente el desarrollo de un código. Para la realización de este proyecto se utilizó la librería *VUdon – Array Extensions* la cual agrega distintos métodos compatibles con los arreglos de *Udon* que permiten funciones adicionales para facilitar el uso de estos elementos. [21]

3.2 Elementos básicos



Imagen 7. Exterior del laboratorio virtual

En la *Imagen 7* se observa el exterior del laboratorio virtual, el usuario puede salir del laboratorio y moverse alrededor de este, en esta versión las mesas y sillas que se observan son objetos decorativos y no se puede interactuar con ellas.

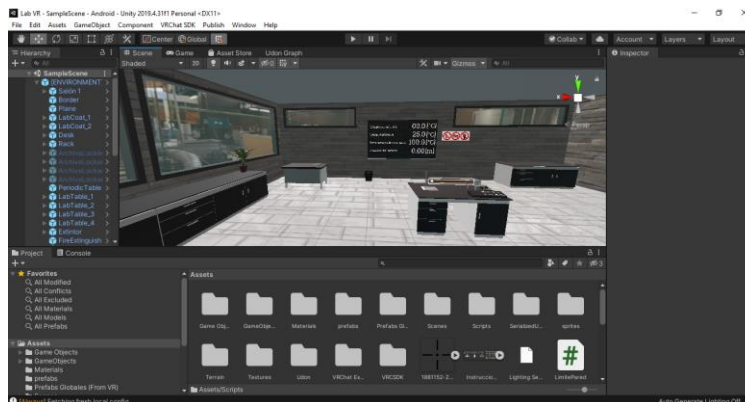


Imagen 8. Interior del Laboratorio virtual en Unity

En la *Imagen 8* se puede observar el área principal del entorno virtual donde se lleva a cabo el experimento. Como se puede observar en el árbol de jerarquías de *Unity* la escena incluye varios objetos los cuales forman el laboratorio virtual.

La mayoría de los objetos presentes son decorativos y su función es permitir que el usuario sienta que se encuentra en un laboratorio real. Los elementos más importantes de este entorno son el pizarrón que se encuentra en el fondo y la mesa donde se encuentra la barra, estos elementos contienen scripts que permiten realizar el experimento. A continuación, se describirá de manera detallada el funcionamiento de dichos elementos.

3.2.1 Pizarrón

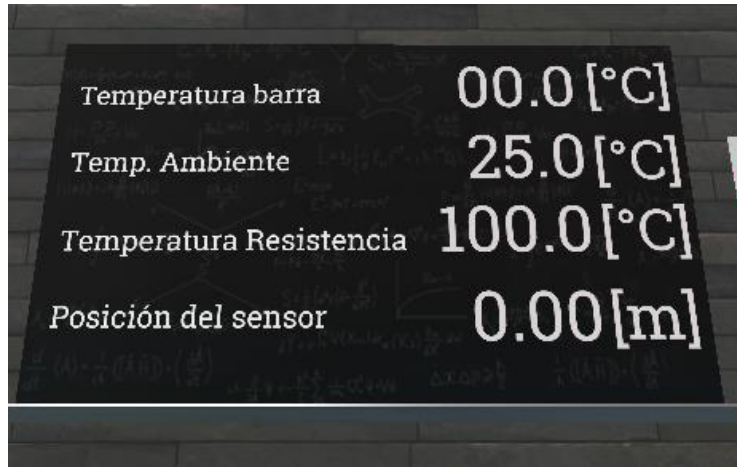


Imagen 9. Pizarrón donde se observan los resultados del experimento

El propósito del pizarrón observado en la *Imagen 9* es permitir a los usuarios visualizar los resultados del experimento, los datos que se obtienen de dicho experimento son los siguientes:

- **Temperatura barra:** Indica la temperatura de la barra en un punto específico, este valor se actualiza al mover el sensor. Este punto corresponde a la posición del sensor. La temperatura es calculada utilizando la siguiente fórmula²:

$$\theta = \frac{\theta_r \operatorname{senh}[L - x] + \theta_h \operatorname{senh}[mx]}{\operatorname{senh}[mL]}$$

- **Temp. Ambiente:** Temperatura ambiente en °C, la cual puede ser modificada utilizando los botones correspondientes
- **Temperatura Resistencia:** Temperatura de la resistencia en °C, la cual puede ser modificada utilizando los botones correspondientes
- **Posición del sensor:** Distancia a la que se encuentra el sensor con relación a la resistencia. Este valor se actualiza al mover el sensor.

² Las variables utilizadas en esta fórmula fueron descritas anteriormente en el capítulo 2.3.1

3.2.2 Experimento



Imagen 10. Área donde se realiza el experimento

El elemento principal del experimento es la barra de metal, en su extremo izquierdo se encuentra una resistencia, en este punto la temperatura de la barra es igual a la temperatura de la resistencia. En el extremo derecho de la barra se encuentra un enfriador con temperatura de 0 [°C].

Para obtener la temperatura en algún punto específico de la barra se utiliza un sensor de temperatura que se desliza horizontalmente a lo largo de la barra. Este sensor está representado en el laboratorio virtual con un *slider*³.

Otra de las características del experimento es la capacidad de cambiar el material de la barra de metal, lo cual permite obtener la relación distancia-temperatura que existe entre distintos materiales.

3.2.3 Panel de control



Imagen 11. Botones que componen al panel de control del experimento

³ El funcionamiento de dicho elemento se describirá con mayor detalle en el capítulo 3.3.1.

El panel de control del experimento consiste en una botonera que permite cambiar el material de la barra de metal y unos botones que permiten modificar la temperatura ambiental y la temperatura de la resistencia de manera que el experimento se pueda realizar en distintas condiciones. Para modificar estos elementos solo es necesario interactuar haciendo clic sobre los botones para obtener la acción deseada. En la siguiente sección se describirá a detalle el funcionamiento de dicha interacción.

3.3 Interacciones básicas

3.3.1 Sensor de Posición



Imagen 12. Slider que representa el sensor de temperatura

El sensor de posición permite obtener la temperatura de la barra en un punto específico, para realizar esta acción se utiliza un elemento llamado “*slider*”. La función del *Slider* es asignar un valor numérico dentro de un rango determinado. El rango del *slider* utilizado es 0 - 1, que representa la longitud de la barra en metros. El valor numérico del *slider* indica la distancia desde la resistencia hasta el punto a analizar, este valor es observado en el pizarrón y permite realizar los cálculos necesarios para encontrar la temperatura en dicho punto.

Se siguen los siguientes pasos para la creación del *Slider*:

1. Colocar un *Canvas*⁴ en la posición deseada, es necesario que el *Render Mode*⁵ sea *World Space*⁶. Ajustar las dimensiones del *Canvas* para obtener el tamaño deseado

⁴ *Canvas* es un tipo de objeto en Unity, se agrega desde el árbol de jerarquías

⁵ Modo de renderizado, se utiliza para renderizar el *Canvas* en el espacio de la pantalla o el espacio del mundo.

⁶ Es un modo de renderizado que permite que el *Canvas* se comporte como cualquier otro objeto en la escena.

2. Dentro del *Canvas* colocar un *Panel*.
3. Dentro del *Panel* colocar el *Slider* y ajustar el tamaño hasta obtener las dimensiones deseadas.
4. Se establecen los parámetros del *Slider*, además se crea un evento (*On Value Changed*) que nos permita actualizar el valor al mover el *Slider*, en este elemento se encuentra el script que permite calcular la temperatura de la barra en un punto específico.

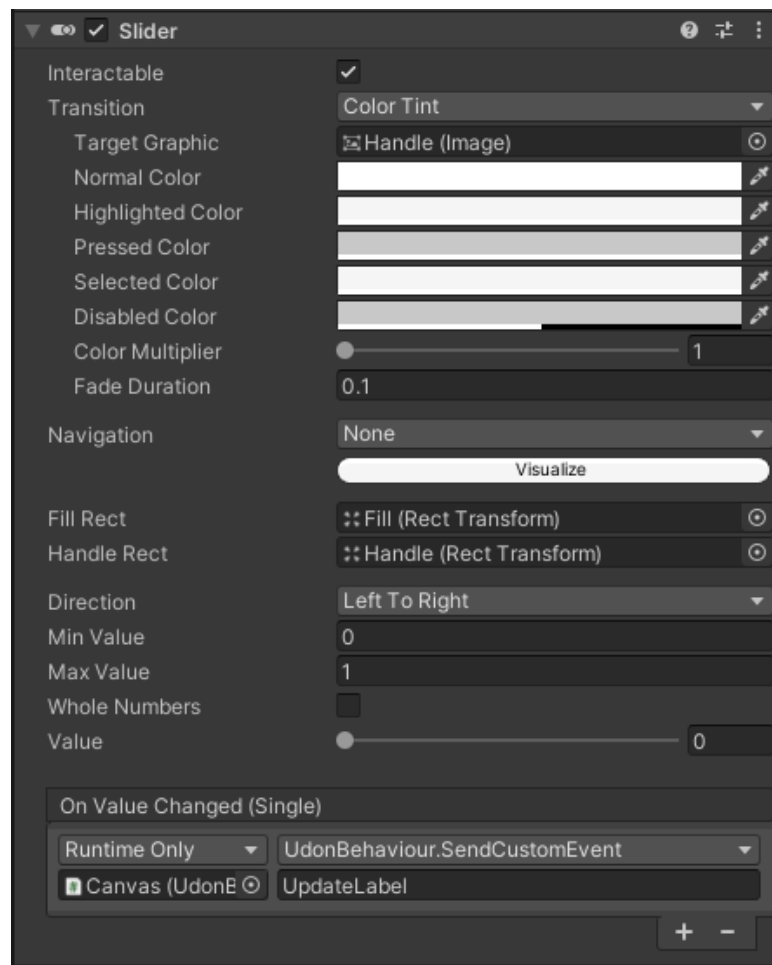


Imagen 13. Parámetros utilizados en el *Slider*

En el elemento *Canvas* se encuentra un script *Udon Behaviour* llamado *SyncSlider*, este script permite obtener la temperatura en el punto donde se encuentra el sensor. El script utilizado es del tipo *UdonGraph*.

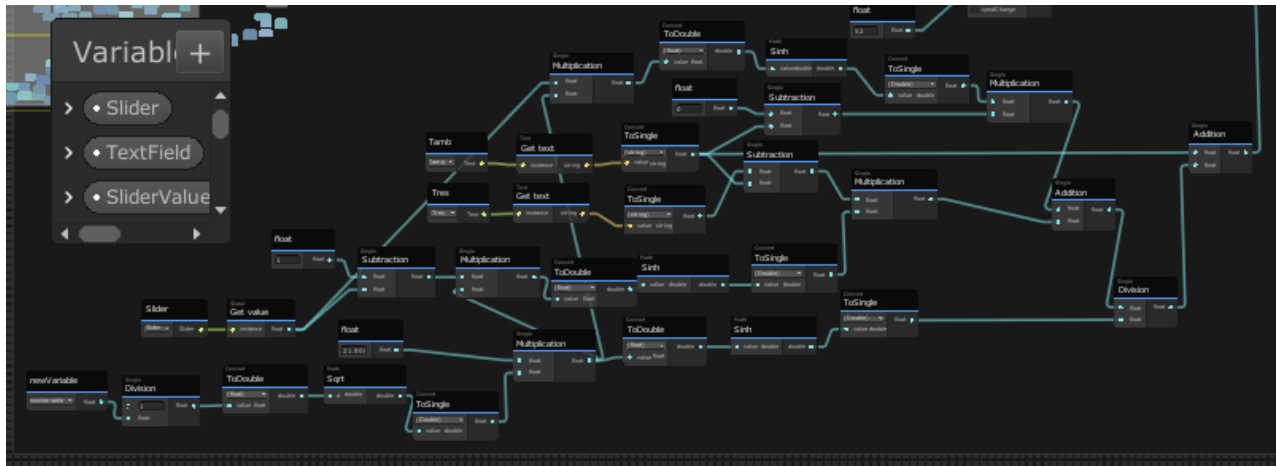


Imagen 14. Primera parte del script SyncSlider

En la *Imagen 14* se observa la primera parte del código, esta corresponde al cálculo de la temperatura de la barra en un punto específico utilizando la fórmula correspondiente.

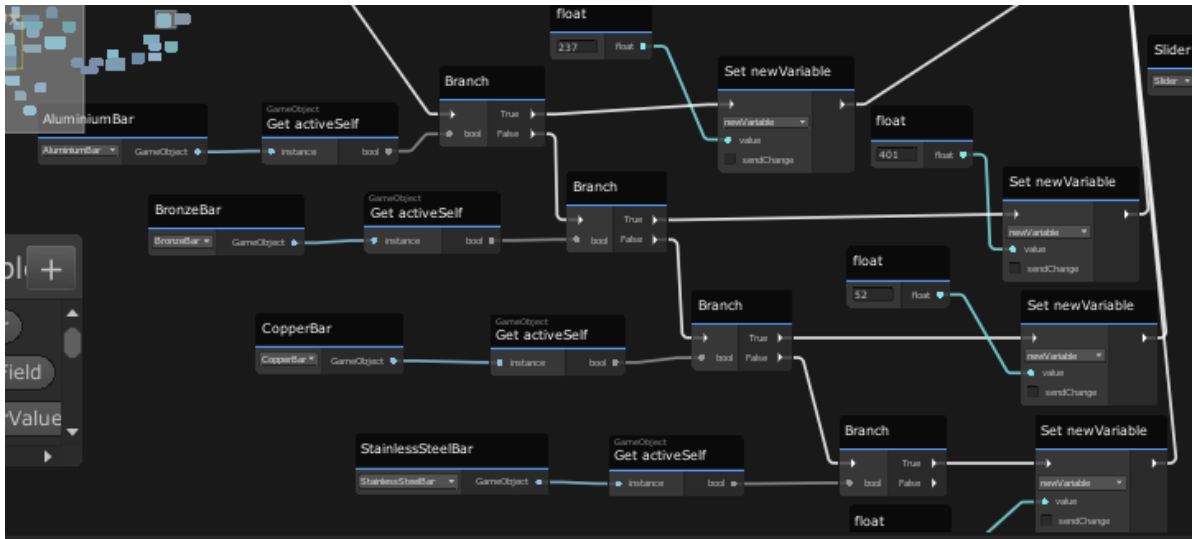


Imagen 15. Segunda parte del script SyncSlider

La segunda parte del *script*, observada en la *Imagen 15*, tiene como propósito obtener el valor de la conductividad térmica del material que se encuentra activo, esta variable es utilizada en la primera parte del código para obtener la temperatura correspondiente, es importante mencionar que esta parte cuenta con un error que impide que esta acción se realice de manera correcta por lo que el valor de la conductividad térmica se mantiene igual para todos los materiales.

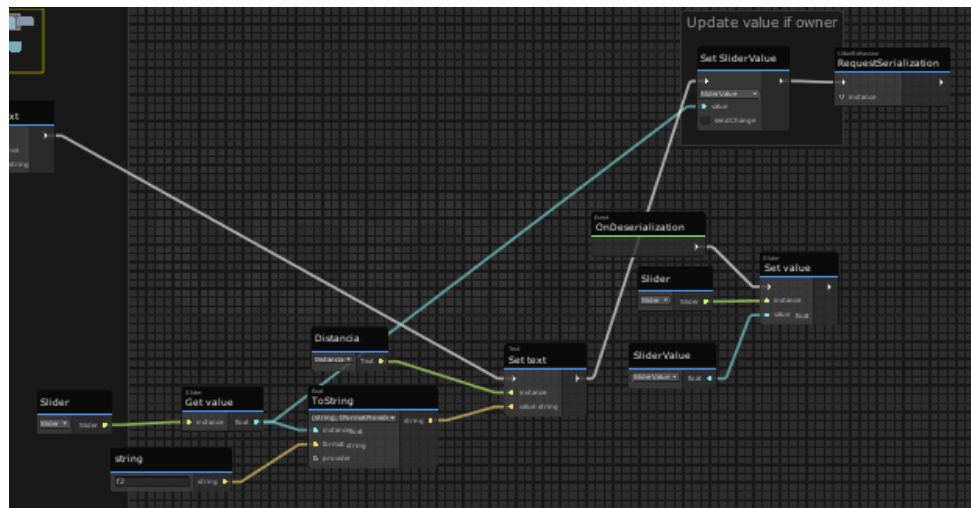


Imagen 16. Tercera parte del script SyncSlider

Finalmente, la *Imagen 16* corresponde a la tercera parte del código la cual permite actualizar el valor de la distancia al mover el *slider*, este valor es utilizado para calcular la temperatura correspondiente. El valor de la distancia actualizado es visible en el pizarrón.

A continuación, se describen las variables utilizadas en el script *SyncSlider*:

- *Slider*: Se trata del *slider* previamente creado, representa el sensor de temperatura
- *TextField*: Variable de texto que nos indica la temperatura de la barra en un punto determinado, este valor se obtiene mediante el script *SyncSlider*.
- *Tamb*: Variable de texto que nos indica la temperatura ambiente, el valor es tomado del pizarrón
- *Tres*: Variable de texto que nos indica la temperatura de la resistencia, el valor es tomado del pizarrón
- *Distancia*: Variable de texto que nos indica la posición del sensor, el valor es tomado del pizarrón
- *Slider_1*: Hace referencia al *slider* previamente creado
- *AluminionBar*: Barra de Aluminio
- *BronzeBar*: Barra de Bronce
- *CopperBar*: Barra de Cobre
- *StainlessSteelBar*: Barra de Acero Inoxidable

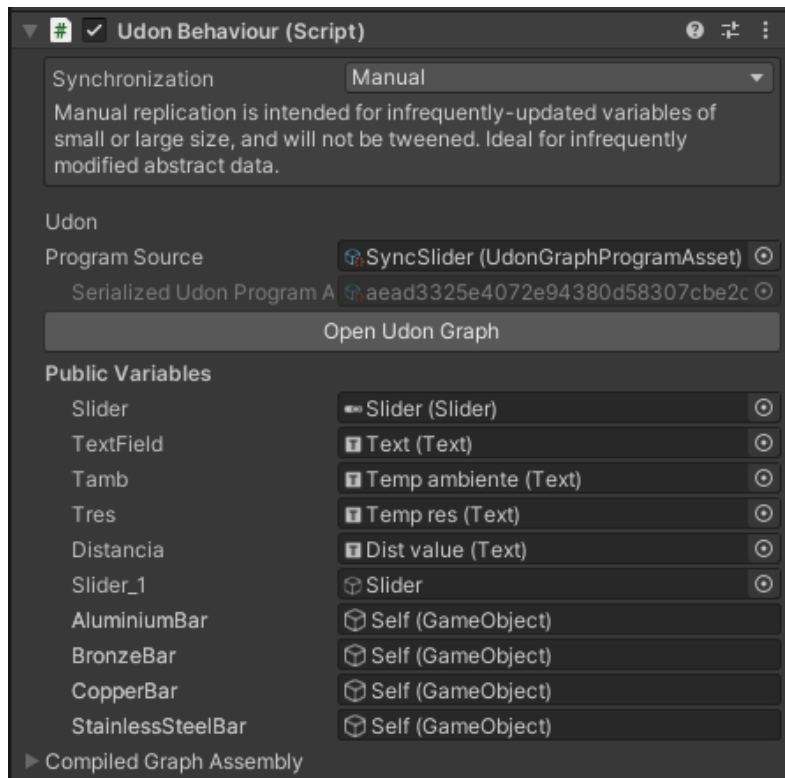


Imagen 17. Variables utilizadas en el script SyncSlider

3.3.2 Cambio de material

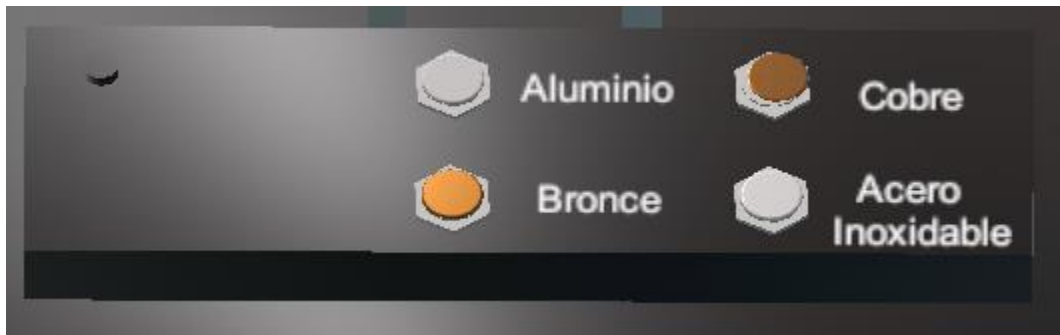


Imagen 18. Botones utilizados para cambiar el material de la barra

Para realizar el cambio de material se utilizan botones, cuando el usuario hace clic en uno de ellos la barra cambia al material correspondiente, en la *Imagen 18* se observan los botones utilizados en el laboratorio virtual. Para conseguir esta acción se utiliza el evento “*Interact*⁷”, al realizar esta acción se activa la barra del material deseado y se desactivan las

⁷ Este evento se agrega un objeto, cuando el usuario hace clic en este se genera la acción deseada.

demás. El código base observado en la *Imagen 19* se utiliza en todos los botones de cambio de material.

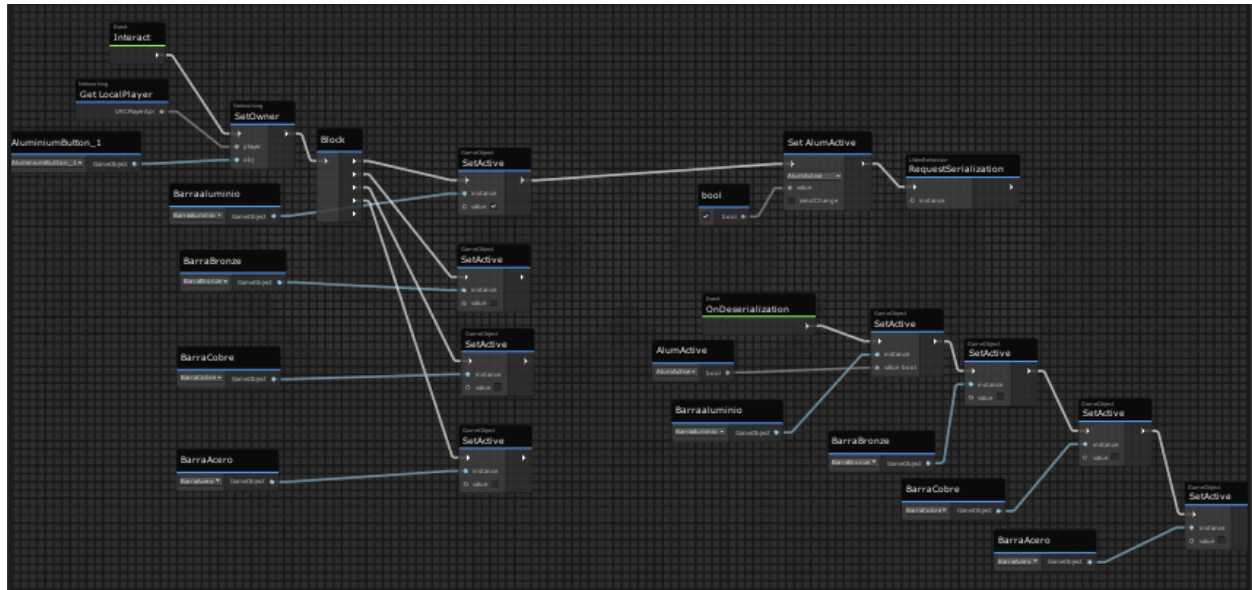


Imagen 19. Código base para el cambio de material

3.3.3 Cambio de temperatura ambiental

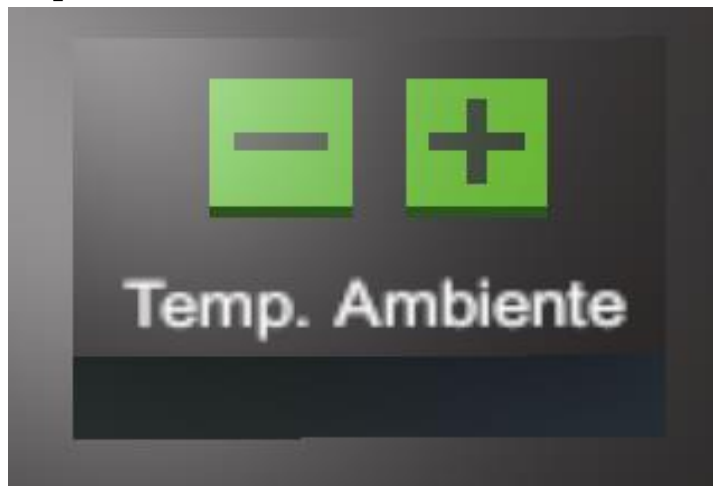


Imagen 20. Botones utilizados para modificar la temperatura ambiental

La temperatura ambiental es una variable que se utiliza para obtener la temperatura de la barra de metal en un punto específico, para realizar el experimento en el laboratorio virtual en distintas condiciones es posible modificar este valor utilizando los botones correspondientes (*Imagen 20*), el botón “+” permite aumentar la temperatura 0.5[°C] por cada clic mientras que el botón “-” la disminuye, para realizar esta interacción se utiliza el

evento “*Interact*”. Se decidió utilizar este intervalo (0.5 [cm]) para modificar la temperatura debido a que nos permite manejar las variables de manera más precisa. Además, es conveniente ya que en el manual de prácticas de química las mediciones se realizan cada 10 [cm] a partir de una distancia de 10.5 [cm], el uso de este intervalo permite replicar mediciones que se realizarían en el laboratorio físico así como agregar puntos extras que le permitan a los alumnos analizar otro punto dentro de la barra que consideren relevante.

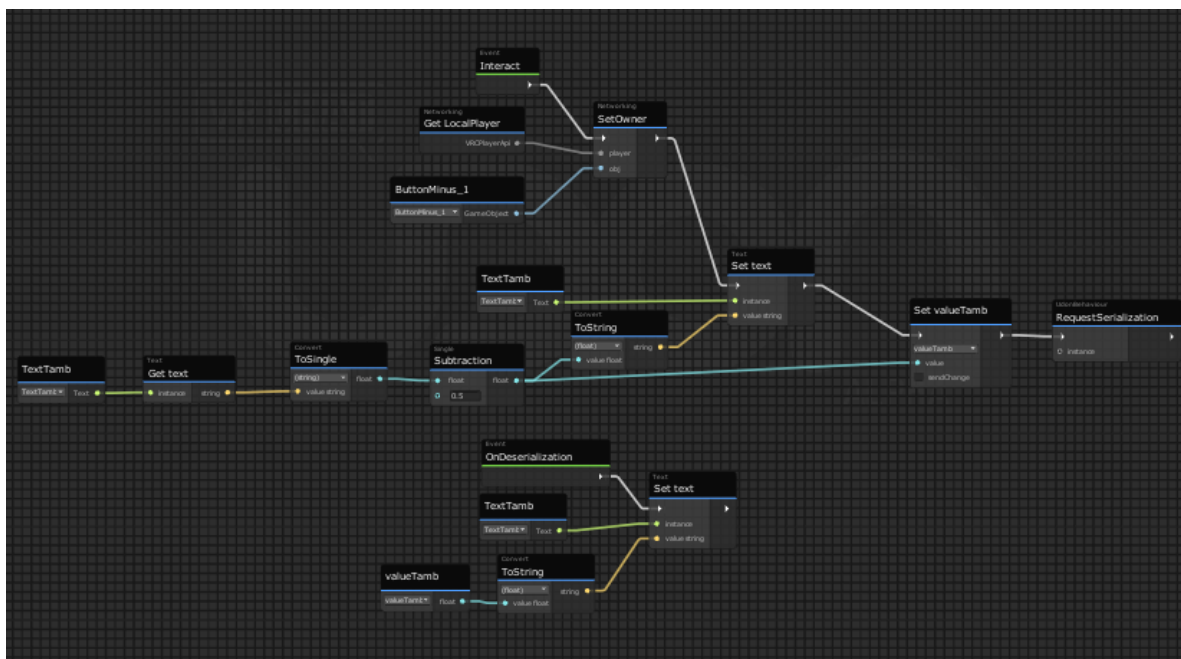


Imagen 21. Código utilizado para disminuir la temperatura

En la *Imagen 21* se puede ver el código utilizado para disminuir la temperatura, se utiliza la función “*Subtraction*” para realizar esta acción, con la función “*Set text*” se modifica el valor de la temperatura ambiental en el pizarrón. Para aumentar la temperatura se utiliza un código similar, en este caso se utiliza la función “*Addition*” en lugar de “*Subtraction*”

3.3.4 Cambio de temperatura de la resistencia

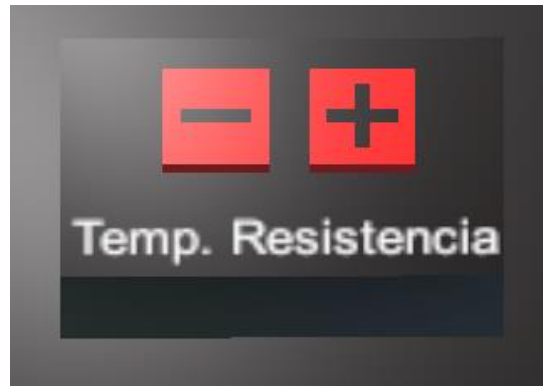


Imagen 22. Botones utilizados para modificar la temperatura de la resistencia

La temperatura de la resistencia también es una variable que se utiliza para calcular la temperatura de la barra en un punto específico, el funcionamiento de estos botones (*Imagen 22*), es igual al descrito previamente para los botones utilizados para modificar la temperatura ambiental, el código utilizado es igual al observado en la *Imagen 21*.

3.4 Identificación de los puntos a mejorar

- Para que el experimento pueda ser realizado de manera correcta es necesario modificar el código actual para permitir que al cambiar el material de la barra también se modifique el valor de la conductividad térmica para que los cálculos pueden ser realizados de manera correcta.
- Agregar el nombre del material a la interfaz del pizarrón con la finalidad de facilitarle al usuario la identificación de los resultados obtenidos al realizar el experimento.
- Agregar interacciones con algunos objetos decorativos para mejorar la experiencia de inmersión, se busca agregar una interacción que le permita al usuario utilizar las sillas que se encuentran fuera del laboratorio virtual.
- Agregar una gráfica con el propósito de brindarle al usuario un apoyo visual que le permita identificar de manera más sencilla los resultados y la relación que existe entre la distancia y la temperatura de la barra. Cada material es representado con un color diferente para permitirle al usuario distinguirlos de manera sencilla y comparar los resultados entre distintos materiales.

4 Desarrollo

4.1 Cambio de material

En el botón correspondiente a cada material se asigna una variable k que corresponde al valor de la conductividad térmica para cada material, este permite que al presionar el botón de cada material este valor se actualice.

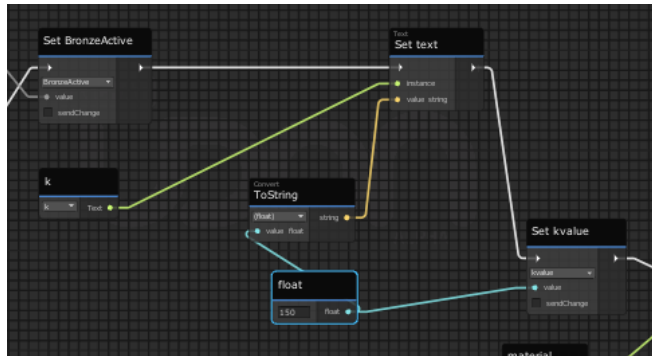


Imagen 23. Asignación de la variable k en cada material

Se modificó el script descrito anteriormente en el capítulo 3.3.1, se modificó la parte dos del código de manera que en lugar de asignar el valor de la conductividad térmica del material desde esa parte esta se obtuviera a partir de otra variable asignada directamente en la barra de metal (Variable k). A esta nueva variable se le llamo Tk (en el código original sustituye a la variable New Variable), al interactuar con el botón correspondiente este valor se modifica de acuerdo con el nuevo material seleccionado. Con esta modificación es posible eliminar la segunda parte del código observada en la *Imagen 15*.

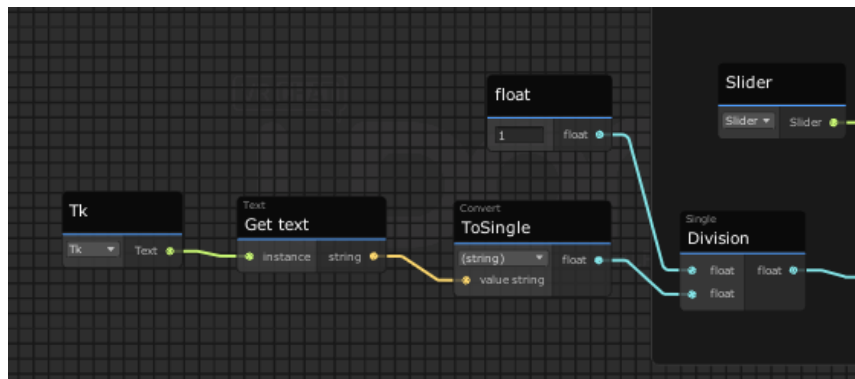


Imagen 24. Ubicación de la variable Tk en el script SyncSlider

Gracias a estos dos cambios es posible obtener los resultados esperados al realizar el experimento en el laboratorio virtual.

4.2 Interfaz del pizarrón

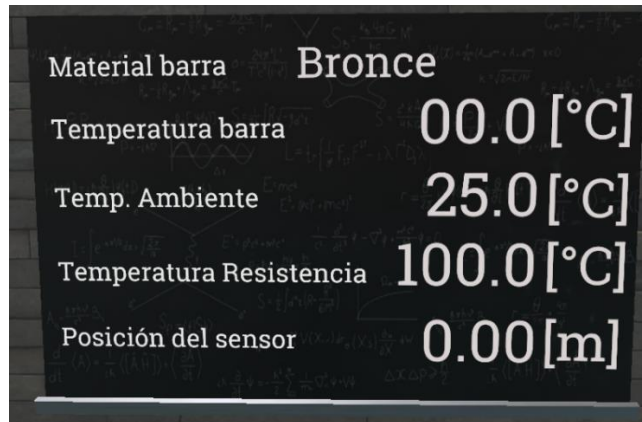


Imagen 25. Versión actualizada del pizarrón

Para visualizar el nombre del material en el pizarrón se agregó una variable de tipo *Text* en el script correspondiente a cada material, esto se repite en todos los botones de cambio de material. El valor de esta variable se asigna al cuadro de texto correspondiente en el pizarrón. Esta nueva interacción continúa el flujo del código original, por lo cual cuando se presiona el botón correspondiente se actualiza el valor en el pizarrón.

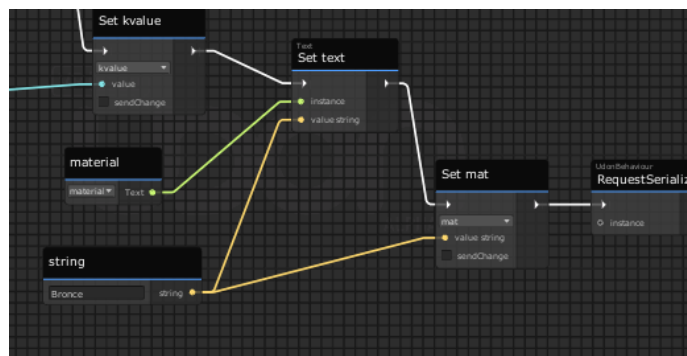


Imagen 26. Asignación de la variable material

4.3 Interacciones con el ambiente

Para mejorar la experiencia de inmersión se agregó una interacción que le permite a los avatares sentarse en las sillas que se encuentran fuera del laboratorio virtual. Para lograr esta interacción solo es necesario agregar el componente “*VRC Station*” a cada silla. Es necesario ajustar la posición (coordenadas y orientación) de este componente para que la animación ocurra de manera correcta, como se puede observar en la *Imagen 27*, de no ajustar de manera adecuada este elemento el avatar puede aparecer desplazado de la silla o en la

orientación incorrecta. El elemento utilizado para ajustar la posición es *Seat2(Transform)*. El ajuste varía para cada elemento y es necesario probarlo hasta obtener el resultado deseado.



Imagen 27. Interacción del avatar del usuario con las sillas.

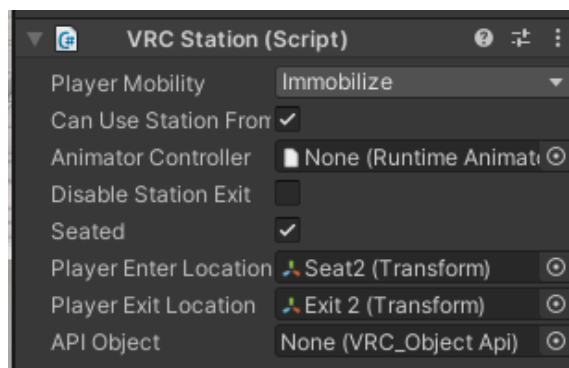


Imagen 28. Componente VRC Station.

4.4 Gráfica Distancia – Temperatura

Para permitirle a los usuarios visualizar de mejor manera la relación distancia-temperatura que existe al realizar el experimento se agregó un espacio que permite ver de manera gráfica dicha relación. Como se puede observar en la *Imagen 29* se pueden agregar diversos puntos que representan a cada material, en la gráfica es posible observar y comparar las diferencias que existe al realizar el experimento con cada material.

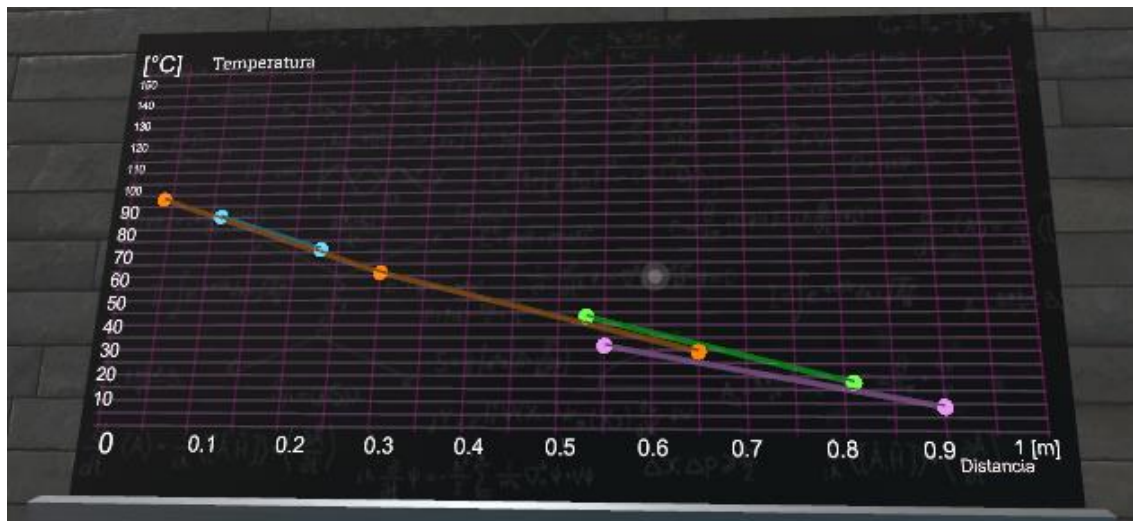


Imagen 29. Gráfica Distancia-Temperatura

Para generar los puntos en la gráfica se agregaron dos botones: *Agregar Punto* y *Unir punto*

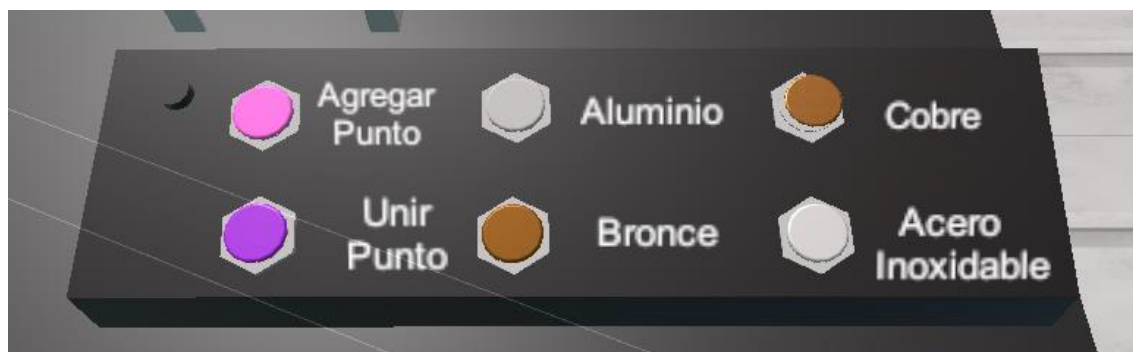


Imagen 30. Botones agregados en la nueva versión del laboratorio

4.4.1 Descripción de los elementos que componen la gráfica

A continuación, se hará una breve descripción de los elementos que permiten generar los puntos deseados en la gráfica.

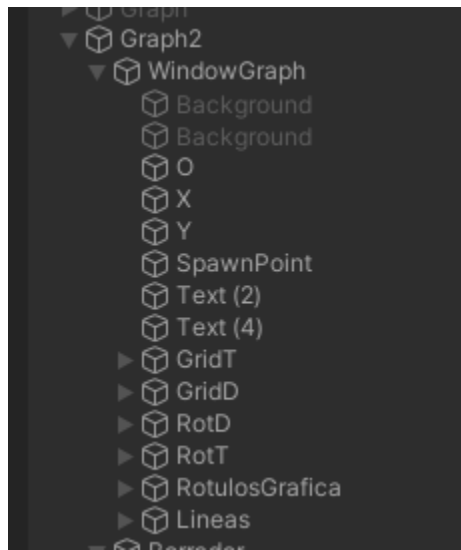


Imagen 31. Objetos que componen a la gráfica

- *WindowsGraph*: Es el área donde se encuentra la gráfica, se compone de dos elementos tipo *Panel* colocados en *Canvas*
- *O*: Punto de origen de la gráfica, se encuentra en las coordenadas (0,0)
- *X*: Indica el punto final del eje X de la gráfica, se encuentra en las coordenadas (0,1)
- *Y*: Indica el punto final del eje Y de la gráfica, se encuentra en las coordenadas (150,0)
- *SpawnPoint*: *EmptyObject* que permite generar el punto deseado, se encuentra en el centro de la gráfica
- *Lineas*: Grupo de objetos del tipo *LineRenderer*, este componente permite generar un arreglo de dos o más puntos en un espacio 3D y crea una línea entre cada punto. [22] Existe una línea por cada material, como se observa en la *Imagen 29*.
- El resto de los elementos corresponden a las líneas tabuladoras de la gráfica y sus rótulos, tienen una función decorativa y de guía para distinguir las coordenadas de cada punto.
- *Points*: Además de estos elementos se utilizó un elemento del tipo *Sprite* en forma de círculo para crear los puntos de la gráfica, se creó un punto para cada material asignándoles diferentes colores y nombres para poder distinguirlos, estos 4 elementos

(PointAcero, PointBronce, PointCobre y PointAluminio) se movieron a la carpeta prefabs⁸ y se les asignó el tag Point a cada uno

4.4.2 Interacción para añadir un punto a la gráfica

Para agregar un punto nuevo a la gráfica se creó el script *DatoButtonScript.cs*, debido a la complejidad del código se utilizó Udon Sharp para la programación. A continuación, se explican los elementos más importantes de dicho script.

- Declaración de variables: En esta parte del código se declaran todas las variables que permiten identificar los puntos importantes de la gráfica, así como instanciar los elementos deseados.

```
//Comunicación con el script BorrarDatos
public UdonBehaviour BorrarScript;
public BorrarDatos BorrarDatoScript;

//Sprites de los puntos a generar
public GameObject PointBronce;
public GameObject PointAluminio;
public GameObject PointAcero;
public GameObject PointCobre;

//Empty Objects que permiten determinar el origen y los ejes de la gráfica
public GameObject SpawnPoint;
public GameObject O;
public GameObject Xa;
public GameObject Va;

// Array para los puntos generados de cada material
public GameObject[] GraphBronce;
public GameObject[] GraphAluminio;
public GameObject[] GraphAcero;
public GameObject[] GraphCobre;

//Variable para identificar la interacción borrar
public bool eliminar;

//Variables necesarias para localizar el punto en la gráfica
[UdonSynced]public float dis;
[UdonSynced] public float temp;
[UdonSynced] public float XSP;
[UdonSynced] public float YSP;
[UdonSynced] public float XAxis;
[UdonSynced] public float YAxis;

//Valores observados en el pizarrón
public Text temperaturaText;
public Text distanciaText;
public Text materialText;
```

Imagen 32. Variables declaradas

- Método Update(): Permite identificar si es necesario borrar los elementos presentes en la gráfica, es decir, si se ha interactuado con el “borrador”. Esta información la

⁸ El prefab actúa como una plantilla a partir de la cual se pueden crear nuevas instancias del objeto en la escena. [23]

obtiene con ayuda de una variable booleana (*borrar*) que se encuentra en el script *BorrarDatos.cs*, este valor es asignado a la variable *eliminar*. Si la variable *eliminar* es verdadera los arreglos donde se almacenan los puntos de la gráfica se reinician a cero elementos.

```
public void Update()
{
    //Permite identificar el estado de la variable del tipo bool "borrar" del script BorrarDatos, este valor es asignado a la variable eliminar
    eliminar = (bool)BorrarScript.GetProgramVariable("borrar");
    eliminar = BorrarDatoScript.borrar;

    //Si la variable eliminar es verdadera el tamaño de los Array donde se almacenan los puntos se actualiza a 0
    if (eliminar == true)
    {
        GraphBronce = GraphBronce.Resize(0);
        GraphAcero = GraphAcero.Resize(0);
        GraphAluminio = GraphAluminio.Resize(0);
        GraphCobre = GraphCobre.Resize(0);
    }
}
```

Imagen 33. Método Update()

- Método Start(): Permite identificar la distancia de los ejes X y Y, la calcula a partir de los objetos “O”, “X” y “Y” de la gráfica.

```
void Start()
{
    //Permite identificar la dsitancia de los ejes X y Y
    XAxis = Vector3.Distance(O.transform.position, Xa.transform.position);
    YAxis = Vector3.Distance(O.transform.position, Ya.transform.position);
}
```

Imagen 34. Método Start()

- Método Interact(): Esta función se realiza cuando el usuario hace clic en el botón Agregar Punto, permite iniciar con el método AddDato(), utilizando la función *SendCustomNetworkEvent()* el método se encuentra disponible para todos los usuarios presentes en el laboratorio virtual.

```
public override void Interact()
{
    //Cuando se interactua con el boton "Agregar Punto" se inicia el metodo AddDato()
    //Este evento se encuentra disponible para todos los usuarios en el Network
    SendCustomNetworkEvent(VRC.Udon.Common.Interfaces.NetworkEventTarget.All, "AddDato");
}
```

Imagen 35. Método Interact()

- Método AddDato(): Obtiene el valor que se encuentra visible en el pizarrón y transforma la variable a un elemento del tipo *float* para obtener el valor numérico de la distancia y la temperatura, después de obtener estos valores permite iniciar el método *CreatePoint()*.

```
public void AddDato()
{
    //Se transforma el valor del cuadro de texto del pizarrón a una variable del tipo float
    dis = float.Parse(distanciaText.text);
    temp = float.Parse(temperaturaText.text);

    //Se crean el punto en la gráfica utilizando el metodo CreatePoint()
    CreatePoint();
}
```

Imagen 36. Método AddDato()

- Método CreatePoint(): Con los valores de distancia y temperatura obtenidos en la función *AddDato()* se crea el punto deseado, para lograr esto se ajustan los valores obtenidos al correspondiente a la distancia de los ejes. Se utiliza el elemento *SpawnPoint* para instanciar el punto correspondiente, la función *new Vector3(x, y)* permite que este punto se encuentre en la posición deseada.

Para generar el punto del material deseado se utiliza la función *if()*, la cual se repite para cada material. Con esta función se identifica el material de la barra a partir del nombre observado en el pizarrón y se instancia el punto correspondiente en la posición del *SpawnPoint* previamente generado, además se agrega este punto a un arreglo que contiene todos los puntos correspondientes a cada material. La función *AddUnique()* se utiliza para agregar los elementos al arreglo y evitar que estos se repitan.

```
private void CreatePoint()
{
    //Se ajusta el valor observado en el pizarrón a una escala correspondiente en los ejes de la gráfica
    XSP = (XAxis / 1f) * dis;
    YSP = (YAxis / 150f) * temp;

    //Se establecen las coordenadas donde se va a generar el punto creado (SpawnPoint)
    SpawnPoint.transform.position = new Vector3(0.transform.position.x - XSP, 0.transform.position.y + YSP, 0.transform.position.z);

    if (materialText.text=="Bronce") //Se identifica el material para generar el punto adecuado
    {
        var graphPoint = Object.Instantiate(PointBronce); //Se crea el punto del material correspondiente
        graphPoint.transform.position = SpawnPoint.transform.position; // La posición del punto se iguala al SpawnPoint previamente generado
        GraphBronce= GraphBronce.AddUnique(graphPoint); //El punto generado se agrega al Array correspondiente, AddUnique evita tener valores repetidos
    }
}
```

Imagen 37. Método CreatePoint()

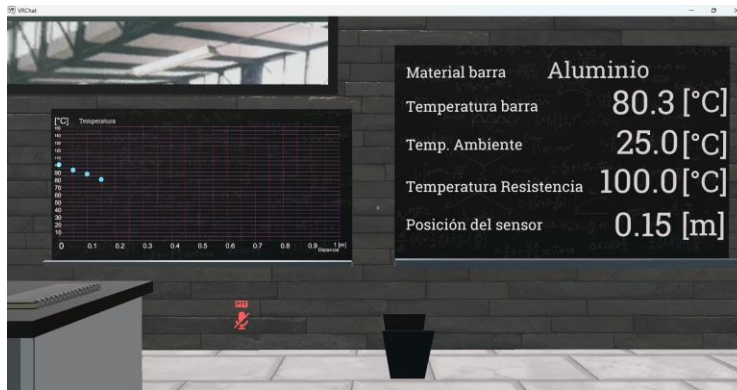


Imagen 38. Puntos generados utilizando el botón "Agregar Punto"

4.4.3 Interacción para unir los puntos de la gráfica

Después de generar los puntos deseados se utiliza el botón Unir Punto para unirlos, esta interacción permite generar líneas diferentes para cada material sin importar el orden en que fueron agregados los puntos, para lograrlo se utiliza el elemento *LineRenderer*. Si después de generar las líneas correspondientes se agregan más puntos es posible volver a presionar el botón para ajustar la línea de manera que los nuevos puntos también se encuentren unidos y sea más sencillo distinguir el comportamiento de la gráfica. Para lograr esta interacción se utiliza el script *GraphPoint.cs*.

- Declaración de variables: Se declaran las variables necesarias para distinguir los puntos que componen la gráfica y generar las líneas para unirlos.

```
//Comunicación con otros scripts (DatoButtonScript y BorrarDatos)
public UdonBehaviour DButtonScript;
public DatoButtonScript ThedatoButtonScript;
public UdonBehaviour BorrarScript;
public BorrarDatos BorrarDatosScript;

//Arrays de los puntos visualizados en la gráfica
public GameObject[] Bronce;
public GameObject[] Aluminio;
public GameObject[] Acero;
public GameObject[] Cobre;

//Variable para identificar la interacción borrar
public bool eliminar;

//Variables para identificar si existe algún punto de cada material
public bool bronceE;
public bool aluminioE;
public bool aceroE;
public bool cobreE;

//Variables para identificar si todos los puntos han sido eliminados
public bool fin=false;

//LineRenderer utilizados para unir los puntos de cada material
public LineRenderer lrCobre;
public LineRenderer lrBronce;
public LineRenderer lrAluminio;
public LineRenderer lrAcero;
```

Imagen 39. Variables declaradas

- Método Interact(): Al interactuar con el botón Unir Punto se inicia el método *CreateGraph()*, este evento se encuentra disponible para todos los usuarios presentes en el laboratorio virtual.

```
public override void Interact()
{
    //Cuando se interactua con el boton "Unir Punto" se inicia el metodo CreateGraph()
    //Este evento se encuentra disponible para todos los usuarios en el Network
    SendCustomNetworkEvent(VRC.Udon.Common.Interfaces.NetworkEventTarget.All, "CreateGraph");
}
```

Imagen 40. Método Interact()

- Método Update(): Obtiene el valor de la variable booleana *borrar* a partir del script *BorrarDatos.cs*, este valor es asignado a la variable *eliminar*. Si la variable *eliminar* es verdadera da inicio al método *BorrarDatos()*, además se verifica que todos los puntos de la gráfica hayan sido eliminados, de ser así la variable *fin* se vuelve verdadera.

```
public void Update()
{
    //Permite identificar el estado de la variable del tipo bool "borrar" del script BorrarDatos, este valor es asignado a la variable eliminar
    eliminar = (bool)BorrarScript.GetProgramVariable("borrar");
    eliminar = BorrarDatosScript.borrar;

    //Si la variable eliminar es verdadera se eliminan los puntos generados, esto se realiza en método BorrarDatos()
    if (eliminar == true)
    {
        BorrarDatos();

        //Se verifica que todos los puntos hayan sido eliminados, de ser así la variable fin es verdadera
        if (bronceE == false & aceroE == false & cobreE == false & aluminioE == false)
        {
            fin = true;
        }
        else
            fin = false;
    }
}
```

Imagen 41. Método Update()

- Método CreateGraph(): A partir de los arreglos generados para cada material en el script *DatoButtonScript.cs* se crea un arreglo nuevo para cada material con los mismos elementos. Se utiliza la función *if()*, la cual se repite para cada material, para comprobar que existan al menos dos elementos en los arreglos recién creados, de ser así se da inicio al método *OrdenarPuntos(,)*, el cual utiliza como argumentos el arreglo y el nombre correspondiente a cada material.

```

public void CreateGraph()
{
    //Se obtiene el Array con los puntos generados en el script "DatoButtonScript" para cada material
    Bronce = (GameObject[])DButtonScript.GetProgramVariable("GraphBronce");
    Bronce = ThedataButtonScript.GraphBronce;

    Acero = (GameObject[])DButtonScript.GetProgramVariable("GraphAcero");
    Acero = ThedataButtonScript.GraphAcero;

    Aluminio = (GameObject[])DButtonScript.GetProgramVariable("GraphAluminio");
    Aluminio = ThedataButtonScript.GraphAluminio;

    Cobre = (GameObject[])DButtonScript.GetProgramVariable("GraphCobre");
    Cobre = ThedataButtonScript.GraphCobre;

    //Se identifica que cada material cuente con más de un valor en el arreglo correspondiente
    //De ser así, se ordenan los puntos en el método OrdenarPuntos()
    //El método OrdenarPuntos() requiere el Array donde se encuentran los puntos y el nombre de material correspondiente
    if (Bronce.Length > 1)
    {
        OrdenarPuntos(Bronce,"Bronce");
    }
}

```

- Método OrdenarPuntos(,): Este método utiliza como argumentos el arreglo que contiene los puntos de la gráfica y la cadena que indica el nombre del material. Se utiliza el ordenamiento Burbuja para ordenar los puntos de la gráfica de menor a mayor dependiendo de su distancia al eje x. Una vez que los elementos del arreglo han sido ordenados se da inicio al método *CreateLine(,)*, el cual utiliza como argumentos el arreglo ordenado y el nombre del material.

```

public void OrdenarPuntos(GameObject[] Graph, string material)
{
    //Se utiliza el método de ordenamiento Burbuja para ordenar el arreglo de cada material de menor a mayor distancia (Posición del sensor)
    for (int x = 0; x < Graph.Length; x++)
    {
        for (int i = 0; i < Graph.Length - 1; i++)
        {
            int j = i + 1;

            if (Graph[i].transform.position.x > Graph[j].transform.position.x)
            {
                Vector3 temporal = Graph[j].transform.position;
                Graph[j].transform.position = Graph[i].transform.position;
                Graph[i].transform.position = temporal;
            }
        }
    }

    //Cuando el Array ha sido ordenado se utiliza el método CreateLine() para generar la línea que une los puntos
    //El método CreateLine() requiere el Array donde se encuentran los puntos ya ordenados y el nombre de material correspondiente
    CreateLine(Graph,material);
}

```

Imagen 42. Método OrdenarPuntos()

- Método CreateLine(,): Este método utiliza como primer argumento el arreglo ordenado que contiene los puntos de la gráfica y el nombre del material, utilizando la función *switch()* se identifica el material para utilizar la línea correspondiente, cada *case* utilizado en el *switch* depende del material, en cada *LineRenderer* se iguala el número de posiciones al tamaño del arreglo correspondiente utilizando *positionCount* y posteriormente la posición de cada vértice de la línea se iguala a la de cada punto

de la gráfica que compone al arreglo utilizando *SetPosition(,)*. Esto se repite para cada material.

```
private void CreateLine(GameObject[] Graph, string material)
{
    //Las líneas son generadas utilizando un LineRenderer
    //Se identifica el nombre del material para generar la línea correspondiente.
    switch (material)
    {
        case "Bronce":
            //Se iguala el numero de posiciones de cada LineRenderer al tamaño del Array correspondiente
            lrBronce.positionCount = Graph.Length;
            //La posición de cada vértice de la línea se iguala a la posición de cada punto del Array correspondiente
            for (int i = 0; i < Graph.Length; i++)
            {
                lrBronce.SetPosition(i, Graph[i].transform.position);
            }
            break;
    }
}
```

Imagen 43. Método *CreateLine()*

- **Método *BorrarDatos()*:** Este método permite eliminar todos los elementos de la gráfica, para cada material se iguala el número de posiciones del *LineRenderer* correspondiente a cero, posteriormente se comprueba si el arreglo que contiene los puntos de la gráfica tiene al menos un elemento utilizando la función *if()*, de ser así se crea un nuevo arreglo son elementos del tipo cadena de la misma longitud que el arreglo que contiene los elementos de la gráfica, cada elemento de este nuevo arreglo lleva la cadena "*PointMaterial(Clone)*" dependiendo del material correspondiente como se ve en la *Imagen 44*.

Posteriormente se utiliza la función *foreach(in)* para identificar cada elemento que compone el arreglo, si el elemento "*PuntoMaterial(Clone)*" existe este se elimina y la variable booleana "*materialE*" es verdadera, esto se repite hasta que todos los puntos del arreglo han sido eliminados, cuando esto sucede la variable "*materialE*" se vuelve falsa.

```
public void BorrarDatos()
{
    // Eliminar gráfica Bronce
    lrBronce.positionCount = 0; //Se iguala el numero de posiciones de cada LineRenderer a 0, esto permite eliminar la línea
    if (Bronce.Length > 0)
    {
        //Se crea un Array del tipo string del mismo tamaño que el Array donde se encuentran los puntos de cada material
        string[] PointsBronce = new string[Bronce.Length];

        //A cada espacio del Array del tipo string se le asigna el nombre del GameObject que se quiere eliminar, es decir "PuntoMaterial(Clone)"
        for (int i = 0; i < Bronce.Length; i++)
        {
            PointsBronce[i] = "PointBronce(Clone)";
        }

        //Se identifican todos los GameObjects con el nombre correspondiente
        foreach (string p in PointsBronce)
        {
            GameObject point = GameObject.Find(p); //Nos permite encontrar el GameObject con el nombre "PuntoMaterial(Clone)"
            //Se identifica si "PuntoMaterial(Clone)" existe, esto se asigna a la variable "materialE"
            if (point)
            {
                bronceE = true;
                Destroy(point.gameObject); //Si "PuntoMaterial(Clone)" existe se elimina dicho objeto
            }
            else
                bronceE=false;
        }
    }
}
```

Imagen 44. Método *BorrarDatos()*

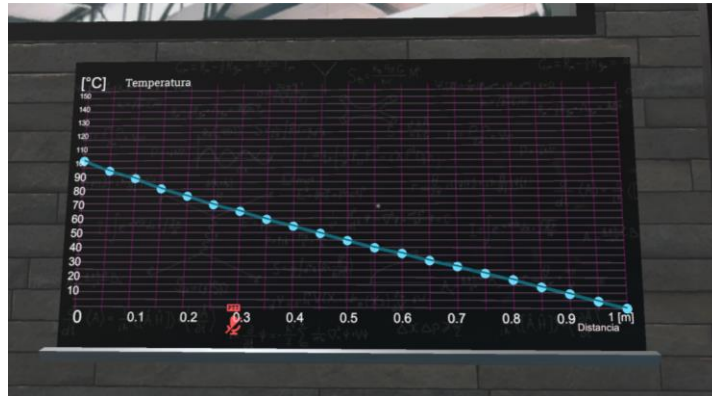


Imagen 45. Línea generada utilizando el botón “Unir Punto”

4.4.4 Interacción para eliminar la gráfica

Al interactuar con el elemento “borrador” se eliminan todos los elementos de la gráfica. El borrador funciona de la misma manera que el resto de los botones, el evento deseado ocurre cuando el usuario hace clic sobre él. Este elemento utiliza el script *BorrarDatos.cs*.

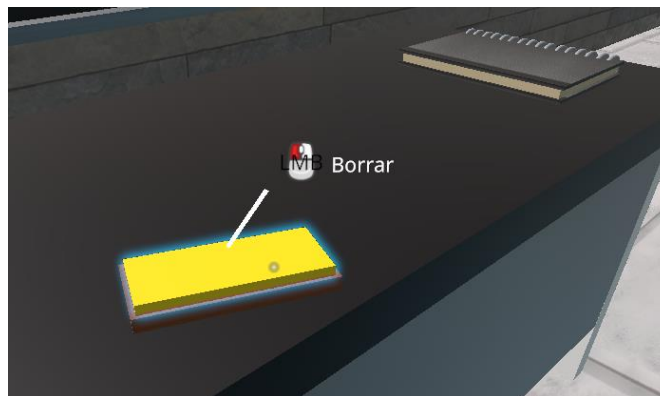


Imagen 46. “Borrador” utilizado para eliminar los elementos de la gráfica

- Declaración de variables: Se declaran las variables necesarias para eliminar los elementos de la gráfica

```
//Comunicación con otros scripts (GraphPoint)
public UdonBehaviour GraphScript;
public GraphPoints GraphButtonScript;

//Variable para identificar la interacción borrar
public bool borrar;

//Variable para identificar si se han eliminado todos los puntos
public bool end;
```

Imagen 47. Declaración de variables

- Método Interact(): Al hacer clic sobre el “borrador” se inicia el método *BorraDatos()*, el cual se encuentra disponible para todos los usuarios del laboratorio virtual

```
public override void Interact()
{
    //Cuando se interactua con el "Borrador" se inicia el metodo BorraDatos()
    //Este evento se encuentra disponible para todos los usuarios en el Network
    SendCustomNetworkEvent(VRC.Udon.Common.Interfaces.NetworkEventTarget.All, "BorraDatos");
}
```

Imagen 48. Método Interact()

- Método Update(): Del script *GraphPoint.cs* se obtiene la variable booleana “*fin*”, este valor es asignado a la variable “*end*”, si es verdadero la variable “*borrar*” es falsa.

```
public void Update()
{
    //Permite identificar el estado de la variable del tipo bool "fin" del script GraphPoint, este valor es asignado a la variable end
    end = (bool)GraphScript.GetProgramVariable("fin");
    end = GraphButtonScript.fin;

    //Si la variable fin es verdadera la variable borrar es falsa
    if (end == true)
    {
        borrar = false;
    }
}
```

Imagen 49. Método Update()

- Método BorraDatos(): La variable “*borrar*” se vuelve verdadera y su valor es utilizado en los scripts *DatoButtonScript.cs* y *GraphPoint.cs* para eliminar los elementos de la gráfica y reiniciar el valor de los arreglos correspondientes.

```
0 referencias
public void BorraDatos()
{
    //La variable borrar se vuelve verdadera, este valor utilizado en los scripts GraphPoint y DatoButtonScript
    borrar = true;
}
```

Imagen 50. Método BorraDatos()

5 Resultados

5.1 Pruebas realizadas

Para validar la efectividad del laboratorio virtual realizado se tomaron mediciones en la simulación para cada material cada 5 [cm], el valor obtenido se comparó con el valor teórico esperado y se obtuvo el porcentaje de error. Los parámetros utilizados fueron Temp. Ambiente = 25[°C] y Temp. De la Resistencia = 100[°C]. Los resultados de esta prueba pueden observarse en la *Tabla 1*.

Se tomó la decisión de realizar las mediciones cada 5 [cm] debido a que es un intervalo lo suficientemente pequeño para obtener valores precisos que nos permiten analizar la calidad de la simulación, pero es lo suficientemente grande como para no saturar la tabla de resultados y facilitar su interpretación.

Tabla 1. Resultados de la prueba realizada

Resultados Laboratorio VR												
Posición Sensor	Aluminio T 237	Aluminio P	Alum %EE	Bronze T 150	Bronze P	Bronze %EE	Acero T 52	Acero P	Acero %EE	Cobre T 380	Cobre P	Cobre %EE
0	100.00	100	0.00%	100.00	100	0.00%	100.00	100	0.00%	100.00	100	0.00%
0.05	93.28	92.8	0.51%	92.43	91.8	0.68%	89.01	88	1.14%	93.88	93.9	0.02%
0.1	86.90	87.5	0.68%	85.39	85.6	0.24%	79.50	79	0.63%	87.99	88.3	0.36%
0.15	80.84	80.3	0.67%	78.84	78.8	0.06%	71.25	70.8	0.64%	82.29	82.3	0.02%
0.2	75.07	75.1	0.05%	72.73	72.8	0.10%	64.07	63.9	0.27%	76.77	76.8	0.04%
0.25	69.54	69.2	0.49%	66.99	66.6	0.58%	57.80	58.3	0.87%	71.41	71.1	0.44%
0.3	64.24	64.6	0.56%	61.59	61.3	0.47%	52.28	52.6	0.61%	66.20	66.1	0.16%
0.35	59.14	59.1	0.07%	56.48	56.1	0.67%	47.39	46.9	1.04%	61.13	61.4	0.45%
0.4	54.21	54.4	0.34%	51.62	51.4	0.43%	43.02	42.7	0.76%	56.16	56.1	0.11%
0.45	49.43	49.5	0.13%	46.98	47.2	0.46%	39.07	38.8	0.70%	51.30	50.9	0.77%
0.5	44.78	44.6	0.40%	42.51	42.5	0.04%	35.45	35.3	0.41%	46.51	46.9	0.83%
0.55	40.22	39.8	1.05%	38.19	37.9	0.76%	32.06	31.8	0.81%	41.80	42.2	0.96%
0.6	35.74	36	0.72%	33.97	34.1	0.39%	28.84	28.8	0.13%	37.14	36.9	0.64%
0.65	31.32	31.2	0.38%	29.82	29.4	1.40%	25.70	25.6	0.40%	32.51	32.1	1.27%
0.7	26.93	27.3	1.38%	25.71	25.5	0.81%	22.59	22.6	0.06%	27.91	28	0.31%
0.75	22.55	22.7	0.68%	21.60	21.3	1.40%	19.41	19.2	1.10%	23.32	22.9	1.81%
0.8	18.15	18.5	1.92%	17.47	17.5	0.17%	16.11	16.2	0.55%	18.73	18.8	0.39%
0.85	13.72	13.7	0.16%	13.28	13.5	1.67%	12.60	12.3	2.40%	14.11	13.8	2.21%
0.9	9.24	9.3	0.70%	8.99	8.8	2.14%	8.81	8.7	1.23%	9.46	9.1	3.83%
0.95	4.67	4.5	3.63%	4.58	4.2	8.25%	4.64	4.7	1.30%	4.76	4.8	0.78%
1	0.00			0.00			0.00			0.00		
			0.73%			1.04%			0.75%			0.77%

También se realizó una prueba de repetibilidad al momento de cambiar de material, se selecciono un valor aleatorio para la posición a analizar, 0.43 [m], y registro el valor de la temperatura para cada material, después, se modificó el orden de elección de los materiales de manera aleatoria para registrar los resultados. El objetivo de esta prueba es comprobar que el valor de temperatura permanece constante para cada material en una posición específica. Los resultados de esta prueba se observan en la *Tabla 2*.

Tabla 2. Resultados de la prueba de repetibilidad

Material	Posición [m]	Temperatura [°C]
Bronce	0.43	48.5
Aluminio	0.43	51.1
Cobre	0.43	52.8
Acero	0.43	40.9
Bronce	0.43	48.5
Cobre	0.43	52.8
Aluminio	0.43	51.1
Acero	0.43	40.9
Cobre	0.43	52.8
Acero	0.43	40.9
Aluminio	0.43	51.2
Bronce	0.43	48.4

Otra de las pruebas realizadas consistió en la comprobación de la sincronización del laboratorio virtual para dos usuarios, en esta prueba se utilizaron dos cuentas para comprobar que los cambios realizados por el usuario 1 se ven reflejados en la otra pantalla del usuario 2 y viceversa, es decir, cuando un usuario agrega o borra un punto el otro usuario es capaz de ver estos cambios desde su interfaz.

5.2 Análisis de resultados

Los cambios realizados al laboratorio virtual facilitan la lectura de los datos para el usuario, la gráfica agregada permite visualizar la relación de los resultados de manera más sencilla.

La temperatura en cada punto de la barra es menor en el Acero Inoxidable y mayor en el cobre, esto se encuentra relacionado al valor de la conductividad térmica de cada material. Se puede concluir que la conductividad térmica es directamente proporcional a la temperatura de la barra en cada punto.

Como se puede observar en la *Tabla 1* el porcentaje de error para cada material se encuentra por debajo del 1% en la mayoría de los casos, se puede observar que cuando el sensor de temperatura se encuentra más alejado de la resistencia el error suele ser mayor. Esto puede ser debido a que al ser cantidades más pequeñas la diferencia es más notoria.

Los errores obtenidos entre 3% y 8.25% son considerablemente mayores al promedio del resto de las mediciones, esta discrepancia se debe al redondeo de unidades que modifican el valor del resultado obtenido. El redondeo de las unidades de distancia solo se ve reflejado

en el pizarrón, pero el valor real es utilizado al momento de realizar los cálculos, esto provoca que los resultados visualizados en el pizarrón varíen en comparación al valor teórico esperado.

El error promedio obtenido para el aluminio fue de 0.73%, para el bronce de 1.04%, el acero inoxidable de 0.75% y el del cobre de 0.77%, debido a que los porcentajes de error son muy pequeños se considera que el laboratorio virtual es funcional y se puede utilizar como herramienta didáctica para la elaboración de prácticas.

En la *Tabla 2* se puede observar que es posible repetir los resultados esperados para la temperatura de un material en un punto específico, también se puede notar que para el último cambio de material en el Aluminio y el Bronce el valor de la temperatura registrada tuvo una ligera variación en comparación a los otros dos valores, esta pequeña variación se debe al redondeo de unidades que existe en el valor de la posición del sensor con la cuál se realizan los cálculos.

La prueba de sincronización realizada nos permite comprobar que es viable el uso del laboratorio para varios usuarios de manera simultánea. La sincronización funciona de manera adecuada y permite que varios usuarios interactúen con la misma interfaz.

6 Conclusiones

La realidad virtual puede ser utilizada como herramienta en el ámbito educativo, como se vio en la elaboración de este proyecto esta puede ser utilizada para realizar simulaciones que permitan a los estudiantes realizar prácticas sin la necesidad de estar físicamente en un laboratorio, lo cual lleva varias ventajas; permite que los usuarios se familiaricen con los instrumentos a utilizar lo que genera que exista una menor probabilidad de sufrir un accidente o dañar el equipo, al poder realizar la práctica de manera asíncrona se puede reducir el tiempo de elaboración de la práctica si la simulación es utilizada antes de entrar al laboratorio debido a que le permite a los estudiantes conocer la metodología a seguir, también ofrece la ventaja de reponer la práctica en caso de no poder acceder al laboratorio y de reforzar los conocimientos adquiridos durante su elaboración.

Para la elaboración de este proyecto fue muy importante el uso de librerías externas (*Udon Sharp* y *VUdon – Array Extensions*), estas librerías permiten implementar funciones más complejas a los *scripts* elaborados lo cual es de gran utilidad para obtener los resultados deseados en el laboratorio virtual. La programación por bloques es de gran utilidad al momento de programar funciones sencillas además es un método más amigable para usuarios que no se encuentran familiarizados con la programación, sin embargo, cuando se requiere de un *script* más complejo con diversos métodos, funciones y variables la interfaz gráfica de la programación por bloques se vuelve complicada de utilizar y de comprender para los demás programadores. El uso de *Udon Sharp*, que utiliza una sintaxis similar a *C#*, permite generar *scripts* más largos y complejos de una manera ordenada y que facilita la interpretación del código elaborado.

Debido a que en el entorno virtual no existen condiciones externas que puedan afectar los resultados del experimento los resultados obtenidos son algo diferentes a los que se obtendrían al realizar la práctica de manera física en el laboratorio. Las condiciones del entorno virtual permiten obtener valores muy cercanos a los esperados teóricamente lo cual es importante debido a que permite comprobar la funcionalidad de la simulación. Para aproximar los resultados del laboratorio virtual a la realidad es necesario tomar en cuenta algunos factores que alteran los resultados, como el redondeo de unidades, esto provoca que exista una ligera diferencia entre los resultados obtenidos y el valor teórico esperado. Otro aspecto que considerar es que para el experimento realizado se considera un estado estacionario, por lo cual los resultados no cambian durante la elaboración del experimento en el entorno virtual.

La elaboración de este proyecto requirió el uso de un entorno 3D lo que agrega cierta dificultad al momento de programar en comparación a un entorno 2D. En el entorno 3D la posición de los objetos se encuentra en tres dimensiones (x, y, z) lo que genera que los cálculos necesarios para modificar la posición de los objetos sean más complejos, los objetos en 3D tienen una geometría más compleja, además, es necesario cuidar aspectos como la perspectiva que tendrán los usuarios al momento de colocar los objetos en el entorno virtual para que la interacción con el entorno sea más inmersiva. Esto conlleva mayor trabajo para el procesador lo que puede conducir a problemas de rendimiento por lo que es necesario buscar maneras de optimizar el entorno virtual creado.

Otro aspecto por considerar es la importancia de la compatibilidad de el entorno virtual creado para ser utilizado con una computadora o con un equipo inmersivo, el software utilizado (*VRChat* y *Unity*) permite que el laboratorio virtual pueda ser importado de manera automática para utilizarse con cualquiera de estos equipos, pero fue necesario considerar las limitaciones técnicas del equipo inmersivo, al tener un procesador menos potente algunos elementos que funcionaban de manera adecuada en la computadora presentaban algunos problemas de rendimiento con el equipo inmersivo, fue necesario hacer pruebas en ambos equipos para corregir estos errores y optimizar el rendimiento del entorno virtual.

Durante el desarrollo del proyecto se aplicaron conocimientos adquiridos a lo largo de la carrera, principalmente aquellos enfocadas al área de programación. También fue necesario investigar y aplicar otros conocimientos técnicos como la base teórica del experimento a realizar para lograr comprender el resultado deseado, también fue necesario familiarizarse con las herramientas (*VRChat* y *Unity*) que permitieron la elaboración del laboratorio virtual.

Los objetivos planteados al inicio de este proyecto se cumplieron de manera satisfactoria, con los cambios realizados el laboratorio virtual es funcional para la elaboración del experimento ya que los resultados obtenidos son muy similares a los valores teóricos, además se agregaron elementos que facilitan la interpretación de los resultados como el nombre del material y la visualización de la gráfica distancia-temperatura. Finalmente, las interacciones agregadas a los objetos decorativos sirven para mejorar la experiencia del usuario.

7 Trabajo Futuro

El estado actual del laboratorio virtual es funcional y permite la elaboración de prácticas y la obtención de resultados muy cercanos a lo esperado teóricamente. Algunas de la propuestas para trabajar a futuro son:

- Agregar más interacciones con los objetos decorativos para mejorar la experiencia de inmersión. Por ejemplo, permitir a los usuarios interactuar con los libros los cuales pueden funcionar como acceso a una ventana emergente con un manual de uso del laboratorio virtual, simulando un manual de prácticas. El objetivo de agregar estas interacciones es mejorar la experiencia del usuario de manera que tenga un mayor interés al utilizar el laboratorio virtual.
- Solucionar el problema que existe por el redondeo de unidades en el valor de la posición, esto con el objetivo de reducir el error obtenido en el cálculo del valor de la temperatura y garantizar la repetibilidad del experimento.
- Agregar más mesas de trabajo para permitir que más usuarios puedan realizar la prueba al mismo tiempo. Esto permitiría utilizar el laboratorio virtual durante alguna clase, de manera que la mayoría de los alumnos puedan participar en la elaboración de la práctica. La interacción entre usuarios permite el intercambio de ideas y fomenta el trabajo en equipo. Sin embargo, es necesario considerar que una gran cantidad de usuarios podría generar problemas de rendimiento el entorno virtual.
- Analizar y solucionar los posibles problemas de rendimiento ocasionados por la presencia de varios usuarios en el entorno virtual, esto se puede solucionar optimizando las texturas y geometrías de los elementos gráficos presentes en el laboratorio virtual.
- Agregar una interacción que permita eliminar solo los puntos no deseados de la gráfica en lugar de eliminar todo, de esta manera los usuarios pueden eliminar puntos que se agregaron por error sin afectar el resto de los puntos agregados o se podrían eliminar los puntos de un material en específico.
- Realizar pruebas con usuarios para validar el adecuado funcionamiento y comprobar la utilidad del laboratorio virtual en sustitución o complemento a la práctica. Con esta prueba se buscaría verificar el rendimiento gráfico del entorno virtual cuando hay varios usuarios conectados.

8 Referencias

- [1] «Realidad Virtual, ¿qué es y qué aplicaciones tiene?,» 4 Agosto 2021. [En línea]. Disponible en: <https://www.edsrobotics.com/blog/realidad-virtual-que-es/>. [Último acceso: 21 Julio 2023].
- [2] E. Sandoval, «¿Qué es la realidad virtual?,» Escuela Británica de Artes Creativas y Tecnología, 22 Septiembre 2023. [En línea]. Disponible en: <https://ebac.mx/blog/realidad-virtual>.
- [3] Noticias El Comercio Perú, «Oculus Quest 2,» El Comercio, Septiembre 17 2020. [En línea]. Disponible en: <https://elcomercio.pe/tecnologia/tecnologia/oculus-quest-2-asi-son-los-nuevos-lentes-de-realidad-virtual-de-facebook-videojuegos-tecnologia-noticia/>. [Último acceso: 15 Marzo 2024].
- [4] S. Ferrario, «Historia de la Realidad Virtual,» Xperimenta Cultura, 3 Mayo 2016. [En línea]. Disponible en: <https://xperimentacultura.com/historia-de-la-realidad-virtual/>. [Último acceso: 20 Mayo 2024].
- [5] A. Cross, «The Evolution Of Virtual Reality: Exploring The Past, Present And Future,» Forbes, 9 Noviembre 2023. [En línea]. Disponible en: <https://www.forbes.com/sites/forbesbusinesscouncil/2023/11/09/the-evolution-of-virtual-reality-exploring-the-past-present-and-future/>. [Último acceso: 5 Junio 2024].
- [6] S. García-Bullé, «Realidad virtual y aumentada en educación,» 17 Julio 2022. [En línea]. Disponible en: <https://observatorio.tec.mx/edu-news/realidad-virtual-y-aumentada-en-educacion/>. [Último acceso: 15 Julio 2023].
- [7] R. Sousa Ferreira, R. A. Campanari Xavier y A. S. Rodrigues Ancioto, «La realidad virtual como herramienta para la educación básica y profesional,» *Revista Científica General José María Córdova*, vol. 19, pp. 223-241, 2021.
- [8] NC Tech, «La realidad virtual está transformando la educación y la capacitación,» 19 Junio 2023. [En línea]. Disponible en: <https://www.linkedin.com/pulse/la-realidad-virtual-est%C3%A1-transformando-educaci%C3%B3n-y-capacitaci%C3%B3n/>.

- [9] S. García-Bullé, «Aprendizaje y realidad virtual,» Instituto para el Futuro de la Educación, 18 Febrero 2021. [En línea]. Disponible en: <https://observatorio.tec.mx/edu-news/tecnologias-extendidas/>. [Último acceso: 15 Julio 2023].
- [10] Y. A. Cengel y A. J. Ghajar, Transferencia de calor y masa. Fundamentos y aplicaciones, México: McGraw Hill, 2011.
- [11] «Propiedades térmicas,» [En línea]. Disponible en: <https://www.infinitiaresearch.com/laboratorio-ingenieria-industrial/propiedades-termicas/>. [Último acceso: 15 Julio 2023].
- [12] F. Fernández-Rojas, C. J. Fernández-Rojas, K. J. Salas P., V. J. García y E. E. Marinero, «Conductividad térmica en metales, semiconductores, dieléctricos y materiales amorfos.,» Septiembre 2008. [En línea]. Disponible en: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0798-40652008000300001&lng=es&tlng=es.. [Último acceso: 12 Julio 2023].
- [13] G. Martínez, «¿Qué es Unity y para qué sirve?,» 13 Septiembre 2021. [En línea]. Disponible en: <https://ebac.mx/blog/que-es-unity-y-para-que-sirve>. [Último acceso: 4 Abril 2024].
- [14] D. García Erosa, «Qué es Unity y características principales,» OpenWebinars S.L., 10 Junio 2019. [En línea]. Disponible en: <https://openwebinars.net/blog/que-es-unity/>. [Último acceso: 5 Abril 2024].
- [15] Unity Technologies, «Unity user manual (2019.4 LTS),» [En línea]. Disponible en: <https://docs.unity3d.com/2019.4/Documentation/Manual/UnityManual.html>. [Último acceso: 6 Marzo 2024].
- [16] E. Rosicart, «¿Qué es VrChat?,» 12 Agosto 2022. [En línea]. Disponible en: <https://metaverse-news.es/que-es-vrchat/>. [Último acceso: 15 Marzo 2024].
- [17] E. Petrovich, «VRChat: 15 worlds every player should visit,» 16 Abril 2021. [En línea]. Disponible en: <https://gamerant.com/vrchat-best-worlds-to-visit/>. [Último acceso: 17 Marzo 2024].

- [18] VrChat Inc., «Getting Started,» [En línea]. Disponible en: <https://creators.vrchat.com/sdk/>. [Último acceso: 15 Marzo 2024].
- [19] VRChat Inc., «Creating your first world,» [En línea]. Disponible en: <https://creators.vrchat.com/worlds/creating-your-first-world>. [Último acceso: 15 Marzo 2024].
- [20] VrChat Inc., «Udon,» [En línea]. Disponible en: <https://creators.vrchat.com/worlds/udon/>. [Último acceso: 15 Marzo 2024].
- [21] Varneon, «VUdon-ArrayExtensions,» 2022. [En línea]. Disponible en: <https://github.com/Varneon/VUdon-ArrayExtensions>. [Último acceso: 2023].
- [22] Unity Technologies, «Line renderer component,» [En línea]. Disponible en: <https://docs.unity3d.com/Manual/class-LineRenderer.html>. [Último acceso: 10 Abril 2024].
- [23] Unity Technologies, «Prefabs,» [En línea]. Disponible en: <https://docs.unity3d.com/es/530/Manual/Prefabs.html>. [Último acceso: 10 Abril 2024].