

Revista Internacional Socio-Innova-Tec del Altiplano

Design and implementation of a cyber-physics laboratory with 3D immersion for teaching digital electronics

Diseño e implementación de laboratorio ciber-físico con inmersión 3D para la enseñanza de electrónica digital

Juan Camilo, ESPITIA¹, Arley, BEJARANO MARTÍNEZ¹, Andrés Felipe, CALVO SALCEDO^{1*}

¹*Univerisidad Tecnológica de Pereira. Carrera 27 #10-02 Barrio Álamos. Pereira Risaralda. Código postal: 66000. Colombia (0000-0001-9409-8982; 0000-0002-0470-537X; 0000-0002-5622-8798)*

Sent date: 03/09/2022 Acceptance date: 22/June/2023

Abstract:

In this paper we present a methodology for the design and validation of a digital electronics laboratory with 3D immersion. This system executes educational practices with real electronic devices, controlling and configuring these systems through a 3D virtual environment designed in Unity. Additionally, we present a case study implementing the methodology with different logic circuits. Our method shows the ease of design and creation of these systems, as well as the potential for the development of technology designed to operate in virtual environments, but with real physical behavior.

Keywords: Unity, Virtual Reality, Digital Systems, Programming, Serial Communication.

Resumen:

En este trabajo se presenta una metodología para el diseño y validación de un laboratorio de electrónica digital con inmersión 3D. Este sistema ejecuta prácticas educativas con dispositivos electrónicos reales, controlando y configurando estos sistemas a través de un entorno virtual 3D diseñado en Unity. Adicionalmente se presenta un caso de estudio donde se implementa y evalúa la metodología con diferentes circuitos lógicos. Los resultados muestran la facilidad para diseño y creación de estos sistemas, así como el potencial para el desarrollo de tecnología pensada para operar en entornos virtuales, pero con comportamiento real.

Palabras clave: Unity, Realidad Virtual, Sistemas digitales, Programación, Comunicación serial.

* Corresponding author. E-mail: afcalvo@utp.edu.co
Tel. 3148899237, Fax (606) 3137494

1. Introducción

Un laboratorio es un lugar equipado con los medios necesarios para realizar experimentos, investigaciones o trabajos de carácter científico o técnico. Esta infraestructura es esencial para la formación práctica de los estudiantes (IMCYC, 2018). En los últimos años, estos espacios no han sido ideales para el proceso de enseñanza, debido a factores como la sobrepoblación de estudiantes, la falta de presupuesto para la adquisición de equipos actualizados y recientemente la necesidad de funcionamiento a distancia por la pandemia por COVID-19. Esto afecta directamente el proceso, ya que no se pueden lograr competencias prácticas (Puentes *et al.*, 2020).

Aunque actualmente se pueden desarrollar prácticas de un laboratorio mediante herramientas de software, la interacción con el usuario es uno de los factores más relevantes a la hora de comparar un software de simulación con un laboratorio físico (Canchari Moreyra *et al.*, 2021). La experiencia del usuario al utilizar un software de simulación está muy alejada de la interacción que provee un laboratorio real, en cuanto a factores como ruido, incertidumbre, conexiones físicas, entre otras (Márquez *et al.*, 2010). Esto se debe a que en las herramientas de simulación se realizan montajes por medio de esquemáticos o bloques, esto no permite a los usuarios conocer realmente el sistema físico y las consideraciones de uso (Gómez-Prada, 2014). Existen distintos software para realizar simulaciones, estos tipos se pueden encontrar en formatos 2D y 3D (Bueno Lopez *et al.*, 2013). Las aproximaciones 2D poseen el problema mencionado anteriormente, la interacción con el usuario. Por otro lado, las interfaces en 3D mejoran en gran medida la interacción del usuario, acercándose más a la experiencia de un laboratorio real (G. Lorenzo *et al.*, 2000). No obstante, existe otro factor relevante a la hora de comparar los laboratorios virtuales con los reales y es que normalmente las simulaciones se basan en aproximaciones matemáticas, que no logran reproducir de forma exacta un fenómeno, eliminando factores fundamentales de un experimento real (P. Trentsios *et al.*, 2020). Las soluciones anteriores pueden ser un complemento para observar cómo debería comportarse un sistema antes de realizar la implementación física, sin embargo, se debe tener en cuenta que las simulaciones y los montajes físicos van de la mano, pero no se reemplazan entre sí.

Los sistemas ciber-físicos pueden dar solución al problema de los resultados obtenidos mediante simulaciones, debido a que obtienen los resultados mediante sistemas reales (Capraro *et al.*, 2010; Gracia Villegas, 2021; Ortiz *et al.*, 2020). De esta forma se pueden realizar prácticas de laboratorio de una forma mucho más acorde a la realidad en cuanto a resultados se refieren. Estos sistemas pueden tener dos tipos de funcionamiento: local o remoto. Los sistemas ciber-físicos locales, son controlados mediante una herramienta de software, sin embargo, los resultados del experimento no necesariamente se muestran por este medio, debido a que la persona que realiza el experimento, se encuentra en el lugar en el que está ubicado el módulo. De esta forma, el usuario puede observar físicamente el comportamiento del sistema. Al igual que los enfoques locales, los sistemas ciber-físicos remotos son controlados mediante una herramienta de software, sin embargo, este tipo de

sistemas se controlan a distancia mediante protocolos de comunicación por internet, por lo que, al no tener contacto físico con el módulo, el software que lo controla debe ser más robusto en cuanto a la información que maneja. Es decir, al no observar cómo se comporta el sistema, la interfaz debe poder controlar más variables del módulo y mostrar los resultados obtenidos durante la experimentación. Los sistemas ciber-físicos se componen de una interfaz gráfica, un módulo físico y los protocolos de comunicación entre ellos. Generalmente, el módulo físico se conecta a un sistema embebido (Raspberry PI, Arduino, entre otros) con el objetivo de controlar su funcionamiento, enviar información y adquirir datos. Para sistemas ciber-físicos locales, los sistemas embebidos se pueden comunicar con la interfaz gráfica mediante protocolos de comunicación serial como USB y RS-232. Para sistemas remotos, la comunicación generalmente se realiza mediante protocolos de red como HTTP y TCP IP. Por otro lado, la mayoría de las interfaces gráficas desarrolladas actualmente para este tipo de aplicaciones, se han construido utilizando Matlab, LabView o Java. Por esta razón, estas interfaces no presentan un nivel de inmersión adecuado, ya que estos lenguajes están pensados para realizar interfaces gráficas más tradicionales utilizando elementos simples, como botones, texto o gráficas (Khairudin *et al.*, 2019; Trentsios *et al.*, 2020; Pan *et al.*, 2006).

Existen diferentes laboratorios ciber-físicos que intentan solucionar el problema de la inmersión. Por ejemplo, el laboratorio remoto eLab3D creado en la Universidad Politécnica de Madrid, tiene una interfaz 3D que se comunica con el servidor mediante un protocolo HTTP y controla módulos del sistema mediante USB y GPIB (Ortiz, 2020). El problema principal de esta solución es la interacción con el usuario, ya que, al controlarse mediante el ratón y el teclado, se pierde la inmersión. La mayoría de estos sistemas ciber-físicos no se acercan a la experiencia real, debido a que sus interfaces son 2D. Por esta razón, se empezaron a implementar sistemas ciber-físicos en realidad virtual. Este tipo de laboratorios intentan mezclar la experiencia de estar en un sistema físico y la fiabilidad de los resultados de este. Sin embargo, muchos de los desarrollos actuales no poseen el nivel de inmersión esperado y son muy pocos los que proponen un ambiente 3D (Xiaopan *et al.*, 2020). Para desarrollar un laboratorio remoto en realidad virtual, es idóneo comenzar con un sistema ciber-físico local y luego este se puede expandir a su funcionalidad mediante estrategias de comunicación como MQTT, debido a las actuales ventajas que ofrece la proliferación electrónica. De esta forma, un sistema ciber-físico se puede transformar en un laboratorio remoto. Con base en el planteamiento anterior, se evidencia la necesidad de desarrollar un laboratorio ciber-físico con una interfaz de realidad virtual en el área de electrónica digital, que permita un funcionamiento con circuitos reales y una adecuada inmersión del estudiante. Es por esto que en este escrito se presenta el diseño y simulación de laboratorio de electrónica digital utilizando inmersión 3D por medio de la plataforma Unity. Los principales aportes de este trabajo son:

- Una estrategia para la creación de un ambiente virtual 3D en Unity, que permite una adecuada inmersión para conexión física de elementos electrónicos.

- Un algoritmo para generar la funcionalidad física de los elementos electrónicos existentes en el entorno virtual.
- Un algoritmo para integrar el entorno 3D y con un hardware físico que reproduzca el resultado y envíe los estados reales a Unity.

2. Materiales y Métodos

En la figura 1, se presenta la estructura para la implementación de un laboratorio ciber-físico con inmersión 3D para la enseñanza de electrónica digital. Esta metodología consta de cinco procesos. Inicialmente, se creó un entorno virtual para el manejo del sistema ciber-físico. Posteriormente, se desarrollaron las funcionalidades de los elementos 3D introducidos en el entorno de desarrollo. Además, se propuso un hardware que permite la implementación física de las funciones lógicas y el diseño del sistema electrónico que se conecta a la interfaz de realidad virtual. Por último, se realizó la validación de funcionalidad por parte de los usuarios.

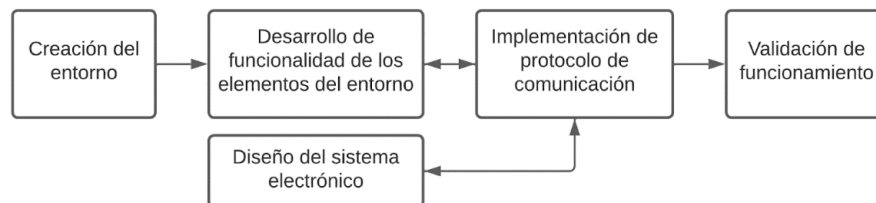


Figura 1. Metodología.

2.1. Creación del entorno virtual

La creación del entorno virtual se realizó mediante el entorno de desarrollo Unity, siguiendo la metodología mostrada en la figura 2. Como punto de partida, se realizó la búsqueda y creación de los elementos 3D necesarios para representar un laboratorio de electrónica digital (Salón, mesa de laboratorio, placa de pruebas, resistencias, compuertas lógicas, cables, LEDs y fuente de alimentación). La mayoría de los modelos se obtuvieron gratuitamente mediante páginas web dedicadas a la venta de modelos 3D. Los modelos que no se lograron adquirir, se crearon usando Shapr3D, una herramienta de modelado CAD 3D que permite realizar figuras tridimensionales mediante extrusión de planos 2D y operaciones booleanas entre elementos 3D (Unión, Diferencia, Intersección, entre otras). Después de tener todos los modelos 3D necesarios, se realizó la texturización de cada uno de los elementos haciendo uso de Blender, esto con el fin de que los componentes se asemejen a su apariencia física real.

Luego de tener preparados los elementos 3D, se realizó la integración de estos elementos en el entorno de desarrollo Unity. Para esto, se importaron al proyecto los modelos en formato FBX, luego, se ingresaron los modelos a utilizar en la escena del laboratorio, ubicando cada elemento en base a la disposición normal que se tendría en un laboratorio físico. Finalmente, usando el “XR Interaction Toolkit” de Unity se implementó la interacción en realidad virtual.

En la escena del laboratorio, se agregó un objeto llamado “XR Rig” y otro con el nombre de “XR interaction manager”. El “XR Rig” tiene como función controlar la cámara mediante las gafas de realidad virtual y las manos del usuario mediante los controles. Por otro lado, el “XR interaction manager”, tiene como función controlar el funcionamiento de los controles de las gafas de realidad virtual utilizadas. Por otro lado, a cada elemento del entorno que el usuario puede manipular se le agregó un script de este toolkit llamado “XR Grab Interactable”, este script permite agarrar los objetos mediante los controles de las gafas de realidad virtual.

Las compuertas lógicas tienen unas zonas específicas para colocarse, debido a que es necesario optimizar el espacio en la placa de pruebas para poder utilizar la mayor cantidad de compuertas posibles. Estas zonas de colocación se implementaron utilizando un script del toolbox de XR llamado “XR Socket Interactor”. Este script permite fijar un objeto en una zona específica. Cuando el usuario suelta el objeto en esta zona, este se queda fijo en una posición predeterminada, después, el usuario puede recoger el objeto simplemente agarrándolo de nuevo. Para interactuar con el entorno, se utilizaron las “Oculus Rift S”, sin embargo, el “XR Interaction Toolkit” de Unity soporta diferentes plataformas de realidad aumentada como ARCore, ARKit, HoloLens, Windows Mixed Reality, entre otras.

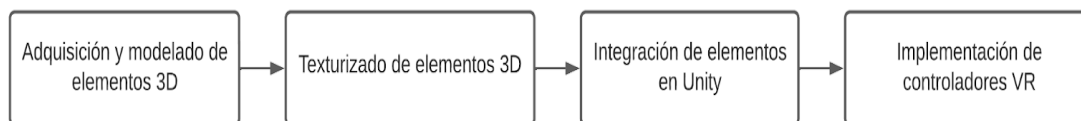


Figura 2. Metodología entorno virtual.

2.2. Funcionalidad de elementos virtuales

El desarrollo de la funcionalidad de los elementos virtuales se dividió en dos actividades. En primer lugar, se asignaron los disparadores necesarios para cada tipo de objeto. Un disparador o trigger, en el entorno de desarrollo de Unity, es una propiedad que se utiliza para determinar si un objeto interactúa con otro, y en base a esto realizar una o varias funciones específicas. Para las resistencias, cables, LEDs y la fuente de tensión se utilizaron dos disparadores, cada uno ubicado en cada pata del elemento o en el caso de la fuente en cada bornera. Para las compuertas lógicas y el dip-switch, se utilizó un disparador por cada pin del elemento. Por ejemplo, en el caso de las compuertas lógicas, al tener 14 pines, se les asignaron 14 activadores en total. Luego de tener asignados todos los disparadores de los elementos 3D, se crearon diferentes archivos de código o scripts para desarrollar las diferentes funcionalidades de cada componente del entorno virtual. En la figura 3, se presenta un diagrama en el que se puede observar la interacción que existe entre los scripts dentro de la escena del laboratorio virtual.

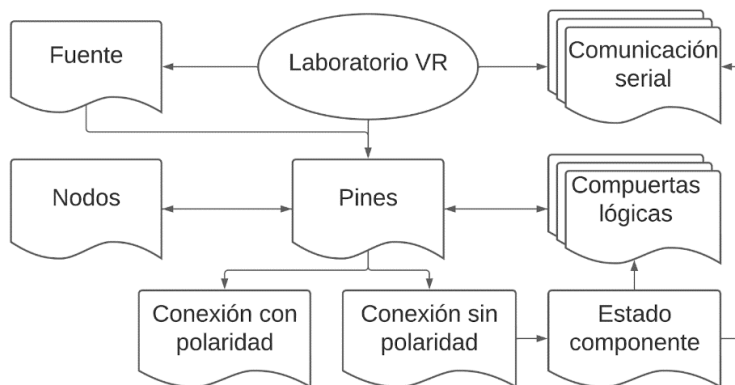


Figura 3. Relación entre los scripts del programa principal.

2.2.1 Estado Componente

La función de este script es almacenar el estado en el que se encuentra un elemento. Este archivo se encuentra asignado a todos los componentes del entorno: fuente, resistencias, LEDs, compuertas lógicas, nodos, entre otros. Existen cuatro estados diferentes: “VCC”, “GND”, “ON” y “-”, este último estado hace referencia a un estado de no conexión. En el caso de elementos pasivos (cables, resistencias y nodos) almacena “VCC”, “GND” o “-” dependiendo de a dónde está conectado el elemento. Por otro lado, para los elementos activos (LEDs y compuertas lógicas) este archivo almacena si el componente está encendido con el estado “ON” o si está apagado con el estado “-”.

2.2.2 Fuente

El objeto que representa la fuente de tensión, como se mencionó anteriormente, posee dos disparadores asignados por medio del archivo “Estado componente”. El script “Fuente” se encuentra asignado al objeto fuente, y tiene como función cambiar el estado de sus objetos hijos. Es decir, si la fuente está apagada, el estado de sus objetos hijos será “-”. Por otro lado, si la fuente está encendida uno de los hijos tomará el estado “VCC” y el otro el estado “GND”, para así empezar a distribuir estos estados a lo largo del circuito. Para encender y apagar la fuente se tiene un botón sobre ella, con este botón se cambia entre encendido y apagado.

2.2.3 Nodos

Mediante este script se puede determinar si el nodo toma el valor del componente, el componente toma el valor del nodo o si hay un corto circuito. Esto se hace comparando el estado del nodo con el del disparador del pin que entra en contacto con él. Para utilizar este script se crearon disparadores independientes y se utilizaron en el modelo 3D de la placa de pruebas simulando los nodos de la misma. Este archivo tiene como función tomar o asignar valores de “VCC” o “GND” para que otros elementos se puedan conectar y distribuir su

estado a otros nodos. Por ejemplo, si la pata de un elemento está en “VCC” o “GND” y el nodo no está conectado a nada (“-”), el nodo toma el valor de la pata del elemento. Por otro lado, si la pata de un elemento está en “VCC” y el nodo tiene un valor ya asignado de “GND” se puede determinar que se realizó un corto circuito.

2.2.4 Pines

Este código tiene como función almacenar el estado del disparador al que esté asignado. El archivo se encuentra en cada uno de los disparadores asignados a las patas de todos los elementos del entorno. Está directamente relacionado con el script “Nodos”, ya que los pines y los nodos se pueden cambiar el estado entre sí con el fin de distribuir los estados a lo largo de la placa de pruebas.

2.2.5 Conexión sin polaridad

Como su nombre lo indica, este script se encuentra asignado a elementos sin polaridad: resistencias y cables. Su función es determinar el estado del componente (“VCC”, “GND” o “-”) dependiendo de cómo están conectados sus pines. El estado de los pines lo obtiene mediante el script “Pines”. Por otro lado, este código también se encarga de intercambiar estados entre los pines, con el fin de distribuir los estados entre los nodos. También puede determinar si existe un cortocircuito.

2.2.6 Conexión con polaridad

Este script se encuentra asignado a elementos con polaridad: LEDs y compuertas lógicas. Al igual que el archivo “Conexión sin polaridad”, su función es determinar el estado del componente dependiendo de cómo se encuentran conectados sus pines. El estado de los pines lo obtiene mediante el script “Pines”. Una de las diferencias principales con respecto a la conexión sin polaridad es que los componentes pasan de tener como estado (“VCC” o “GND”) a (“ON” o “-”), esto se debe a que la conexión con polaridad tiene como objetivo encender o apagar un determinado elemento mientras que la conexión sin polaridad se utiliza para transmitir estados.

2.2.7 Compuertas lógicas

En esta sección se plantea el funcionamiento de las compuertas lógicas AND 74LS08, OR 74LS32 y NOT 74LS04. Esta tarea se realiza utilizando las tablas de verdad que tiene cada función lógica. Según el estado de cada pin de entrada, se realiza la operación lógica respectiva al tipo de compuerta y se asigna a su salida correspondiente.

2.3. Diseño del sistema electrónico y protocolo de comunicación

El objetivo de esta etapa es desarrollar un circuito electrónico para activar y desactivar cargas desde Unity utilizando comunicación serial. Esto con el fin de poder observar en el mundo físico los resultados de los circuitos lógicos implementados en el entorno.

En el laboratorio virtual se pueden utilizar hasta 4 LEDs para observar las salidas de los circuitos lógicos implementados. Cada uno de estos tiene una representación en el sistema físico, esto con el fin de controlar los LEDs del sistema físico mediante el entorno virtual.

El sistema electrónico se implementó mediante un Arduino Nano como se muestra en la figura 4. El código desarrollado, utiliza el puerto serial para comunicar el sistema embebido con Unity. Esto permitirá que se apaguen o prendan los LEDs físicos, así como los LEDs del entorno virtual.

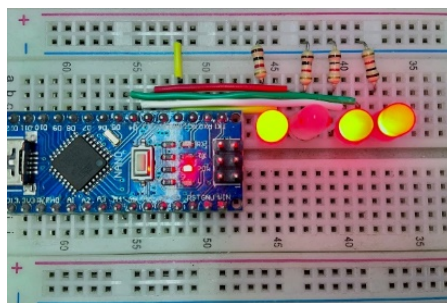


Figura 4. Sistema físico.

Para implementar la comunicación serial desde Unity se desarrollaron dos scripts: Administrador Serial y Datos Serial. El archivo “Administrador Serial”, utiliza la herramienta “System.IO.Ports” de Unity para la comunicación serial. Este permite crear una comunicación asíncrona a una velocidad de 9600 baudios.

Por otro lado, se utiliza programación multihilo utilizando la herramienta “System.Threading” para crear un proceso secundario que permita enviar los datos sin afectar el desempeño de la interfaz gráfica. Inicialmente, se ejecuta el programa principal, el cual crea el hilo de funcionamiento de todo el entorno, este hilo envía la información de los estados del circuito al hilo secundario, para que posteriormente se envíen al sistema embebido por medio de la comunicación serial. El hilo secundario se encarga de configurar y ejecutar la comunicación serial.

2.4. Validación de funcionamiento

Esta etapa está dividida en 6 pruebas, cuyo objetivo es realizar la validación del entorno, el funcionamiento de los elementos virtuales y la interacción con el sistema físico. Como punto de partida, se debe tener en cuenta que la fuente de alimentación está conectada a la placa de pruebas por defecto.

2.4.1 Prueba 1

El objetivo de esta prueba es determinar si los cables y las resistencias pueden distribuir estados entre diferentes nodos. Para esto se debe conectar un cable entre el nodo de VCC y un nodo cualquiera. Seguido a esto, se debe conectar una resistencia entre el nodo de GND y otro nodo cualquiera de la placa de pruebas. Luego de realizar las conexiones correspondientes, se debe encender la fuente de alimentación. Si el nodo al que se conectó VCC cambia su color a rojo y el nodo al que se conectó GND cambia su color a azul, se puede determinar el correcto funcionamiento de los cables y las resistencias al momento de transferir estados entre nodos.

2.4.2 Prueba 2

El objetivo de esta prueba es determinar si se realiza un posible cortocircuito. Para esto se debe utilizar como base la prueba 1, con la diferencia de que en esta prueba solo se utilizarán cables. Se debe conectar el nodo de VCC con el nodo de GND utilizando un cable. Esto con el fin de realizar un cortocircuito. Si el nodo que comparten los cables se pone de color gris, significa que se deshabilitó el estado del nodo y comprobaría el buen funcionamiento de la detección de cortos.

2.4.3 Prueba 3

Esta prueba valida el funcionamiento de un LED en el entorno virtual, verificando que el led se encienda cuando está bien conectado y que se mantenga apagado si no lo está. Para realizar esta prueba se debe conectar un LED con polaridad positiva hacia el ánodo. Si al encender la fuente se enciende el LED, su funcionamiento es correcto. Por otro lado, si se invierte la conexión el LED no debe encender.

2.4.4 Prueba 4

El objetivo de esta prueba, es validar la polarización de las compuertas lógicas. Para esto, se debe colocar una compuerta lógica en la placa y se debe polarizar el integrado, conectado VCC y GND en el número de pines que describe la hoja de datos del integrado. Si los pines de la compuerta cambian a color verde, se concluye que la prueba es exitosa. Al igual que en la prueba anterior, se debe invertir la polarización del elemento y verificar que los pines de la compuerta no se enciendan.

2.4.5 Prueba 5

En esta prueba se proponen dos funciones lógicas, en las que se verifica el estado de sus salidas, según sus valores de entrada. Si el circuito presenta el mismo comportamiento que la tabla de la verdad, se determina que el sistema logra reproducir el funcionamiento del

circuito lógico real. Las funciones lógicas son y su comportamiento se puede observar en la tabla 1. El símbolo * corresponde a la operación AND y el símbolo +, corresponde a la operación OR. Una línea sobre una variable u operación hace referencia a una negación lógica. Para realizar las pruebas de funcionamiento se plantearon las ecuaciones 1 y 2. Los resultados esperados de estas ecuaciones, se muestran en la tabla 1, en donde se observa el estado de cada LED dependiendo de las entradas del circuito.

$$\text{LED1} = \overline{(A * B) + C} \quad (1)$$

$$\text{LED2} = \overline{(A * B) + A + C} \quad (2)$$

Tabla 1. Tabla de verdad de la prueba 5

A	B	C	LED 1	LED 2
0	0	0	1	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

2.4.6 Prueba 6

Esta prueba tiene como objetivo verificar el sistema de comunicación serial, por lo que se envía el resultado de los estados de cada led y se verifica si estos encienden en el montaje físico. Por otro lado, se realizan 200 repeticiones de una trama de datos y verifica el número de veces que esta llega al sistema embebido.

3. Resultados y Discusión

En esta sección se presentan los resultados obtenidos al desarrollar el entorno virtual y al realizar las seis pruebas planteadas anteriormente, con el fin de validar el funcionamiento de todo el entorno. Luego de llevar a cabo la metodología descrita en la sección del entorno virtual, se obtuvo como resultado el entorno 3D que se muestra en la figura 5, en donde se observan todos los modelos utilizados. Adicionalmente en la figura 6 se presenta una demostración de la interacción entre el usuario y el entorno virtual.

Al llevar a cabo la prueba 1 descrita en la sección anterior, se pudo observar que el nodo conectado a GND por medio de la resistencia tiene un color azul y el nodo conectado a VCC por medio del cable tiene un color rojo, esto quiere decir que los cables y las resistencias transmiten correctamente los estados entre nodos. El resultado obtenido de esta prueba se muestra en la figura 7a.

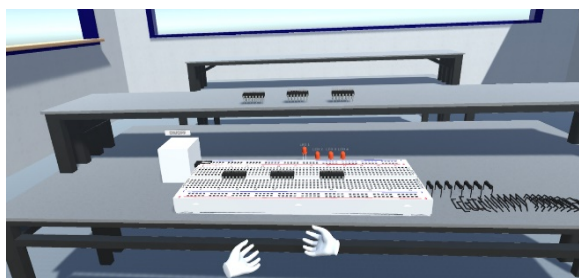


Figura 5. Entorno virtual.



Figura 6. Interacción con el usuario.

Como se mencionó anteriormente, la segunda prueba consistió en validar que los nodos detecten si hay un cortocircuito en la placa de pruebas. Como se puede observar en la figura 7b, se conectaron dos cables desde la línea de alimentación a dos nodos de la placa de pruebas, luego de esto, se conectó un cable entre los dos nodos alimentados. Al encender la fuente se observó que uno de los nodos cambió su color a gris. El resultado obtenido en esta prueba, valida el funcionamiento de la detección de cortocircuitos, ya que el nodo al cambiar de color a gris desactiva su estado y ningún otro elemento puede transmitir su estado.

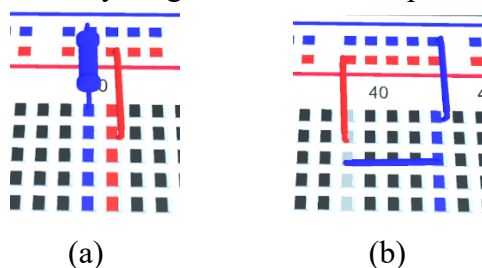


Figura 7. Pruebas de cables, resistencias y nodos. (a) Prueba 1. (b) Prueba 2.

Para la prueba 3 se tuvieron en cuenta dos casos: cuando el LED estaba bien conectado y cuando no. En la parte izquierda de la figura 8, se puede observar que el LED 1 tiene el cátodo conectado a GND y el ánodo conectado a VCC. Por otro lado, en la parte derecha de la figura 8, se observa que el LED 2 tiene el cátodo conectado a VCC y el ánodo conectado a GND. Al encender la fuente de alimentación el LED 1 se encendió y el LED 2 no. Esto significa que los resultados en esta prueba fueron los esperados. Por lo tanto, el funcionamiento de los LEDs en el entorno virtual es el correcto.

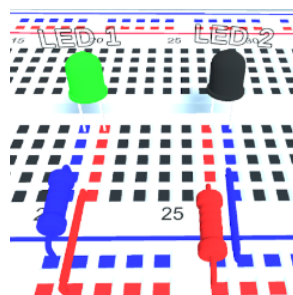


Figura 8. Prueba de LEDs.

En la prueba 4, el objetivo fue el mismo que en la prueba anterior, determinar que el elemento enciende o no dependiendo de su polarización. En este caso, se conectaron dos compuertas lógicas, una AND (izquierda) y una OR (derecha) como se muestra en la figura 9. La compuerta AND se polarizó correctamente y OR se conectó de forma inversa. Al energizar el circuito, se pudo observar que los pines de la compuerta AND cambiaron su color a verde, mientras que los de la OR no cambiaron su color. De esta forma se logró validar la polarización de las compuertas lógicas.

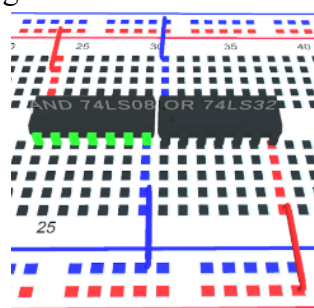


Figura 9. Polarización de compuertas lógicas.

La prueba 5 se puede considerar la más importante, ya que junta todo lo realizado en el desarrollo. En esta prueba se validó el funcionamiento general del sistema desarrollado, montando el circuito correspondiente a las ecuaciones 1 y 2. Para el montaje de este circuito se realizaron las conexiones necesarias para ejecutar las funciones lógicas. En las figuras 10 y 11 se pueden observar dos de los casos obtenidos al momento de realizar la comprobación de la tabla de verdad mostrada en la tabla 1. Como se puede observar en la figura 10a, el LED 1 se enciende y en la figura 10b se aprecia que el LED 1 del sistema físico también lo hace. La misma relación sucede en las figuras 11a y 11b. Al realizar todas las combinaciones posibles planteadas en la tabla de verdad, se pudo determinar que el circuito lógico montado en el entorno virtual simula perfectamente el comportamiento real de este tipo de circuitos.

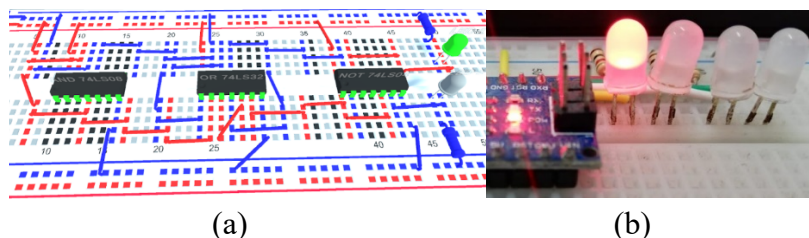


Figura 10. Caso: $A=1$, $B=0$, $C=0$. (a) Montaje en el entorno virtual. (b) Respuesta en el sistema físico.

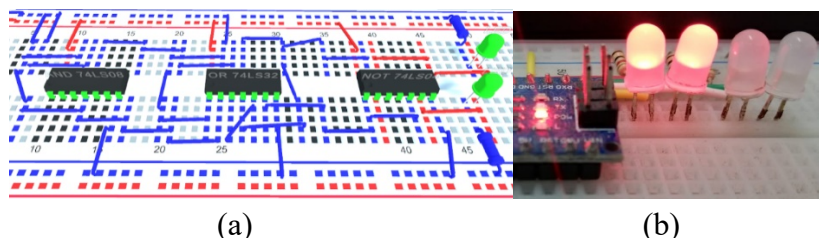


Figura 11. Caso: $A=0$, $B=0$, $C=0$. (a) Montaje en el entorno virtual. (b) Respuesta en el sistema físico.

Finalmente, en la prueba 6, se realizó la prueba de la comunicación serial, en donde se conectó el LED 1 en el entorno virtual como se hizo en la prueba 3. Luego se encendió y se apagó la fuente 200 veces, con el fin de determinar el porcentaje de acierto en la activación y desactivación de los LEDs mediante el puerto serial. Durante la prueba, se obtuvieron en total 400 datos: 200 encendiendo el LED y otros 200 apagándolo. Al finalizar se observó que el LED del sistema físico encendió 200 veces y apagó otras 200, lo que se traduce en un 100% de acierto en la comunicación serial, esto significa que no existe ningún tipo de error al momento de enviar información al sistema físico.

4. Conclusiones

Se desarrolló una metodología para el diseño e implementación de un laboratorio ciber-físico con inmersión 3D, que permite la enseñanza de electrónica digital. Para esto se utilizó el entorno de desarrollo Unity, un kit de realidad virtual Oculus Rift S y Arduino. Este trabajo permite realizar cualquier tipo de circuito de lógica combinatorial de una forma cercana a la realidad y ver representados los resultados en el mundo físico, además, en el entorno virtual es posible identificar un cortocircuito y el lugar en donde se generó, de esta forma se puede corregir y realizar una retroalimentación de la situación. Por otro lado, el desarrollo de este proyecto es un primer paso para la educación en el metaverso sin dejar de lado la interacción con el mundo físico. Con este desarrollo se abren nuevas posibilidades para los laboratorios educativos de nueva generación.

5. Referencias

- Bueno López, M., Arteaga Pérez, M., y Candea Leite, A. (2013). Modelado de sistemas de visión en 2D y 3D: un enfoque hacia el control de robots manipuladores. *Tecnura*, 12-21.
- Canchari Moreyra, L. Z., Leyva Lucas, C. L., Puchuri Meza, A. V., & Retamozo Mendoza, N. C. (2021). Desarrollamos la competencia explícita con laboratorios virtuales.
- Capraro, F., Tosetti, S., y Serman, F. V. (2010). Laboratorio virtual y remoto para simular, monitorizar y controlar un sistema de riego por goteo en olivos. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 73-84.
- G. Lorenzo, J. Suárez Rivero and M. García Domínguez (2000), "Desarrollo de un Laboratorio Virtual para el Estudio y Simulación de Mecanismos en la Ingeniería". https://www.researchgate.net/profile/Jose-Suarez-24/publication/228544369_Desarrollo_de_un_Laboratorio_Virtual_para_el_Estudio_y_Simulacion_de_Mecanismos_en_la_Ingenieria/links/540f6a4a0cf2df04e75a2fc1/Desarrollo-de-un-Laboratorio-Virtual-para-el-Estudio-y-Simulacion-de-Mecanismos-en-la-Ingenieria.pdf.
- Gracia Villegas, V. M. (2021). Laboratorio virtual remoto para el control de la planta multiprocesos.
- IMCYC (2018). La importancia de los laboratorios. *Revista Ingeniería Construcción y Tecnología*, <http://www.revistacyt.com.mx/index.php>
- K. N. Triana Ortiz, D. C. Herrera Muñoz, y W. N. Mesa Mendoza (2020), «Importancia de los laboratorios remotos y virtuales en la educación superior», wpecbti.
- Khairudin, M., Triatmaja, A. K., Istanto, W. J., & Azman, M. N. A. (2019). Mobile virtual reality to develop a virtual laboratorium for the subject of digital engineering.
- Márquez, J. A., y Sanguino, T. M. (2010). Diseño de laboratorios virtuales y/o remotos. Un caso práctico. *Revista Iberoamericana de Automática e Informática Industrial RIAI* 64-72.
- Ortiz, K. N. T., Muñoz, D. C. H., & Mendoza, W. N. M. (2020). Importancia de los laboratorios remotos y virtuales en la educación superior. *Documentos de Trabajo ECBTI*.
- Pan, Z., Cheok, A. D., Yang, H., Zhu, J., y Shi, J. (2006). Virtual reality and mixed reality for virtual learning environments. *Computers & graphics*, 20-28.
- Puentes, A. Y. P., Vargas, E. E., y Barrera, R. H. (2020). Laboratorios virtuales para la enseñanza de la química: una oportunidad para avanzar. *I Simposio Sul-Americano de Pesquisa em Ensino de Ciências*.
- U. Gómez-Prada (2014), "Learning Environment for Decision Making In Complex Dynamic Systems", CIINATIC. https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo10_per_12.pdf.

- Sebastià, J. M. (1987). ¿Qué se pretende en los laboratorios de física universitaria? Enseñanza de las ciencias: revista de investigación y experiencias didácticas, 196-204.
- Trentsios, P., Wolf, M., & Frerich, S. (2020). Remote lab meets virtual reality—enabling immersive access to high tech laboratories from afar. *Procedia Manufacturing*, 25-31.
- Xiaopan, Z., y Zhiyi, Y. (2020). Construction of Design System of Network Virtual Chemistry Laboratory based on Virtual Reality Technology. In 2020 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), 19-22.