

Convergencia de entornos virtuales 3D con tarjetas electrónicas como estrategia de apoyo a la educación virtual – Laboratorios virtualizados

Convergence in 3D virtual environments using electronic boards like as strategy to support of virtual education – virtualized laboratories

Víctor Cuchillac

Ingeniero en Electrónica por la Universidad Don Bosco.
Maestro en Informática Aplicada en Redes por la Universidad Francisco Gavidia
Investigador para el área de ingeniería y tecnología, ICTI-UFG
vcuchillac@ufg.edu.sv

RESUMEN

Este artículo es el resultado del diseño y pruebas de un modelo que permita la virtualización de laboratorios para las áreas de física o robótica como apoyo a las clases de carreras en línea. El modelo posee dos escenarios: el primero consiste en un ecosistema en donde todos los recursos están en línea, mientras que el segundo incorpora la integración de una tarjeta electrónica de bajo costo como interfaz entre la PC del alumno y las variables o sensores físicos.

Palabras clave: Entornos 3D, Arduino. Unity 3D, videojuegos, educación virtual.

ABSTRACT

This article is the result of design and testing of a model that allows virtualization of laboratories for physical or robotics areas to support careers online. The model has two settings, first consisting of an ecosystem where all resources are online, while the second includes integration of an inexpensive electronic card like as PC interface between the student and the physical variables or sensors.

Keywords: 3D environments, Arduino, Unity 3D, video games, virtual education.

Introducción

Contar con laboratorios virtualizados para una carrera en línea o presencial provee muchas ventajas, tanto en el proceso de enseñanza y aprendizaje, en la motivación de los estudiantes como en la gestión del equipamiento de laboratorios. Es por ello que el objetivo de este artículo es brindar un modelo que permita la integración de tecnologías líderes de bajo costo para la creación y el alojamiento de escenarios virtuales basados en tecnologías de videojuegos y tarjetas electrónicas para la vinculación con las variables físicas del entorno.

En este artículo se desarrolla un modelo propuesto para acceder y ejecutar los laboratorios virtualizados, el cual presenta dos escenarios: el primero, enfocado a trabajar desde cualquier navegador Web en una computadora personal; y el segundo escenario donde se muestra la vinculación de una tarjeta del tipo Arduino con la plataforma 3D de videojuegos; esto como resultado de las pruebas con las tecnologías expuestas más adelante.

Es del conocimiento general que cada vez las nuevas cohortes que ingresan a estudiar en las universidades poseen un perfil más intrínseco con las tecnologías de la información y las comunicaciones digitales; y qué decir de las competencias en videojuegos, en donde hasta con un teléfono de gama baja se pueden ejecutar juegos “simples” cuyo flujo y versatilidad son impresionantes. Es por ello que es imperante vincular en los procesos de formación contenidos, recursos y estrategias acordes a las características de las nuevas cohortes y los dispositivos que ellos utilizan de forma cotidiana.

La simulación y la emulación como estrategias para la formación no es algo nuevo, y para algunos que estudiaron asignaturas relacionadas con la electrónica, ya desde la década de los noventa existían tecnologías que utilizaban datos en consola o interfaces de baja resolución que les ayudaron significativamente en la descripción, cálculo y comprensión de los fenómenos eléctricos y magnéticos de los circuitos eléctricos. Y estas estrategias no solo beneficiaban a los estudiantes sino que también a las instituciones formadoras, porque se reducía la inversión de más “puestos de prácticas”, la cantidad de horas que el instructor debía dedicar a los estudiantes para revisar o exponer los fenómenos, la cantidad de elementos necesarios para las prácticas y el material dañado por las pruebas de los noveles practicantes.

1. Tendencia en la educación

Una galopante tendencia no solo en la región centroamericana sino que en todo el mundo es la formación de carreras universitarias en modalidad virtual; para la cual muchas instituciones educativas en la región ya cuentan con estos recursos: una infraestructura tecnológica, las plataformas para proveer la enseñanza virtual, algunos docentes con los perfiles requeridos, y los contenidos interactivos. Sin embargo, en carreras relacionadas con las ingenierías las competencias en donde se requiera del uso de laboratorios se complica la modalidad virtual. Complicar no significa que no se puedan generar los entornos que mejoren o garanticen las condiciones para el logro de las competencias, sino que esto supone un nuevo paradigma en el cual converjan las tecnologías y las tendencias que permitan la creación de dichos entornos.

Una inquietud que surge es ¿Son igual de efectivos los laboratorios virtuales que los laboratorios tradicionales?, a lo cual es posible encontrar mucha documentación al respecto; pero citando resultados muy recientes Marjorie Darrah y Roxann Humbert en su artículo *Are Virtual Labs as Effective as Hands-on Labs for Undergraduate Physics? A Comparative Study at Two Major Universities*, exponen que “No existe diferencia alguna entre los logros y, competencias desarrollados por los estudiantes que utilizaron laboratorios virtualizados, con respecto a los estudiantes que utilizaron los laboratorios tradicionales” (Darrah, Humbert, Finstein, Simon, & Hopkins, 2014); y al analizar el artículo se observa que los laboratorios por sí mismos no son la única herramienta o recurso educativo utilizado, es decir, los laboratorios serán otra actividad dentro de la asignatura así como serán otros recursos aparte de los textos, videos y autoevaluaciones.

1.1 ¿Por qué utilizar tecnologías tipo 3D?

Aunque existen muchos sitios con contenido orientado a la enseñanza de la física aplicada, su énfasis es más matemático y su representación es en 2D tal, como el gran trabajo del profesor Ángel Franco García, que aborda muchas áreas de la física, desarrollando laboratorios en JAVA (Franco García, 2010); pero estas tecnologías no presentan el potencial creativo y visual de las actuales, sin contar con las nuevas consideraciones de seguridad que presentan para ejecutar complementos (*plugin*) de terceros en los equipos de los usuarios y la complejidad para interactuar con otros dispositivos.

En concordancia con lo anterior se debe tomar en cuenta que para el desarrollo de esos entornos es necesario considerar varias soluciones informáticas como estas: las tecnologías de los videojuegos, donde el realismo cada vez es más innegable; el uso de tecnologías móviles, las cuales cada vez más incorporación mayores capacidades de procesamiento, conectividad, almacenamiento y visualización, siendo en muchas ocasiones la preferencia tecnológica del usuario en lugar de las computadoras de escritorio; el advenimiento de tecnologías de bajo costo para la interconexión de las computadoras, otros equipos y la realidad (sensores), como los innumerables prototipos de tarjetas basadas en microcontroladores, como por ejemplo la famosa tarjeta Arduino UNO.

1.2 Características del modelo propuesto

Con el objetivo de proponer un modelo que facilite el desarrollo de laboratorios virtualizados en la educación en línea se presentan dos escenarios que integran tecnologías de videojuegos y tarjetas electrónicas, basados en la interrogante ¿Es posible integrar de forma sencilla tecnologías de bajo costo que permitan la virtualización de laboratorios de física y robótica como apoyo a la educación en línea?

Para realizar la integración de las tecnologías que permitan construir y ejecutar los laboratorios virtualizados para contenidos de física o robótica ha sido necesario definir las características generales y funciones que el modelo o macrosistema debe realizar. Después de analizar la tendencia de algunas tecnologías se elaboró un grupo de características generales, las cuales se detalla a continuación:

- El uso de un *framework* de trabajo, que permita la aplicación de modelos físicos y matemáticos involucrados en los escenarios de los laboratorios.
- La capacidad de comunicarse con dispositivos electrónicos, para obtener una interacción con el usuario. Ya sea como parte de las capacidades intrínsecas del *framework* o extendida por el uso de complementos.
- La capacidad de conectarse y registrar en una base de datos los datos de la sesión del laboratorio (usuario, fecha, notas, tiempos, etc.). Las bases de datos consideradas fueron MS-SQL, My-SQL, *Postgres* y *Oracle*.
- La compatibilidad con formatos estándares (o los más utilizados en el modelado), elaboración de piezas y objetos en 3D. Con lo cual se podrán utilizar estos beneficios: recursos gratuitos y disponibles en la Internet, flexibilidad en el uso de las herramientas informáticas de diseño, ya sean gratuitas versiones de estudiantes y de pago.
- El uso de texturas, luces y realismo al estilo de las consolas de videojuegos. Así, la experiencia de los videojuegos hará sentir más confortable a los discentes y se podrá tener una vinculación con las tecnologías de uso lúdico y las tecnologías de enseñanza-aprendizaje.
- La tecnología que se utilice debe ser multiplataforma, ya que los estudiantes podrán utilizar equipos de PC (Windows, Linux y MAC OS X), equipos móviles (Android, IOS, Widows phone), entre otros.

2. Escenarios propuestos

En función de las características generales ya descritas se proponen dos escenarios:

• Escenario 1. Laboratorios en la nube.

La idea de este escenario es que el usuario pueda realizar los laboratorios utilizando un navegador Web desde una computadora ubicada tanto dentro como fuera del campus. Y se prevé que no habrá necesidad de interconectarse con un dispositivo, instrumento o equipo del laboratorio, debido a que todos los objetos e instrumentos utilizados en las guías del laboratorio estarán virtualizados.

• Escenario 2. Laboratorios ejecutados en computadoras del laboratorio.

Aquí se deberá instalar el software con las prácticas de laboratorio en las computadoras que se encuentran en los salones donde se realizan las prácticas de laboratorios, y a diferencia del modelo anterior existe la necesidad de conectar un instrumento, sensor o dispositivo electrónico a la computadora. Por supuesto que podría acoplarse objetos que no son electrónicos por medio del uso de una tarjeta electrónica utilizada como interfaz, que permita interactuar con la realidad y la computadora.

A continuación se detallan los aspectos generales de cada escenario, en donde se listan además las ventajas y desventajas.

Escenario 1: Laboratorios en la nube.

En este escenario la ejecución de los laboratorios virtualizados se realiza por medio de un

navegador Web, utilizando la plataforma virtual que posea la institución educativa como medio de acceso y repositorio de los recursos. En este escenario se prima la ubicuidad de los estudiantes para acceder a los laboratorios.

Ventajas:

- Los usuarios acceden a la información por medio de la autenticación de la plataforma virtual.
- Los laboratorios se actualizan más fácilmente, ya que solo se deben colocar en la plataforma virtual, sin necesidad de distribuirse en repositorios especializados (*Play Store de Android, Widows Phone Store o Apple Store*) o instalarse en los equipos Windows, Linux o Mac OS X.
- La capacidad de llevar estadísticas de acceso, lugar de la conexión, frecuencia de uso, laboratorios, etc. Estos registros son independientes de los que llevaría el sistema como ID usuario (carnet), puntaje y tiempo de ejecución de los laboratorios, entre otros.
- Fortalecer la plataforma virtual con contenido más interactivo.
- Proteger los laboratorios virtualizados del acceso no autorizado para utilizarse por otras instituciones no autorizadas, en el contexto que es un activo fijo y valioso de la institución.

Desventajas:

- Al no ser una tecnología compatible con HTML5 se requiere que los estudiantes instalen un complemento “*plug-in*” para ejecutar el archivo; sin embargo, es un *plugin*

muy pequeño que muchos antivirus y firewall reconocen como una aplicación válida, permitiendo que se instale y se ejecute en los navegadores de los equipos.

- Si se desea la configuración para integrar tarjetas electrónicas será más complejo el desarrollo de la aplicación y la latencia en las comunicaciones, ya que se deberá abrir más puertos.
- Las condiciones de conectividad del usuario pueden afectar el desarrollo de las actividades.

En la Figura 1 se ilustran los componentes involucrados en el Escenario 1 y en el Cuadro 1 se enumeran y se listan las tecnologías seleccionadas. (Ver Figura 1)

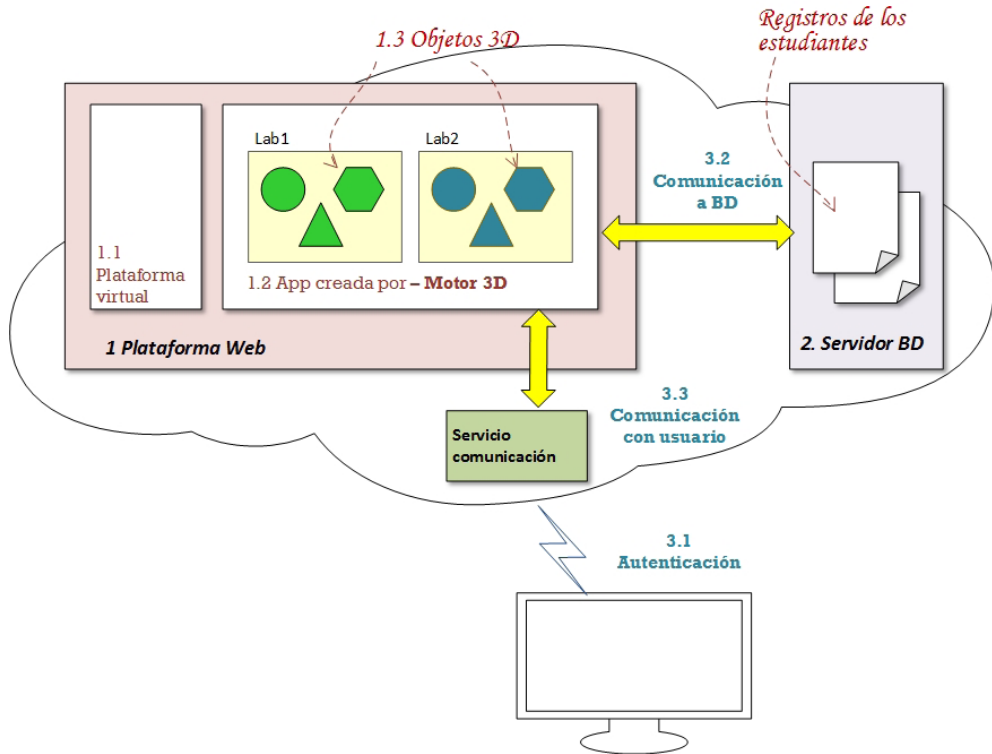
Para la comprobación del escenario 1, se necesitará de un navegador reciente (Firefox u Opera de preferencia), tener instalado el complemento gratuito de Unity 3D y acceso al sitio http://cuchillac.net/lab_virtuales/escenario1.html en donde se ha colocado un ejemplo para el escenario 1.

2.2 Escenario 2: Laboratorios ejecutados en equipos cliente.

En este escenario la ejecución de los laboratorios virtualizados se realiza desde las computadoras ubicadas en las instalaciones donde se realizan las prácticas de laboratorio, en las que podría o no estar conectada una tarjeta electrónica. Sin embargo, para este escenario se considera la existencia de una tarjeta electrónica que pueda medir variables exógenas a la computadora, vinculando con ello la realidad y el laboratorio virtual.

Figura 1

Escenario propuesto 1. Acceso en plataforma Web



Fuente: Elaboración propia.

Ventajas:

- Fortalecer la interacción del usuario y las variables a observar o controlar.
- Conectarse con mayor facilidad y transparencia con tarjetas de bajo costo, como las utilizadas en el proyecto Arduino, y a otros dispositivos que utilicen comunicación serial RS-232, RS-485, I2C, SPI, etc.
- Monitorear sensores, actuadores en tiempo real.
- Controlar actuadores y otros dispositivos inteligentes.
- Brindar la posibilidad de desarrollar una interfaz más realista a paneles SCADA en 2D.
- Virtualizar robots que puedan ser controlados por mando a distancia (Wi-Fi, IR, BT, RF, etc.).

- No hay costos de licenciamiento para el desarrollo o la compilación de proyectos para las siguientes plataformas Windows, Linux, y Mac OS X.

Desventajas:

- Si se desea compilar y distribuir los laboratorios en plataformas para dispositivos móviles como Android, iOS, Windows Phone, es necesario adquirir una licencia de uso comercial. Lo mismo ocurre si se desea compilar los laboratorios en Consolas de videojuegos.
- Si se realiza una modificación en laboratorio o se agregan más ejercicios, es necesario volver a compilar el proyecto e instalar el archivo ejecutable en los equipos que utilicen los estudiantes.
- Se debe crear un mecanismo de autenticación del usuario con la plataforma virtual o entidad de autenticación centralizada (Un servicio con LDAP o Kerberos por ejemplo).
- La comunicación en el software instalado en el cliente debe permitir el registro de los datos del laboratorio hacia la base de datos.

En la Figura 2 se ilustran los componentes involucrados en el Escenario 2, y en el Cuadro 1 se enumeran y puntualizan las tecnologías seleccionadas. (Ver Figura 2)

Para una comprobación del escenario propuesto se requerirá de una tarjeta Arduino UNO (o similar), descargar el programa de interconexión en la computadora donde está conectada la tarjeta de Arduino, tener instalado el complemento

gratuito “Unity 3D Player” y acceso al sitio http://cuchillac.net/lab_virtuales/escenario2.html

3. Breve descripción de las tecnologías a utilizarse

Para comprender los elementos que deben estar presentes en el modelo propuesto para ambos escenarios, se describen muy brevemente a continuación las tecnologías a utilizar y las razones por las cuales se ha escogido cada una de ellas. (Ver Cuadro 1).

3.1. Plataforma Web

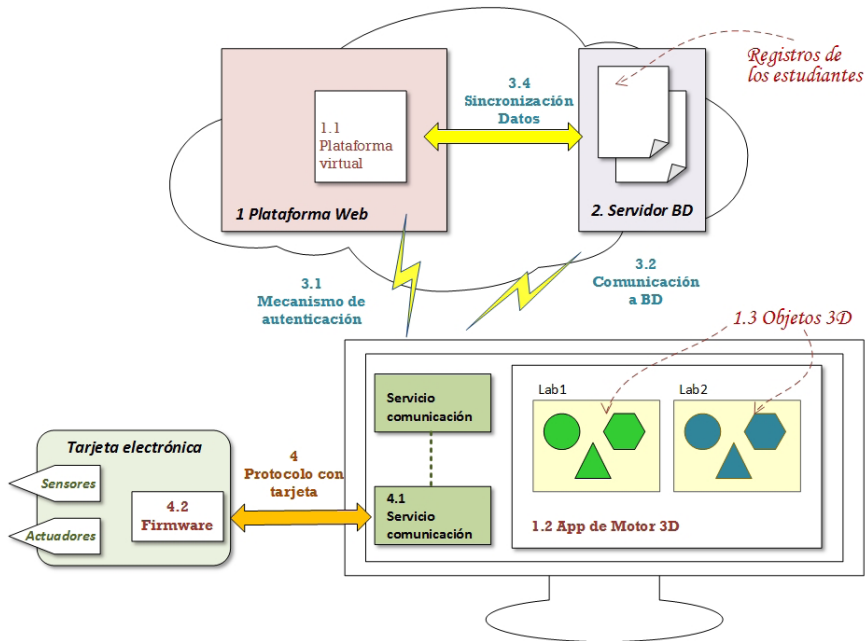
Se ha considerado el servidor web Apache debido a que es un servidor muy estable, muy documentado y multiplataforma. Además, es la tecnología utilizada para alojar a la mayoría de las plataformas de elearning que se utilizan en el país. Para las pruebas y la demostración se utilizó el appliance XAMPP corriendo en Windows.

3.1.1. Plataforma virtual

Es la tecnología que proporciona el acceso a los usuarios para acceder a los materiales de los cursos virtualizados; desde un enfoque técnico es el repositorio donde se encuentran los enlaces o “*frame*”, en donde se encontrarán los archivos de Unity 3D correspondientes a los laboratorios virtualizados. Para las pruebas o la demostración es indiferente si se utiliza Moodle, Sakai o un simple sitio web; ya que la función de esta tecnología o componente del modelo es el alojamiento HTML de los laboratorios. En las pruebas se utilizó una página web con autenticación basada en código para Web. Aunque es de considerar que al utilizarse una

Figura 2

Escenario propuesto n° 2. Laboratorios ejecutados en computadoras del laboratorio.



Fuente: Elaboración propia

Cuadro 1

Tecnologías seleccionadas en el modelo propuesto.

n.º	Elemento	Tecnología que se seleccionó
1	Plataforma Web	Servidor de alojamiento Web Apache (utilizado en las pruebas) o bien Servidor Web de IIS
1.1	Plataforma virtual	Moodle o Sakai
1.2	App del Motor 3D	Archivos binarios compilados por Unity 3D según tipo de escenario (Web, Windows, Linux, Mac OS X)
1.3	Objetos 3D	Objetos virtuales creados en: FreeCAD en formato OBJ, 3D max en formato 3DS
2	Servidor BD	Motor MySQL de Oracle (utilizado en las pruebas) o bien el fork MariaDB
3	Protocolo de comunicación	Se utilizará el lenguaje C#
3.1	Autenticación	Mecanismo de autenticación (difere en cada escenario)

n.º	Elemento	Tecnología que se seleccionó
3.2	Comunicación a la BD	Código y drivers de C# para MySQL
3.3	Comunicación con usuario	Programita en C# para recibir los datos del usuario
3.4	Sincronización de datos	Para elaborar las notas de los laboratorios virtualizados
4	Protocolo con tarjeta	Aplicación o programa para comunicar Arduino y PC
4.1	Servicio comunicación cliente	Programa para comunicación serial
4.2	Firmware de Arduino	Firmata en binario compilado para Arduino

Fuente: Elaboración propia.

plataforma como Moodle o Sakai se puede emplear una variable global para la autenticación hacia la base de datos que almacenará los resultados y datos de los estudiantes en los laboratorios.

3.1.2. Motor 3D – Unity 3D

Tal como se expone en el sitio oficial de la compañía, “Unity es una plataforma de desarrollo flexible y poderosa para crear juegos y experiencias interactivos 3D y 2D multiplataforma. Es un ecosistema completo para todo aquel que busque desarrollar un negocio a partir de la creación de contenido de alta gama y conectarse con sus jugadores y clientes más fieles y entusiastas” (Unity Technologies Esp., 2015).

Entendiendo por multiplataforma el hecho de que los productos creados en Unity pueden ser ejecutados en:

- Equipos de PC como Windows, Linux y Mac OS, (licenciamiento gratuito).
- Plataformas web (licenciamiento gratuito).
- Dispositivos móviles: Android, IOS de Apple,

Windows Phone 8, Tizen y BlackBerry (licenciamiento no gratuito).

- Consolas de videojuegos: PlayStation 3, 4 y Vita; Xbox One y Xbox 360; Wii U (licenciamiento no gratuito).
- Televisores inteligentes: Android TV, Samsung Smart TV (licenciamiento no gratuito).
- Realidad aumentada: Oculus Rift, Gear VR, Microsoft HoloLens (licenciamiento no gratuito).

Al brindar Unity 3D el desarrollo para sitios web y computadoras de forma gratuita, los desarrolladores de contenido de las universidades pueden producir no solo los escenarios de los laboratorios virtualizados, sino que también contenidos, aplicando los paradigmas de la “gamificación”, lo cual hará que los estudiantes puedan interactuar de una manera diferente a como se ha estado haciendo actualmente... donde las tareas más comunes son la descarga, la impresión y la lectura de documentos planos, ver vídeos o escuchar podcast o audios pregrabados.

Cuando se dice “ecosistema completo” y “flexible”, ello significa que la plataforma de desarrollo de

Unity 3D integra muchas herramientas para la producción de material 2D y 3D, y soporta varios formatos de objetos de multimedia. Entre las herramientas o características relacionadas a los laboratorios virtuales tenemos estos:

- **Soporte para algoritmos y modelados matemáticos y físicos**, ya que la empresa soporta NVIDIA® PhysX® 3.X (Unity Technologies, 2015), la cual es una tecnología que nos permite un comportamiento dentro del entorno similar a la realidad, en donde podemos tener fenómenos y efectos como fricción, choques, caídas, aceleraciones, rigidez, etc.
- **Flexibilidad en la programación.** Unity 3D cuenta con tres lenguajes de programación: *Boo*, el cual es un lenguaje propietario de Unity 3D con muy poca documentación en la Internet. *JavaScript*, que presenta facilidad para los programadores noveles y *C#*, el cual nos permitirá realizar tareas más complejas como establecer la comunicación desde Unity 3D hacia la base de datos MySQL, desde Unity 3D hacia un *webservice*, en donde se atenderán las peticiones y los procesos de los clientes, o bien poder crear los medios de vinculación con otras tecnologías.
- **Herramientas para la creación de animaciones.** Con las cuales los desarrolladores reducen el tiempo requerido para el desarrollo de dichas animaciones; son opciones básicas, pero permiten crear la animación sin utilizar otras herramientas de terceros.
- **Editor de terrenos.** En el cual se puede crear entornos de excelente calidad, también se

puede descargar objetos desde la tienda de Unity ya sea previo pago o los gratuitos.

- **Soporta una considerable cantidad de objetos, matrices, sonidos, etc.** Lo cual es muy conveniente, ya que se pueden desarrollar en otras tecnologías de modelado 3D los componentes de los laboratorios como esferas, probetas, pesas, engranajes, medidores, etc.

3.1.3. Objetos 3D

Bajo esta categoría se han considerado los objetos 3D que representarán a los instrumentos, bloques, poleas o cualquier objeto real, que se virtualizarían, y que tanto en la Figuras 1 y 2 se representa con las figuras geométricas. Debido a que el motor de Unity 3D acepta varios tipos de objetos, se ha considerado dos tecnologías acordes a ciertas condiciones que se presenten.

- **3D MAX**

Es una tecnología para el modelado o desarrollo, animación y renderización de objetos 3D; actualmente es parte de los productos que desarrolla y comercializa Autodesk (empresa que desarrolló Autocad) (Autodesk Inc, 2015). Aunque es un programa comercial la empresa ofrece una versión para estudiantes con ciertas limitaciones. 3D MAX es un programa muy completo y es por ello que presenta una curva de aprendizaje muy alta; sin embargo, existe muchísima información en Internet, tal como libros, vídeos tutoriales, foros, etc.

Un aspecto a considerar en el uso de 3D MAX es la cantidad de recursos computacionales que

se requieren, además de que algunas funciones son desarrolladas no tan efectivamente como lo hacen otros productos de la empresa Autodesk en términos de diseño de ingeniería. No obstante, 3D Max es un *software* muy utilizado en estudios de animación. También hay que considerar que la curva de desarrollo es alta (toma mucho tiempo dominar las competencias para ser un experto).

- **FreeCAD**

Es una aplicación informática para la creación de piezas en 3D de forma parametrizada, entre otras funciones, como estas: Ensamble, Renderización sencilla, Presentación de isométricos, Animación muy básica, etc. (FreeCAD Community, 2015). No es precisamente un software como 3D MAX, ya que es del tipo CAD (Diseño Asistido por Computadora) y parametrizado (capacidad de aplicar en los objetos, valores específicos como medidas, ángulos, colores, etc.); sin embargo, es posible crear objetos y exportarlos a Unity 3D o 3DS Max si se utiliza el formato OBJ.

FreeCAD es software libre, y la versión más reciente es la 0.16. Al ser software libre los desarrolladores y la comunidad están agregando (y mejorando) nuevas funciones, a manera de tener una herramienta muy completa para el diseño de piezas 3D.

Una ventaja en FreeCAD es la menor cantidad de recursos computacionales que se requieren para trabajar, comparado con 3D MAX o Inventor. También es de considerar que los objetos 3D creados en FreeCAD pueden ser

exportados a formatos para impresión 3D (impresoras que crean piezas de forma aditiva utilizando polímeros tipo ABS o PLA). La curva de desarrollo es mucho menor que 3D Max y se han realizado pruebas con jóvenes provenientes de Educación Media en la Escuela de Jóvenes Talento en TIC - UFG, en donde en menor tiempo que en 3D Max desarrollaron piezas y objetos 3D.

Una consideración muy importante es el nivel de realismo que deben presentar los objetos 3D en los laboratorios virtualizados, y para lograr esto es necesario en algunas ocasiones utilizar software que permita crear texturas, efectos de luz y otras características que incrementen el realismo, como el software ZBrush (software comercial) o Sculpttris (Software gratuito).

3.2. Base de datos

Tal como se comentó, es necesario que la información de los laboratorios sea almacenada en una base de datos (BD), ya que esto permitirá entre otras ventajas: llevar un registro de los puntajes o notas de los estudiantes y las estadísticas que permitan analizar el desarrollo de los laboratorios. Por ejemplo: número de aciertos o fallas, promedio del tiempo que tardan los estudiantes en finalizar los laboratorios, correlaciones sobre aspectos relacionados de los estudiantes (género, edad, tipo de institución, etc.) y los resultados de los laboratorios.

Aunque existen varias tecnologías para crear bases de datos se recomienda MySQL porque es una tecnología muy estable, ampliamente utilizada, muy documentada, multiplataforma, es software libre (aunque existe licenciamiento

comercial también) y es también la tecnología más utilizada con el servidor Apache y en plataformas de elearning como Moodle o Sakai. Para la implementación de los escenarios es necesario que los estudiantes realicen un proceso de validación, ya sea al servidor BD o en un sistema de autenticación externo.

3.3. Protocolo de comunicación con BD y Arduino

Debido a que Unity 3D posee integración con lenguajes como JavaScript y C#, es posible crear los programas que permitan establecer los mecanismos para abrir una sesión con el servidor MySQL y enviar o recibir datos desde la base de datos. Por mecanismo se puede considerar la ejecución de una página web con código PHP que se encarga del manejo y la integración del servidor Web Apache con la base de datos del Servidor MySQL.

De forma similar a lo anterior, se puede crear el programa en C# que defina el protocolo que realice la recepción y envío de datos desde la tarjeta electrónica hacia Unity 3D, utilizando un concepto similar a un “webservice”.

3.4. Protocolo de comunicación PC con tarjeta Arduino

En el escenario n.º 2 planteado, los usuarios se conectarán desde un equipo, tipo escritorio o portátil, que podría ejecutar un sistema operativo Windows, Linux y Mac OSX, en donde estará conectada la tarjeta de electrónica. La meta de este servicio es instanciar el protocolo que establezca la comunicación entre la tarjeta electrónica y Unity 3D. Para esta comunicación se utiliza el paradigma del cliente de webservice.

Técnicamente, este servicio estará encargado de establecer la comunicación vía red Ethernet o WIFI hacia el socket abierto por Unity 3D con los datos que lleguen al puerto serial enviados por la tarjeta electrónica.

Para instanciar este servicio existen varias opciones: una es utilizar la plataforma de desarrollo Mono la cual es un Framework de desarrollo para .NET bajo el esquema de software libre, en donde se puede crear un programa en el lenguaje C#, y considerando que ya se ha utilizado C# para la comunicación de la Base de datos; utilizar dicho lenguaje permite una reutilización de tecnologías. Otra opción para realizar la comunicación es utilizar el programa Processing, el cual tiene soporte para la comunicación directa con tarjetas del tipo Arduino y la comunicación por socket.

La opción seleccionada en el escenario N.º 2 para el programa de comunicación es Processing, ya que posee muchas opciones requeridas para las comunicaciones seriales o por red, y estas son implementadas de una manera muy fácil. No se requiere pagar licencias, y el programa puede crear archivos compilados en JAVA para las plataformas Windows, Linux o Mac OS X, los cuales son sistemas operativos considerados, que utilizarían los usuarios o estudiantes que desarrollen los laboratorios.

Processing es un entorno de programación bajo el esquema de software libre multiplataforma, que utiliza una estructura similar y compatibilidad con JAVA, el cual fue diseñado inicialmente para la enseñanza de programación, producción de proyectos de multimedia e interacción con

usuarios y dispositivos electrónicos (Processing community, 2015).

Utilizar *Processing* en este proyecto permite controlar la comunicación serial, monitorear el estado de la tarjeta electrónica, enviar y recibir datos hacia dicha tarjeta electrónica, convertir los datos cuando es necesario, enviar los datos recibidos hacia el programa de C# que ejecuta Unity 3D. Además de todo lo anterior, *Processing* permite crear una interfaz gráfica sencilla que mejora la visualización y la interacción con el usuario. Sin embargo, es de mencionar que en las pruebas realizadas para el proceso de compilación se presentaron errores al incrustar los archivos binarios de Java; esto, porque al compilar se escoge una versión de la JRE (*Java Runtime Environment*) que podría ser o no compatible con la versión que tenga el cliente si utiliza Windows 8.1. Las pruebas realizadas en equipos Windows 7 y Linux Ubuntu 12.04 LTS no presentaron dificultades.

3.4.1. Protocolo con la tarjeta

Al utilizar la lectura directa del puerto serial (en donde estará conectada la tarjeta electrónica), la lógica de programación se reduce, ya que es el sistema operativo el que carga los drivers de la tarjeta y una vez reconocido el dispositivo, solo se necesita administrar la lectura de cada dato recibido. Además, si la tarjeta electrónica no estuviera conectada por un puerto serial tipo USB, sino que se utiliza un dispositivo Bluetooth, es el sistema operativo el que establecería la comunicación por medio de los drivers del adaptador Bluetooth, instanciando así los puertos para la comunicación con la tarjeta electrónica. Para la demostración y las pruebas

se utilizó una tarjeta electrónica conectada serialmente. Obviamente, el programa que se utilice deberá tener las siguientes funciones:

- Reconocimiento de la tarjeta electrónica Arduino cuando se conecta por un puerto USB.
- Selección del puerto serial si se utilizaría un dispositivo Bluetooth.
- Manejo de la comunicación si se utiliza un dispositivo Bluetooth.

La tarjeta electrónica que se utilizó para envío de los datos provenientes de los sensores y entradas digitales hacia Unity 3D es una tarjeta Arduino UNO; la selección de esta tarjeta fue debido a los siguientes puntos: Amplia documentación de uso, bajo coste de adquisición, uso de múltiples herramientas gratuitas para la programación, diseño *plug and play* para la integración de sensores, entradas digitales, actuadores, entre otros.

3.4.2. Firmware de Arduino

Para establecer la comunicación entre los pines de la tarjeta electrónica Arduino UNO y la computadora del usuario, se seleccionó el protocolo Firmata, ya que es un protocolo muy versátil, sencillo de entender y con soporte para varias tecnologías como Processing, Python, Perl, Ruby, Java, PHP, .NET, iOS, entre otros.

Firmata es un protocolo desarrollado para la comunicación entre microcontroladores (Atmega de Atmel, PIC de Microchip) y un programa que se ejecute en una computadora, tableta electrónica o teléfono inteligente (Firmata

project, 2015). El protocolo utilizado está basado en mensajes seriales de texto tipo MIDI, en donde se utilizan 8 bits para instrucciones y 7 bits para los datos.

4. Metodología sugerida para implementar laboratorios virtualizados

Para el escenario N°. 1 se utilizan la Fase 1 y la Fase 2, mientras que para el escenario N°. 2 se debe agregar la Fase 3.

4.1. Fase 1: Modelar objetos 3D

Por modelar se comprenderá el proceso de dibujar y de construir los objetos 3D que se utilizarán en los laboratorios; por ejemplo, las esferas y canales que se podrán utilizar en un laboratorio de física. Aquí es muy conveniente trabajar con las escalas reales de los objetos a virtualizarse. Las tecnologías que se emplearon y que se recomiendan en este documento son estas:

- **FreeCAD**, modelar los objetos y exportarlos en formato obj para abrirse en Unity 3D; es muy conveniente utilizar la última versión, para las pruebas se utilizó la versión 0.15.
- **3D max**, el formato utilizado es .3DS, y se empleó la versión 2015; la curva es más lenta pero presenta más opciones que FreeCAD para el *skinning* y el *rigging*. La versión de estudiante es gratis y para universidades existe una licencia especial.

Además de las tecnologías anteriores es posible que se utilicen otras herramientas para crear animaciones, estructuras móviles (*rigging*),

texturización (*skinning*). Esto, con el fin de que el acercamiento a la realidad sea lo mayor posible.

4.2. Fase 2: Programar el escenario

Esta fase se refiere a cómo interactuarán los objetos para conformar el laboratorio; en forma general se recomienda C# porque existe amplia documentación y es posible conectarse a otras tecnologías como bases de datos, *webservices* y otro tipo de aplicaciones de escritorio. Es muy importante documentar y realizar los comentarios que permitan comprender la lógica del código, ya que los laboratorios pueden ser actualizados más adelante por personal que no estuvo al inicio del desarrollo. Como Unity 3D permite utilizar JavaScript es posible utilizar una gran cantidad de códigos que se encuentran en Internet de forma gratuita, y combinarlos con C# si no se tiene personal con competencias en este último lenguaje.

Es importante que el código represente el modelo matemático o físico de la realidad, y la animación, como el modelado 3D, apoye el objetivo del laboratorio en términos visuales y motivacionales para los discentes. Por ejemplo, en un laboratorio de física, si un canal posee un coeficiente de fricción 1 y otro canal posee un coeficiente de fricción 2, el efecto de una esfera deslizándose por ellos debe representar el patrón acorde a cada coeficiente. Lo mismo aplica si se tiene un laboratorio relacionado con la caída libre y los rebotes sucesivos de esferas de diferentes materiales, en donde una esfera simula ser de un material con un cociente restitución de un material 1 (acero) y una segunda esfera simula ser de un cociente de restitución 2 (goma). Además, aunque esto genere más trabajo, será

extremadamente ilustrativo hacer una animación en donde se vea cómo la segunda esfera de goma se aplasta en el primer impacto, por lo que pierde más energía que la esfera de acero la cual no se aplasta. En la vida real debería existir en el laboratorio una cámara de vídeo de alta velocidad para mostrar la deformación antes mencionada; pero obviamente esto incrementa los costos de equipamientos de los laboratorios de física.

De lo ya expuesto es necesario tener claro que los laboratorios deben tener como objetivo modelar y representar los comportamientos matemáticos, físicos, químicos, eléctricos, entre otros de los objetos y que la animación, la textura y la iluminación son complementos necesarios para darle el mayor realismo posible, para que los estudiantes, visual y cognitivamente desarrollen un aprendizaje significativo.

4.3. Fase 3: Integración con *hardware*

Esta fase se debe realizar para el segundo escenario y tiene que ver con la adecuación de las tarjetas que comunican a la aplicación de Unity 3D con la realidad. En las pruebas se utilizó tarjetas Arduino UNO para la comunicación entre sensores y una aplicación en Unity 3D. Es muy favorable utilizar el protocolo firmata, el cual es libre, muy documentado y fácil de implementado en muchas tecnologías y sistemas operativos.

Para las pruebas de comunicación entre Arduino y los sensores, se recomienda utilizar la herramienta gráfica Platex, la cual puede probar el valor de los sensores, estado de las entradas y salidas digitales. Aunque ya no se ha continuado desarrollando una versión más actualizada, la última versión se encuentra disponible en [https://](https://github.com/chiva/Platex)

github.com/chiva/Platex, también se puede utilizar la aplicación firmate test disponible en http://firmata.org/wiki/Main_Page

Una vez garantizada la comunicación de las variables con *Firmata*, se deben realizar las pruebas de comunicación entre la aplicación de *Processing* (podría crearse o utilizarse otra tecnología) y el socket creado en C# para Unity 3D, una herramienta utilizada para dicha comunicación es *RealTerm*, el cual es una herramienta para monitorear, capturar, enviar y recibir streaming o flujos de datos de tipo serial y TCP. LA herramienta es gratuita y está disponible en <http://realterm.sourceforge.net/>

5. Resultados

Las pruebas con los dos escenarios realizados demostraron que es posible la integración de Unity 3D con Bases de datos, plataformas *on-line* y tarjetas Arduino, entre otras. Dado que existe mucha información disponible de cada tecnología en Internet, libros y foros, desarrollar este tipo de recursos educativos presentaría ventajas, tanto cognitivas para los estudiantes como financieras para las instituciones de formación, así como un mejor desarrollo de las competencias en asignaturas como física, química o robótica. Los costos se reorientarían de la adquisición y el mantenimiento del equipamiento del laboratorio actual al desarrollo de objetos en 3D de alta calidad visual por un equipo de profesionales.

Para la publicación (producción) de los laboratorios en las plataformas Windows, Linux y Mac OS X se puede utilizar de forma gratuita Unity 3D, lo mismo para implementaciones en plataforma Web, mientras que, para la publicación en Plataformas Android, IOS, Windows Phone

o las consolas de videojuego se requiere en Unity 3D del pago de una licencia. Si el laboratorio no tendrá interacción con tarjetas electrónicas, se recomienda construir la aplicación tipo Web; mientras que, si hay una comunicación con tarjetas electrónicas, computadoras conectadas a equipos, la mejor opción es crear una aplicación que utilice en un extremo la comunicación serial con el protocolo *Firmata* y crear en el otro extremo un socket IPv4 para el envío de los datos hacia la aplicación en Unity.

En las pruebas realizadas los retrasos de tiempo fueron aceptables, no superando los 600 ms; sin embargo, es muy eficaz enviar una matriz de datos con las variables, que estar interrogando cada pin en las tarjetas electrónicas y enviar su valor. Aunque en las pruebas para el envío de datos a la base de datos de MySQL no se utilizaron mecanismos para la encriptación del canal o de los datos, la aplicación de estos mecanismos es un aspecto muy que debe ser incluido, así como, utilizar una base de datos que reciba la información y que sea consultada por un *script* para extraer los resultados e incluirlos a la base

de datos del registro académico. Ya que, al hacer esto se evita exponer un acceso de una aplicación a la base de datos del registro académico.

La realidad mostrada en los laboratorios dependerá en gran manera de aspectos como el detalle y el modelado de los objetos, uso de escalas reales, calidad en el proceso de renderizado, iluminación y efectos de sonido; por lo cual la integración de personal con las competencias para la animación digital y los videojuegos es necesaria. Es así como, un buen equipo de desarrollo de actividades virtualizadas deberá estar formado por personas con los siguientes roles: experto de la cátedra, quien deberá probar varias veces los escenarios y realizar acciones absurdas para validar la virtualización de la realidad; un diseñador o animador gráfico, quien desarrollará los objetos en 3D; un programador en Unity 3D para desarrollar los escenarios, así como las conexiones a las bases de datos o el protocolos de comunicación y un experto en electrónica para incorporar las tarjetas electrónicas, sensores o consolas desarrolladas con el protocolo de comunicación.

Anexos – Configuraciones y protocolos

Aquí se detallan los códigos más relevantes de la integración, y gracias a los aportes que en su momento compartió Remzi Yagiz Mungan disponibles anteriormente en (https://www.cla.purdue.edu/vpa/etb/resources/tech_workshops.html)

```
// Information about (myu) Max-Unity Interoperability Toolkit
// Developed by
// Ivica Ico Bukvic <ico@vt.edu> <http://ico.bukvic.net>
// Ji-Sun Kim <hideaway@vt.edu>
// Keith Wooldridge <kawoold@vt.edu>
// With thanks to Denis Gracanin
// Virginia Tech Department of Music
```



```
// DISIS Interactive Sound & Intermedia Studio
// Collaborative for Creative Technologies in the Arts and Design

// Copyright DISIS 2008.
// mu is distributed under the GPL license v3 (http://www.gnu.org/licenses/gpl.html)
using UnityEngine;
using System.Collections;
using System;
using System.Text;
using System.Net;
using System.Net.Sockets;
using System.IO;

public class TCPHost : MonoBehaviour
{
    //Para abrir el puerto
    public int miPuerto = 3233;
    private TcpClient clienteTCP;
    private NetworkStream flujodatosDelCliente;
    private TcpListener miServicio;
    private bool varEsperar;

    // Para la comunicación con la tarjeta
    public GameObject targetObjcet;
    public bool LED13 = false;
    public bool IsLED13Changed = false;

    // Inicializar el servicio
    void Start ()
    {
        varEsperar = false;
        miServicio = new TcpListener(IPAddress.Any, miPuerto);
        miServicio.Start();
    }

    // Para actualizar los frames
    void Update ()
    {
```

```

string s;
string[] values;
if (miServicio.Pending()) {
    clienteTCP = miServicio.AcceptTcpClient();
    flujodatosDelCliente = clienteTCP.GetStream();

    varEsperar = true;
}
while (varEsperar && flujodatosDelCliente.DataAvailable) {
    try {
        int numread = 0;
        byte[] tmpbuf = new byte[1024];
        numread = flujodatosDelCliente.Read(tmpbuf, 0, tmpbuf.
Length);

        s = Encoding.ASCII.GetString(tmpbuf, 0, numread);
        s = s.Replace("\n", "");
        values = s.Split(';');
        //print ("values"+values.ToString()); //for debugging

        if (values.Length > 1) {
            for (int i = 0; i < (values.Length-1); i++) {
                Parse(values[i]);
            }
        }
        else Parse(values[0]);

        if(LED13 != IsLED13Changed)
        {
            //enviar espacio como fin de datos
            string TempSend = LED13.ToString();
            TempSend += " ";
            //Debug.Log(TempSend);

            Byte[] data = System.Text.Encoding.ASCII.
GetBytes(TempSend);

            flujodatosDelCliente.Write(data, 0, data.Length);

```

```

        IsLED13Changed = LED13;
    }
}
//Called when flujodatosDelCliente fails to read from the stream.
catch (IOException e) {
    varEsperar = false;
    flujodatosDelCliente.Close();
    clienteTCP.Close();
}
//Called when flujodatosDelCliente has been closed already.
catch (ObjectDisposedException e) {
    varEsperar = false;
    clienteTCP.Close();
}
}
}

void Parse(string toParse)
{
    string[] values = toParse.Split(' ');
    //Acceder al archivo anexo
    targetObjcet.GetComponent<Rotator>().RotationAngle = float.Parse(values[0]);
}
}
}

```

Referencias bibliográficas

- Autodesk Inc. (2015). Autodesk. Obtenido de Descripción de 3DS Max: <http://www.autodesk.es/products/3ds-max/overview>
- Darrah, M., Humbert, R., Finstein, J., Simon, M., & Hopkins, J. (2014). Are Virtual Labs as Effective as Hands-on Labs for Undergraduate Physics? A Comparative Study at Two Major Universities. *Journal of Science Education & Technology*, p. 803-8014. doi:10.1007/s10956-014-9513-9
- Firmata project. (2015). Documentación oficial del protocolo Firmata. Obtenido de <https://github.com/firmata/protocol>
- Franco García, A. (Diciembre de 2010). Física con ordenador. Obtenido de Curso Interactivo de Física en Internet: <http://www.sc.ehu.es/sbweb/fisica/default.htm>
- FreeCAD Community. (2015). FreeCAD. Obtenido de An Open Source parametric 3D CAD modeler: <http://www.freecadweb.org/>

Processing community. (2015). Sitio principal del proyecto Processing. Obtenido de Descripción de Processing: <https://processing.org/>

Unity Technologies Esp. (2015). Sitio principal de Unity 3D en Español. Obtenido de ¿Qué es Unity 3D?: <https://unity3d.com/es/unity>

Unity Technologies. (julio de 2015). Manual de uso de Unity 3D. Obtenido de <http://docs.unity3d.com/Manual/UnityOverview.html>