

Propuesta de implementación de plantas virtuales para la enseñanza de programas de control lógico

Rubén Darío Vásquez-Salazar, Henry Omar Sarmiento-Maldonado & Deinis Sirley Muñoz-Muñoz

*Facultad de Ingeniería, Politécnico Colombiano Jaime Isaza Cadavid, Medellín, Colombia.
rdvasquez@elpoli.edu.co, hosarmiento@elpoli.edu.co, deinismm@gmail.com*

Resumen—

Este artículo presenta una propuesta para la implementación de plantas virtuales utilizando dispositivos de bajo costo y software libre. El objetivo es desarrollar una herramienta para la enseñanza en las asignaturas de las ciencias de ingeniería, donde los estudiantes puedan interactuar y tener más proximidad con los procesos industriales y no se limiten a depender de los laboratorios físicos. Laboratorios que por lo general tienen una disponibilidad limitada y siempre requieren de la asistencia de un profesional. Para la concepción y desarrollo de la propuesta se plantean los requerimientos que se tuvieron en cuenta en el desarrollo de la planta virtual y se explica cómo programarla y ponerla en operación de tal manera que sea una herramienta didáctica que permita simular, observar y analizar el comportamiento de procesos industriales.

Palabras Clave— Planta virtual, Red de Petri, Automatización, PLC, Sketchup, 3D.

Recibido: 5 de febrero de 2016. Revisado: 29 de marzo de 2016.
Aceptado: 4 de abril de 2016.

A proposal for implementing virtual plants for teaching logic control programs

Abstract—

In this paper, a proposal for the implementation of virtual plants using low-cost devices and free software is presented. The purpose is to develop a tool for teaching in the subjects of engineering sciences, where students can interact and have more proximity to industrial processes, thus not dependent on the real laboratories. In general, these laboratories have limited availability and always require the assistance of a professional. The article presents general requirements for the design and proposal development, software programming and manner of operation so that obtained a teaching tool to simulate, observe and analyze the behavior of industrial processes.

Keywords— Virtual plant; Petri Net; Automation; PLC; Sketchup; 3D.

1. Introducción

En el aprendizaje de programación de controladores lógicos la manera más adecuada es por medio de la experimentación. Sin embargo los laboratorios y realizar este tipo de pruebas es costoso porque necesita equipo industrial real. Además, si la capacitación es para más personal, se requieren más estaciones de trabajo, multiplicando así sus costos de adquisición, tanto en procesos, como en controladores programables y computadores para su comunicación [1].

En algunos casos se utilizan procesos clásicos como control de nivel de agua, temperatura o velocidad, los cuales suelen ser costosos, además el número de aprendizajes normalmente es muy grande para la cantidad de procesos reales con los que se cuenta; a través de una arquitectura de bajo costo es posible lograr que cada estudiante simule su propio proceso, accediendo a sus variables de estado y finalmente medir el desempeño del sistema [2].

A nivel mundial en los diferentes programas educativos, especialmente los del área técnica como es el caso de las ingenierías, se deben realizar prácticas con equipos de laboratorio y prototipos didácticos como herramienta fundamental dado que es requerido consolidar los conceptos adquiridos en el aula. Sin embargo, debido a los bajos presupuestos para laboratorios físicos se tienen restricciones de disponibilidad por diferentes causas deteriorando así el proceso de enseñanza aprendizaje. Como consecuencia de las nuevas tecnologías basadas en Internet, la virtualización y la mejora tecnológica de equipos electrónicos, estos pueden ser utilizados para suplir la carencia de laboratorios y además enriquecer el desarrollo de prácticas en espacios y entornos virtuales y remotos con características de mucha flexibilidad y fácil acceso [3].

A nivel industrial, la validación visual se considera como una de las mejores herramientas para mejorar la velocidad y hacer más ágil y flexible la fabricación de productos. Los simuladores virtuales permiten esta validación visual y así verificar programas de control de estos procesos [4], además evitan desperdicios de materia prima, gastos energéticos y posibles daños en maquinaria debido a errores en la programación del controlador.

Los Controladores Lógicos Programables (PLC) fueron inventados en la década de los 60's y han sido altamente utilizados para tareas de automatización, específicamente para procesos secuenciales, es decir, sistemas dinámicos que cambian sus estados de acuerdo a la ocurrencia de eventos. Este tipo de sistemas muestra características de sistemas a eventos discretos [4]. Los PLC son los controladores industriales por excelencia, sin embargo a nivel mundial, automatizan también

Como citar este artículo: Vásquez-Salazar, R.D., Sarmiento-Maldonado H.O. y Muñoz-Muñoz D.S., Propuesta de implementación de plantas virtuales para la enseñanza de programas de control lógico, Rev. Educación en Ingeniería, 11 (22), 46-51, Julio, 2016.

otro tipo de sistemas no industriales como ascensores, parqueaderos, edificios inteligentes, etc. [5].

Para el diseño de los algoritmos de control en PLC se utilizan técnicas empíricas que llevan a término un correcto control de la máquina a través de la prueba y error. Una técnica organizada basada en el modelo del sistema y luego en la síntesis de un controlador permitiría una programación más organizada y sistemática. Para modelar un sistema del tipo mencionado se debe recurrir a la teoría de Sistemas a Eventos Discretos (SEDs), la cual abarca temas como la teoría de lenguajes, conjuntos de eventos, teoría de autómatas, modelos con Redes de Petri (RdeP) y GRAFCET, etc, donde los programas de control lógico deben ser compatibles con el estándar IEC61131 [6], de modo que contenga sintaxis y semántica unificada.

En conclusión, los laboratorios reales tienen los siguientes inconvenientes [3], los cuales se podrían solucionar a través de un laboratorio virtual:

- Altos costos de instalación y mantenimiento
- El estudiante o practicante tiene restricciones para modificar variables de entrada y configuración del sistema
- Impide obtener las competencias en un corto tiempo ya que no permite cambios e innovaciones en el proceso de enseñanza.
- No se puede experimentar libremente sin miedo a sufrir o provocar un accidente
- Tiempos limitados en disponibilidad.
- No propicia el auto aprendizaje, ya que generalmente se requiere la supervisión de un experto.

2. Metodología

La metodología propuesta consiste en una serie de pasos que permitirán lograr la implementación de una práctica virtual. Inicialmente se debe seleccionar un programa informático para la construcción del proceso virtual. Esta selección se realiza a partir de una búsqueda amplia de la oferta disponible en el mercado, de lo cual se espera obtener una gran cantidad de opciones. Para refinar la selección, se deben definir unos requisitos o exigencias, permitiendo así un filtrado y elegir entonces aquel que más se acerca a las necesidades particulares. Una vez seleccionada la herramienta informática, se procede con la selección de un proceso, de aplicación industrial, que contenga elementos típicos de procesos reales, de modo que la práctica se convierta en un ejercicio lo más parecido a la realidad. La construcción del proceso requiere que se incluya una comunicación con el PLC real, lo cual podría requerir el diseño de una interfaz gráfica. Terminado de construir el proceso, se considera finalizada la etapa de modelamiento. La siguiente etapa es la de control. Este controlador se desarrolla mediante la teoría RdeP, el cual puede ser traducido fácilmente a programa de PLC utilizando el método propuesto en [7]. Se programa en lenguaje ladder y se valida el controlador.

2.1. Selección del programa informático para la construcción del proceso virtual

Siguiendo la metodología, se definen requerimientos para refinar la búsqueda, estos son:

Freeware: es preferible pero no necesario, es decir, para el desarrollo de los procesos de forma virtual, se puede utilizar una herramienta licenciada pero que, al momento de ser usada por el cliente, éste cuente con un runtime o ejecutable que no genere costos por concepto de licencias.

Requiere inversión: la inversión se analiza desde el punto de vista del cliente y del servidor. Para el cliente es indispensable el uso de herramientas sin costo, o con runtime o versión demo.

Editable: Este aspecto indispensable es el que permite construir una amplia gama de sistemas o procesos para virtualización. Se evaluarán de acuerdo a su versatilidad, dependiendo si se pueden incluir diferentes tipos de procesos, preferiblemente industriales. Estos sistemas estarán ubicados en el servidor.

Facilidad de manejo: este factor posibilita al cliente usar y poner en marcha la herramienta de una forma intuitiva o utilizando un reducido número de instrucciones. Por lo tanto se evitará el uso de herramientas complejas.

Cantidad I/O: una herramienta con un reducido número de entradas y salidas, sólo permitiría la virtualización de pequeños sistemas, haciendo que el proyecto tenga un alcance limitado y nivel de complejidad bajo.

Comunicación (protocolos): dado que son plataformas informáticas para uso en computadores personales, se prefieren protocolos que comúnmente estén incluidos en estos equipos, como TCP/IP y USB.

Calidad de gráficos: para una adecuada presentación gráfica de los procesos virtuales, se prefieren herramientas que permitan modificar colores a los objetos, añadir texturas y visualización 3D.

Regulación continua: la finalidad de esta característica es que se puedan realizar prácticas con controladores PID o más avanzados si se desea.

Compatibilidad con diferentes fabricantes: para evitar la dependencia de marcas o fabricantes, se prefieren herramientas que tengan una alta compatibilidad.

Tabla 1. Comparación de herramientas para virtualización de procesos

	Freeware	Requiere inversión	Editable	Facilidad de manejo	Cantidad I/O	Comunicación (protocolos)	Calidad de gráficos	Regulación continua	Compatibilidad con diferentes fabricantes
ITS PLC	No	Si	No	Fácil	11DI/8DO	USB	Alta	No	Si
FACTORY I/O	No	Si	Si	Fácil	Desconocido	Varios	Alta	Desconocido	Si
LABVIEW	No	No	Si	Fácil	Ilimitado	Varios	Alta	Si	Si
BITWINE	No	Si	No	No aplica	Ninguna	No	Alta	No	No
SIMIT SCE	No	Si	Si	Medio	Desconocido	MPI	Baja	No	No
VIRTUAL PLANT DEVELOPMENT	No	Si	No	No aplica	Ninguna	Ninguno	Alta	No	No
ROCKWELL SPECTRUM	Si	No	Si	Fácil	Ilimitada	TCP/IP, otros	Alta	Si	Si
FLUIDSIM	No	Si	Si		8in/8Out	Varios	Baja	No	Si
ATOMATION STUDIO	No	Si	Si		8in/8Out	Varios	Media	No	Si
PROTEUS	No	Si	Si		Desconocido	Serial	Media	No	Si
AUTOMGEN	No	Si	Si		Desconocido	Varios	Alta	No	Si
SOFTWARE CAD	No	Si	Si	Difícil	Ilimitado	Varios	Alta	Si	Si
COSIMR	No	Si	No	Fácil	Desconocido	Desconocido	Media	No	No
DELMIA	No	Si	Si	Difícil	Desconocido	Desconocido	Alta	No	Si
EM-PLC	No	Si	Si	Moderado	Desconocido	Desconocido	Alta	No	No
VIRTUAL AND REMOTE LAB. IN PROCESS OF CONTROL EDUC.	No	Si	No	Si	Limitada/ Baja.	Varios	Media	Si	No
ILABS	No	Si	Si	Moderado	Desconocido	Varios- (Internet).	Desconocido	Si	Si
INTERACTIVE VIRTUAL LAB. FOR CONTROL EDUC.	Si	Si	No	Si	Desconocido	Desconocido	Baja	Si	No
NETLAB	No	Si	No	Si	Limitado	Varios. Interno GPIB (IEEE 488.2 standard Interface).	Media	Si	No

Fuente: Los autores

Para determinar que software utilizar para el desarrollo de las plantas virtuales fue necesario analizar cada uno de los programas que se mencionan en la Tabla 1, y así determinar cuál se ajusta a las necesidades planteadas.

Debido a que se requiere una herramienta que se adecue a la totalidad de las necesidades del proyecto, la investigación y análisis realizado permitió seleccionar a Google Sketchup (ver fila sombreada en la Tabla 1) como la herramienta más adecuada para el desarrollo de las plantas virtuales. Google Sketchup es una herramienta especialmente diseñada para arquitectos, diseñadores, constructores, fabricantes e ingenieros, enfocada principalmente hacia el diseño de proyectos en 3D [8]. Se caracteriza por ser una herramienta sencilla, fácil de manejar, de licencia libre, sin perder de vista la funcionalidad y herramientas que necesita un software para Diseño Asistido por Computador (CAD).

2.2. Selección y modelado del proceso

Los PLC utilizan entradas y salidas (periferia) para conocer estados del proceso y controlarlo. Las entradas reciben información a través de sensores y mandos, y el programa lógico interno evalúa esta información, toma decisiones y finalmente envía órdenes a través de las salidas, las cuales controlan los actuadores del proceso.

Los dispositivos y variables más comunes en el ambiente industrial son: Sensores tipo ON/OFF (con señal de 24V – ‘1’ lógico – cuando detectan la variable, o una señal de 0V – ‘0’ lógico – cuando no la detectan), mandos tipo ON/OFF, actuadores tipo ON/OFF, y sensores y actuadores analógicos. Los sensores analógicos más comunes, entregan señales de voltaje entre 0 a 10V, o de -10 a 10V y señales de corriente entre 0 a 20mA, o de 4 a 20mA (V=voltaje, A=amperios).

Dependiendo de los periféricos incluidos, se pueden definir diferentes niveles de complejidad. El modelado de un proceso de baja complejidad debería incluir únicamente sensores, mandos y actuadores del tipo ON/OFF. Este tipo de procesos sería útil a nivel de educación media y primeros 4 semestres de carreras profesionales. Un proceso de moderada complejidad debería incluir además sensores y actuadores analógicos, adecuados para semestres más avanzados de carreras profesionales, o estudiantes de educación media con una adecuada preparación a nivel técnico. Un proceso de complejidad alta incluiría variables continuas y su respectiva regulación (control), lo cual requiere un modelado y síntesis del controlador; este tipo de sistemas sería apropiado para estudiantes de últimos semestres de carreras profesionales. En un nivel mucho más elevado, estarían sistemas que involucren sistemas no lineales, cuyo control requiere de técnicas no convencionales, adecuado para niveles de posgrado.

Para este desarrollo se elegirá una complejidad moderada, con el fin de tener un proceso lo suficientemente adecuado para carreras profesionales y educación media. Los elementos que se incluirán son: Sensores tipo ON/OFF, Actuadores tipo ON/OFF, Actuadores analógicos (motores de corriente alterna y servomotor).

Dada la alta utilización de sistemas de transporte para materiales o productos en una gran cantidad de procesos

industriales, se modelará una banda transportadora que incluya un control con un nivel de complejidad medio.

El proceso contará con un variador para modificar la velocidad a través de facetas automáticas, secuencias programadas o controladores continuos, cilindros neumáticos que se encarguen de mover las cajas hacia adentro o afuera de la banda, junto con sensores instalados en el recorrido para poder hacer un control lógico en lazo cerrado, una mesa giratoria controlada por un servomotor, de modo que las cajas se puedan ubicar en diferentes posiciones sobre su superficie. Se definen para el proceso las siguientes etapas:

Suministro de cajas apiladas (etapa 1): Consiste en una pila de 18 cajas apiladas una encima de la otra. En la parte inferior un cilindro neumático se encarga de suministrarlas una por una en una banda transportadora. Un sensor permite conocer si hay caja disponible en la pila

Banda transportadora 1 (etapa 2): Su tarea principal es transportar las cajas desde la pila hasta la mesa giratoria. Cuenta con un cilindro neumático que mueve las cajas hacia la mesa. Además tiene 2 sensores, uno en mitad del recorrido y otro al final junto a la mesa.

Mesa Giratoria (etapa 3): Su tarea es organizar 4 cajas en forma de estiba, de modo que puedan ser transportadas en grupos de 4. Las cajas son depositadas en cada uno de sus 4 cuadrantes, por lo que la mesa debe girar 90° cada que entre una nueva caja. Un cilindro mueve la mesa a una posición inferior para poder hacer rotación sin que colisione con otros elementos del proceso.

Banda Transportadora 2 (etapa 4): Finalmente, esta banda tiene como tarea transportar las 4 cajas organizadas hacia una etapa posterior del proceso que no se modeló en este artículo. Esta es la parte final del proceso a controlar

El proceso completo diseñado y ubicado en un ambiente que simula una instalación industrial se muestra en la Fig. 1 donde se identifican cada uno de las etapas mencionadas.

2.3. Modelo planta-controlador mediante Red de Petri (RdeP)

El modelado del sistema se desarrolló a partir de la asignación de variables de entrada y salida a cada uno de los sensores y actuadores involucrados en el proceso, a cada variable le corresponde una dirección de PLC. Además se deben definir memorias internas (marcas) a cada uno de los estados (o lugares) de la RdeP y las entradas necesarias para los mandos de start y stop. El resultado son las Tablas 2, 3 y 4.

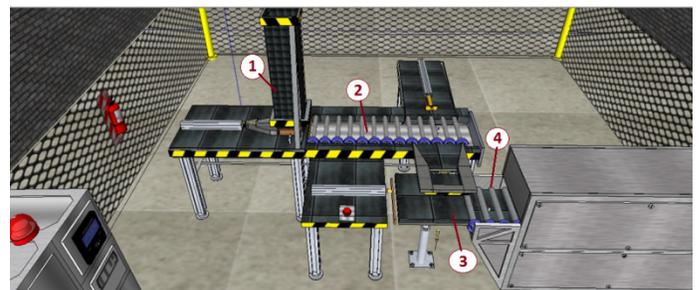


Figura 1. Desapiladora completa construida con Google SketchUp

Fuente: Los autores

Tabla 2. Variables de entrada del proceso

Variable	Tipo de Variable	Dirección	Descripción
SExpCil1	Booleana	I0.0	Sensor Expuls Cil1
SContCil1	Booleana	I0.1	Sensor Contrac Cil1
SCajaPila	Booleana	I0.2	Sensor Caja
SCajaMitadB1	Booleana	I0.3	Sensor Mitad Banda 1
ScajaFinalB1	Booleana	I0.4	Sensor Final Banda 1
SCajaB2	Booleana	I0.5	Sensor Caja Banda 2
Start	Booleana	I0.6	Start
Stop	Booleana	I0.7	Stop
SElevMesa	Entero (16 bits)	IW66	Sensor Elevación Mesa
SCil2	Entero (16 bits)	IW112	Sensor Cil2
SCil3	Entero (16 bits)	IW114	Sensor Cil3

Fuente: Los autores

Tabla 3. Variables de salida del proceso

Variable	Tipo de Variable	Dirección	Descripción
ExpCil1	Booleana	Q0.0	Expulsión Cilindro 1
ContCil1	Booleana	Q0.1	Contracción Cilindro 1
HabMotorB1	Booleana	Q0.2	Habilitar Motor Banda 1
ExpCil2	Booleana	Q0.3	Expulsión Cilindro 2
ContCil2	Booleana	Q0.4	Contracción Cilindro 2
ExpCil3	Booleana	Q0.5	Expulsión Cilindro 3
ContCil3	Booleana	Q0.6	Contracción Cilindro 3
HabMotorB2	Booleana	Q0.7	Habilitar Motor Banda 2
MotorGiroMesa	Entero (16 bits)	QW64	Motor Mesa Girar
MotorB1	Entero (16 bits)	QW66	Motor Banda 1
CilMesa	Entero (16 bits)	QW112	Cilindro Mesa
MotorB2	Entero (16 bits)	QW114	Motor Banda 2

Fuente: Los autores

Tabla 4. Estados de la red de Petri y otras marcas

Estado	Tipo de Vble	Dir	Acciones
Reposo	Booleana	M0.0	
Suministro Caja Banda 1	Booleana	M0.1	Expulsa Cilindro 1 y Activa Banda 1
Caja en Banda 1	Booleana	M0.2	Contracción Cilindro 1
Caja en Pila	Booleana	M0.3	
Cilindro 2 Reposo	Booleana	M0.4	
Suministro Caja a Mesa	Booleana	M0.5	Expulsión Cilindro 2
Caja en Mesa	Booleana	M0.6	Contracción Cilindro 2
Mesa en reposo	Booleana	M0.7	Subir y poner mesa en posición inicial
Bajando Mesa	Booleana	M1.0	Bajar Cilindro Mesa
Girando Mesa	Booleana	M1.1	Girar mesa 90°
Posición Mesa	Entera	MW10	Definir la posición de la mesa (0°=0, 90°=6912, 180°=13824, 270°=20736)
Subiendo Mesa	Booleana	M1.2	Subir Cilindro Mesa
Bajando Mesa Banda 2	Booleana	M1.3	Bajar Cilindro Mesa
Mesa en Banda 2	Booleana	M1.4	
Mesa Abajo	Booleana	M1.5	
Mesa a Pos Inicial	Booleana	M1.6	Volver mesa a 0°
Mesa Arriba Final	Booleana	M1.7	
Reposo Cil 3 y Banda 2	Booleana	M2.0	Banda y Cilindro 3 en reposo
Expulsión cajas	Booleana	M2.1	

mesa			
Cajas en Banda 2	Booleana	M2.2	
Contracción Cilindro Mesas	Booleana	M2.3	Contraer Cilindro 3
Espera Banda 2 Vacía	Booleana	M2.4	

Fuente: Los autores

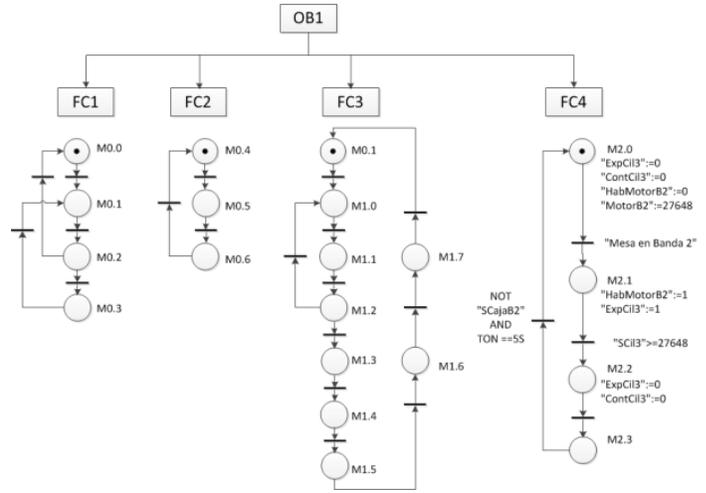


Figura 2. Modelos de Red de Petri completos

Fuente: Los autores

Los modelos en RdeP para controlar el proceso están organizados en submodelos (etiquetados desde FC1 a FC4) y se observan en la Fig. 2. En el submodelo FC4, como ejemplo, se muestran además los diagramas de bloques que se usaron en el PLC con sus respectivas transiciones y estados.

3. Comunicación

Terminado el modelo planta-controlador, la siguiente y última etapa es la comunicación entre el modelo y el PLC donde el programa LabVIEW es utilizado como interface. En SketchUp se hace una programación de la comunicación TCP/IP con el localhost de LabVIEW mediante una Interface de Programación de Aplicaciones (API) con lenguaje de programación Ruby que viene incluido, así se puede intercambiar información de actuadores (órdenes de posición de los cilindros y velocidad de las cajas) y de sensores (posición de las cajas). Una vez LabVIEW contiene toda esta información, se puede hacer una comunicación TCP/IP con el PLC, el cual debe contar con un puerto de comunicación Ethernet (Fig. 3).

4. Validación del controlador en planta virtual

Teniendo en cuenta los diseños anteriores, se procede a implementar el sistema completo, integrando PLC, tarjeta de adquisición de datos, software TIA Portal V13, software SketchUp y software LabVIEW. La programación del PLC se realiza siguiendo las redes de Petri expuestas anteriormente, lo cual permite hacer una programación estructurada por bloques FC en el PLC, así:

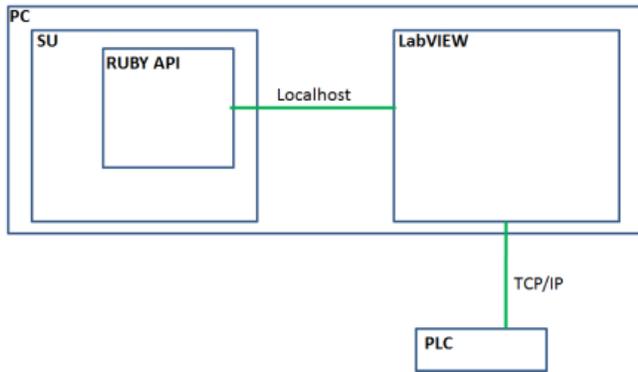


Figura 3. Esquema de comunicación Sketchup-LabVIEW-PLC
Fuente: Los autores

Bloque de arranque (OB100): este bloque permite que los estados de las redes de Petri partan de condiciones iniciales cuando el PLC pasa del modo Stop al modo Run, por lo tanto asigna valores numéricos a los bytes MB0, MB1 y MB2 que son convertidos internamente a binarios por el PLC.

Bloque principal (OB1): genera el paro de emergencia y activa todos los bloques FC1 al FC5. Dado que no es una red de Petri jerárquica, no es necesario que haga una sincronización activando y desactivando bloques, sino que ellos permanecen activos permanentemente.

Pila y Banda 1 (FC1): el bloque FC1 controla el suministro de cajas de la pila y a su vez la banda 1. Esta banda está sincronizada con el cilindro 2, puesto que la banda debe detenerse cuando el cilindro 2 suministre cajas sobre la mesa.

Cilindro 2 (FC2): el bloque FC2 controla el cilindro 2, moviendo las cajas de la banda 1 y llevándolas a la mesa giratoria. Su movimiento únicamente se puede hacer cuando la mesa se encuentre en la posición superior, porque de lo contrario generaría un daño en la caja suministrada.

Mesa (FC3): el bloque FC3 se encarga de subir y bajar la mesa, y a su vez de girarla para ubicar cajas en cada uno de sus cuadrantes.

Cilindro 3 y Banda 2 (FC4): el bloque FC4 se encarga de administrar el cilindro 3 para suministrar cajas sobre la banda 2, y a su vez de controlar la velocidad de esta última.

La validación se completa cuando se pueda evidenciar que la Red de Petri traducida a código PLC, la comunicación entre los diferentes programas involucrados y todo el sistema en conjunto ha funcionado satisfactoriamente. Como soporte de validación está disponible en línea el video referenciado en [9]

5. Resultados

Se realizaron 13 pruebas con estudiantes y egresados de programas de Ingeniería del área de Electrónica, Automatización, Instrumentación o Control, de 4 instituciones diferentes: Politécnico Colombiano Jaime Isaza Cadavid (Medellín – Antioquia), Universitaria de Investigación y Desarrollo (Bucaramanga – Santander), Instituto Tecnológico Metropolitano (Medellín – Antioquia) y Universidad Católica del Norte (Antofagasta – Chile). A cada estudiante se le proporcionó un tiempo de 4h en los cuales podían acceder al computador configurado en el Politécnico, incluyendo una guía

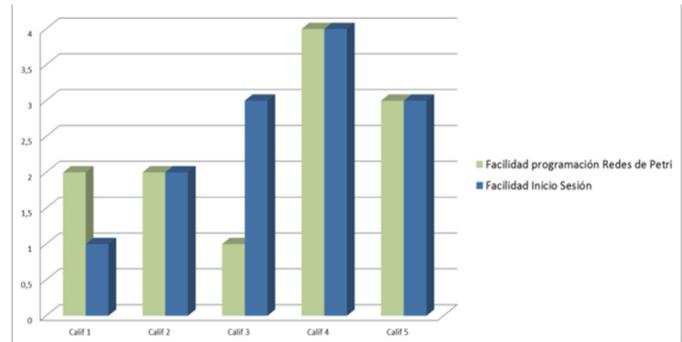


Figura 4. Respuesta a formulario de validación (Facilidad)
Fuente: Los autores

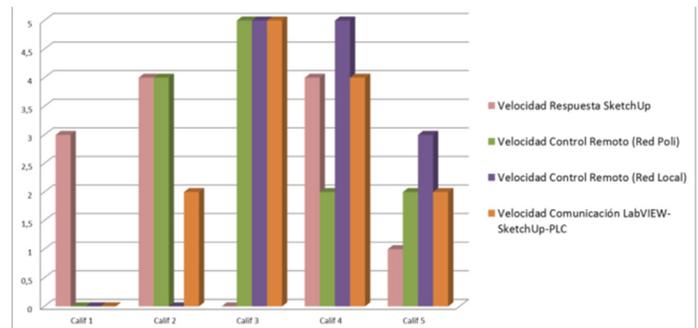


Figura 5. Respuesta a formulario de validación (Velocidad)
Fuente: Los autores

de acceso y configuración, el cual fue suficiente para que todos pudieran acceder. Una vez terminada la práctica debían diligenciar un formulario de validación, con calificación de 1 (la más baja) a 5 (la más alta), donde se les interrogaba principalmente 4 aspectos: Facilidad, Velocidad, Estabilidad y Gusto, este último haciendo referencia a su percepción y agrado de la metodología propuesta

Respecto a la facilidad en relación con la programación estructurada propuesta mediante redes de Petri y a su vez sobre el inicio de sesión en el servidor, la calificación más sobresaliente fue 4 (Fig. 4), sin embargo en los comentarios no se realizaron observaciones a dificultades en estos aspectos.

En velocidad, las observaciones relacionan algunos problemas de lentitud del equipo cuando corren los 3 programas simultáneamente, acción necesaria para el correcto funcionamiento de la propuesta, esto genera una apreciación general de lentitud en el funcionamiento de todo el sistema (Fig. 5). Se evidenció que el equipo utilizado tiene una tarjeta de red de mediana capacidad y por lo tanto se concluye que es requerida una con mejores especificaciones para que el usuario remoto no tenga la percepción de lentitud, especialmente en el programa SketchUp que consume más recursos por ser visualización 3D.

La percepción de estabilidad por parte de los estudiantes fue buena, en el sentido que no se tuvieron inconvenientes de caída de la conexión a internet o sesión en el servidor, la percepción en general fue muy buena (Fig. 6).

Como una última pregunta, sobre la percepción del gusto de la propuesta, la gran mayoría indica que tendría interés de continuar con ejercicios y prácticas siguiendo esta metodología, además de manifestar claramente que fue de su agrado (Fig. 7).

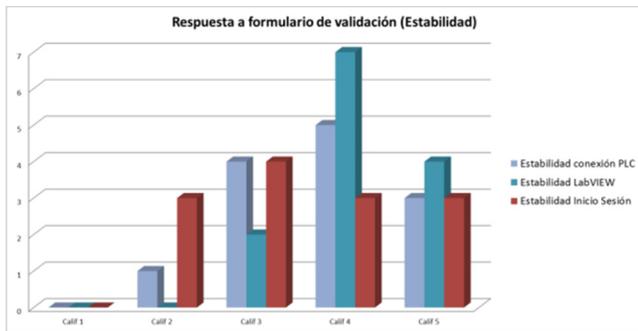


Figura 6. Respuesta a formulario de validación (Estabilidad)
Fuente: Los autores

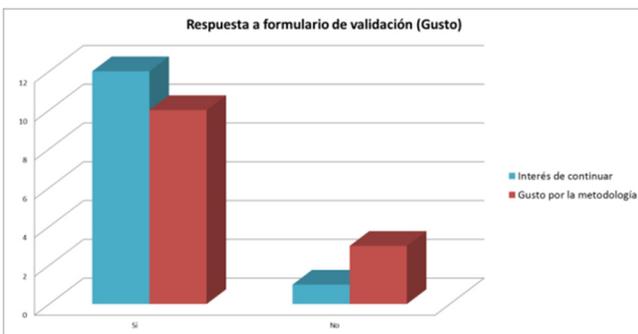


Figura 7. Respuesta a formulario de validación (Gusto)
Fuente: Los autores

6. Conclusiones

Las plantas virtuales presentan viabilidad técnica y práctica puesto que son posibles las prácticas remotas en las asignaturas de ciencias de la ingeniería, aumentando la experiencia de los estudiantes frente a los procesos industriales. En los experimentos realizados con estas estrategias, las simulaciones no sólo sirvieron a los estudiantes como un primer contacto con los fenómenos en estudio, sino también como una forma de obtener datos teóricos que pueden ser usados más tarde en comparación con datos reales, adquiridos de la experimentación a distancia.

Las experiencias remotas permiten a los estudiantes comprobar empíricamente algunas leyes y fenómenos inherentes a partir de un experimento real y comparar estos resultados con los teóricos. Aunque estos experimentos son bastante simples, los estudiantes pueden aprender no solo de su uso, sino también temas como son medición de las variables, el análisis de las señales del proceso, los cálculos de incertidumbres y demás información que se pueda obtener del proceso.

Las condiciones técnicas del equipo servidor, al igual que la calidad de la conexión a internet hacen que la percepción de los usuarios de esta propuesta se vea afectada directamente, por lo tanto es indispensable contar con un equipo de las más óptimas especificaciones y una conexión a internet de buena velocidad y estabilidad.

Referencias

[1] Salazar, E. and Macías, M., Virtual 3D controllable machine models for implementation of automations laboratories, in 39th ASEE/IEEE

Frontiers in Education Conference, San Antonio, TX, 2009, Session T2C.
 [2] Bamidele-Alaba, O., Maximilian N., Dobrescu, R. Asimiyu-Abass, O. and Popescu D., Educational platform for process controlling via internet, 2nd International Conference on Adaptive Science & Technology, pp. 361-365, 2009.
 [3] Lorandi, A., Hermida, G., Hernández, J. y Ladrón, E., Los laboratorios virtuales y laboratorios remotos en la enseñanza de la ingeniería, Academia Journals 4, pp. 24-30, 2011. ISSN 1940-1116
 [4] Park, C.M., Bajimaya S.M., Han, K.H. and Chang M., Development of virtual simulator for visual validation of PLC program, in: Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Sydney, 2006.
 [5] Muñoz, J., Guzmán, J., Rodríguez, F., Berenguel, M and Pawlowski, A., Virtual lab for programmable logic controllers, 2009.
 [6] International Electrotechnical Commission IEC. Estándar IEC 61131. 3 ra Ed. [Online]. 2013. Available at: http://webstore.iec.ch/webstore/webstore.nsf/Artnum_PK/47556
 [7] Vásquez, R., Control lógico programable. Medellín: Fondo Editorial ITM, 2010. ISBN 978-958-8351-95-7.
 [8] GOOGLE INC. Sketchup 3D for Everyone. 2013 [Online]. Available at: <http://www.sketchup.com/es>
 [9] Vásquez, R., Sarmiento, H. y Muñoz, D., Video Desapiladora con PLC y TCP. Septiembre, 2014. [Online]. Available at: <https://www.youtube.com/watch?v=cxdiydmbj7A>

R.D. Vásquez-Salazar, se graduó como Ing. de Control en la Universidad Nacional de Colombia- Sede Medellín, Colombia, en el año 2007, allí mismo empezó su trayectoria investigativa como auxiliar de proyectos de investigación. Posteriormente se graduó como MSc. en Ingeniería – Automatización Industrial de la Universidad Nacional de Colombia- Sede Manizales, Colombia, en el año 2009. Inició su carrera como docente de tiempo completo en educación superior en el Instituto Tecnológico Metropolitano (ITM) de Medellín, Colombia, en el año 2009, y posteriormente a partir del año 2012 en el Politécnico Colombiano Jaime Isaza Cadavid (PCJIC), en el cual ha sido investigador en varios proyectos de investigación aplicada en el área de la automatización, algunos de ellos con énfasis en la agroindustria, todos ellos con el grupo de investigación ICARO. Durante el año 2015 se desempeñó como Decano (e) Facultad de Ingeniería del PCJIC, actualmente regresó a su cargo como docente del PCJIC.
 ORCID: 0000-0002-1690-8393

H.O. Sarmiento-Maldonado, se graduó como Ing. Electricista de la Universidad de Antioquia, Colombia, en 1996. Continuó estudios de Especialización en Ciencias Electrónicas e Informática hasta el año 2000 en la línea de énfasis en Automatización Industrial. Posteriormente realizó estudios de Maestría en la Universidad de Antioquia en la línea de énfasis en Energética graduándose en 2008, y siendo integrante del grupo de investigación GIMEL de la misma universidad. Finalmente, se graduó como Dr. en Ingeniería Electrónica de la Universidad de Antioquia, Colombia, en 2014 por la línea de Automatización y Control adscrita al Grupo de Investigación GEPAR de la misma universidad. Fue docente ocasional de la Universidad de Antioquia desde el año 1996 hasta el año 2007, en que ingresó a ser docente asociado (hasta la fecha) del Politécnico Colombiano Jaime Isaza Cadavid e integrarse al grupo de investigación ICARO. En este momento se desempeña como Director de Investigación y Posgrados del Politécnico, y sus áreas de interés son las técnicas de inteligencia artificial, y el control y automatización de procesos.
 ORCID: 0000-0001-8011-1293

D.S Muñoz-Muñoz, se graduó como Ing. en Instrumentación y Control del Politécnico Jaime Isaza Cadavid, Medellín, Colombia, en el año 2014, siempre estuvo vinculado al campo industrial debido a su preparación como Tecnólogo en Electrónica Industrial del Servicio Nacional de Aprendizaje SENA en 2007, la cual le permitió tomar experiencia y formar su propia empresa; en 2015 tuvo la oportunidad de empezar como joven investigador Facultad de Ingeniería del Politécnico Colombiano Jaime Isaza Cadavid, debido a su interés de vincular proyectos industriales con la academia, por tanto ingresó como miembro del grupo de investigación ICARO del Politécnico Colombiano Jaime Isaza Cadavid, área de Instrumentación, Control Automático y Robótica. Actualmente se encuentra adelantando sus estudios de Maestrando en Automatización y Control en la Universidad Federal de Santa Catarina – Brasil.
 ORCID: 0000-0001-8537-3081