

Plataforma para Laboratorios Virtuales y Remotos Orientados a la Simulación de Sistemas Dinámicos Controlados

María Julia Blas¹, Ariel S. Loyarte¹, Jorge R. Vega²

^{1,2}Laboratorio de Sistemas de Control, Departamento de Ingeniería Eléctrica, Facultad Regional Santa Fe, Universidad Tecnológica Nacional
¹{mariajuliablas, ariel.loyarte}@gmail.com
²jvega@santafe-conicet.gov.ar

Abstract. Este trabajo presenta la implementación de una plataforma que da soporte a laboratorios virtuales y remotos basados en plantas de experimentación configuradas en lazo cerrado. La propuesta apunta a cubrir una necesidad educativa existente en las asignaturas relacionadas al área de control automático, ofreciendo numerosas ventajas tanto para alumnos como para docentes de estas disciplinas. El acceso concurrente, la práctica a distancia de ensayos, la interactividad y un mecanismo de configuración simple e intuitivo, son algunos de los beneficios que brinda la plataforma a los distintos laboratorios. La arquitectura diseñada para lograr la comunicación remota, junto con la selección de tecnologías, especificación de modelos de simulación, y el desarrollo de los principales componentes, son algunos de los temas que se presentan. Se describe de esta manera el desarrollo de la herramienta planteada, dejando abierta la posibilidad de innovar en plataformas similares para otras disciplinas.

Keywords. Plataforma, laboratorio virtual y remoto, cliente/servidor, modelo de simulación, control automático, lazo de control.

1 Introducción

La Ingeniería de Control desde sus orígenes se preocupó de la automatización y del control automático de sistemas complejos siendo, hoy en día, una disciplina científica y tecnológica imprescindible para el desarrollo de tales sistemas. Su aplicación se ha diversificado a tal punto que hoy se aplica en múltiples campos, relacionándose de cerca con la Ingeniería Eléctrica y Electrónica [5].

Esta área, considera que un laboratorio es una herramienta fundamental del proceso de aprendizaje ya que los alumnos mediante la interacción con el mismo, adquieren una serie de conocimientos que serían difíciles de asimilar mediante el empleo de cualquier otro elemento [3]. Cuando un estudiante trabaja en un laboratorio destinado a la enseñanza, lo hace para aprender “algo” que los ingenieros en ejercicio se supone que ya saben [20]. De esta manera, la experimentación los obliga a involucrarse en el aprendizaje, haciendo que la herramienta se convierta en parte esencial de la enseñan-

za [4,6]. Tradicionalmente, la realización de tales prácticas requiere que el alumno asista en un horario determinado al laboratorio. No obstante, la experimentación con plantas reales suele ser costosa en términos de recursos (tiempo, personal, energía, etc.), requiriendo además la puesta a punto de las infraestructuras y la presencia física del docente junto al alumno [11].

El avance de las tecnologías de la información y la comunicación (TIC) ha transformado radicalmente diversos aspectos de la sociedad, no quedando exento de este cambio el paradigma educacional [7]. La disponibilidad y la capacidad de las nuevas facilidades en la comunicación, combinadas con el rápido desarrollo de las tecnologías Web y la implementación de modelos de simulación; permite al alumno evitar la asistencia a los laboratorios tradicionales (LT), planteando como alternativa el uso de laboratorios virtuales y remotos (LVR). La Reunión de Expertos en Laboratorios Virtuales (París, 2000), define al laboratorio virtual como “un espacio electrónico de trabajo concebido para la colaboración y experimentación con el objeto de investigar o realizar otras actividades creativas”. Expresa además, que un laboratorio remoto “es un sistema basado en la instrumentación de un laboratorio real o simulado, que permite realizar actividades prácticas de forma remota transfiriendo información entre el proceso y el usuario de manera bidireccional. Quien lo usa, controla los recursos disponibles en el laboratorio a través de las estaciones de trabajo de una red local o de Internet”. Estos conceptos permiten definir a un LVR como un espacio electrónico de trabajo concebido para la experimentación con el objetivo de observar el comportamiento de una planta real por medio del uso de un modelo de simulación.

En los últimos años, diversos autores han centrado esfuerzos en la creación de LVR aplicados a diferentes áreas, entre las que se destacan: Química, Física, Control Automático e Inteligencia Computacional [8, 12, 16, 18]. Ya sea para fines académicos o de investigación, los LVR obtenidos han exhibido resultados satisfactorios. No sólo han ampliado el acceso a los LT, facilitando a los usuarios operar sobre los mismos sin poner en riesgo los dispositivos físicos, sino que también han modificado el proceso de experimentación puesto que los usuarios pueden realizar pruebas sobre el LVR antes de actuar sobre el LT. Además han disminuido los costos de cada ensayo, debido a que no se requiere la puesta a punto de la infraestructura ni recursos específicos para su utilización. De esta manera, los LVR permiten multiplicar el número de prácticas y variantes implementadas sobre un sistema, las cuales pueden llevarse a cabo en horarios flexibles, sin la necesidad de asistencia física al LT.

En este contexto, la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional inició en el año 2010 un Proyecto de Investigación y Desarrollo denominado “Laboratorios Virtuales y Reales con Acceso Local y Remoto Destinados a la Enseñanza y la Investigación en Automatización y Control” (PID 25/O113). Uno de los principales objetivos de este proyecto es la implementación de una herramienta que permita, tanto a docentes como alumnos, la manipulación de diferentes LVR relacionados al área de Control Automático. Este trabajo, presenta la plataforma de software desarrollada en respuesta a dicha necesidad, la cual constituye una pieza clave en el funcionamiento de diversos LVR.

En las siguientes secciones se detalla la implementación propuesta junto con su aplicación a un LVR, concluyendo sobre las ventajas y desventajas de la plataforma.

2 Plataforma para LVR

Una plataforma es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o software. Queda definida por un estándar alrededor del cual se establecen los tipos de módulos, sistemas operativos, lenguajes de programación y/o interfaz de usuario compatibles [14]. Como se ha definido previamente, un LVR es aquel laboratorio que permite, al ser accedido por medio de una red basada en el protocolo IP, manipular y observar la conducta de plantas de experimentación reales implementadas en modelos de simulación, dando como resultado el fenómeno simulado. Consta de dos elementos: uno en la máquina del usuario y otro ubicado en una máquina que contiene el modelo de simulación y la herramienta que le da soporte.

Vinculando ambos conceptos, puede establecerse que una plataforma para LVR debe facilitar la manipulación de modelos de simulación desarrollados en diferentes herramientas. Estos modelos deben configurarse en un entorno local y ejecutarse en un entorno remoto que debe dar soporte a los simuladores. Con el objetivo de que el usuario extraiga conclusiones de la experimentación realizada, los resultados de la simulación deben retornar a la máquina local, y la plataforma debe ofrecerle al operador un conjunto de funcionalidades básicas que facilite su interpretación. La Figura 1 esquematiza el intercambio de información entre ambos entornos y los pasos llevados a cabo para ejecutar un LVR particular, evidenciando la interacción existente entre la máquina que contiene la plataforma y la máquina que contiene los laboratorios.

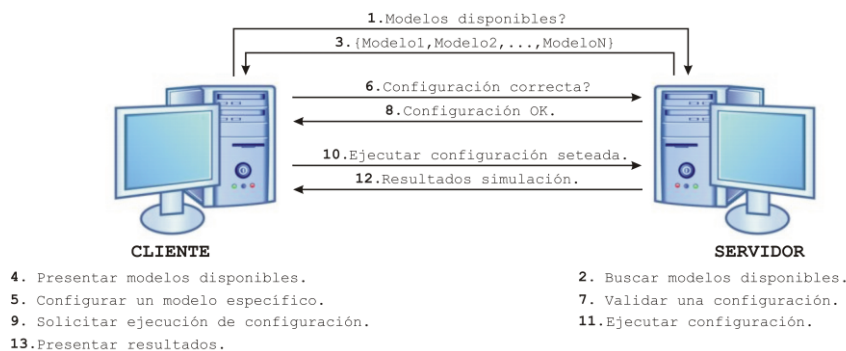


Fig. 1. Pasos para la ejecución de un LVR.

La plataforma implementada como herramienta de aprendizaje para el estudio e investigación de LVR relacionados al área de control automático, funciona de acuerdo al esquema precedente.

2.1 Arquitectura

La arquitectura de la plataforma se definió en base a un esquema cliente/servidor, el cual se enmarca en el estilo arquitectónico de procesos comunicantes [1]. Éste, es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores

de recursos o servicios, llamados servidores; y los demandantes de estos recursos o servicios, llamados clientes [19].

Idealmente ambos componentes deben tener una carga de procesamiento similar, pero debido a que la meta consiste en reflejar el comportamiento de los LVR se optó por trabajar con cargas asimétricas. De esta manera, mientras que el servidor se encarga de gestionar tanto los modelos de simulación como la ejecución de los mismos, el cliente se responsabiliza de gestionar la configuración y los resultados obtenidos. La Figura 2 muestra la arquitectura diseñada en términos de módulos y de mecanismos de comunicación.

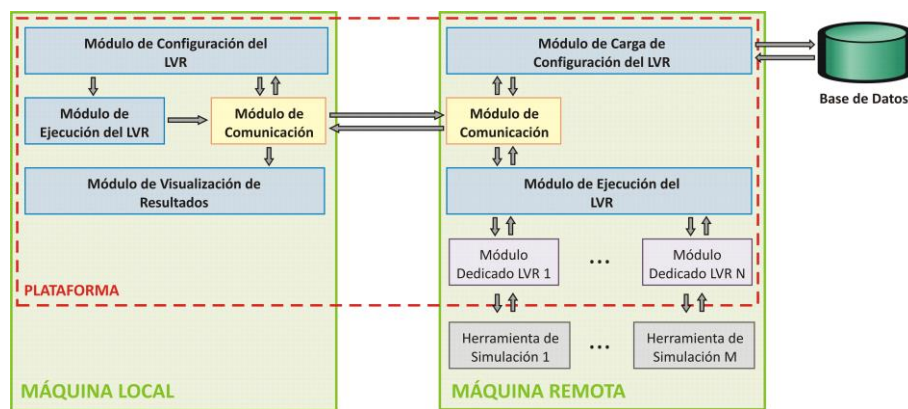


Fig. 2. Arquitectura propuesta.

El nivel de generalización obtenido con este esquema es adecuado dado que facilita la interacción con múltiples simuladores. Cada LVR asociado a la plataforma conlleva un módulo dedicado que tiene por finalidad gestionar el laboratorio. Este componente se encarga de setear los parámetros en el modelo correspondiente, ejecutar la simulación en la herramienta adecuada y empaquetar el resultado de la simulación en un formato entendible por la plataforma. De esta manera, la incorporación de un nuevo LVR a la plataforma requiere sólo una modificación del aplicativo servidor, sin verse afectada la aplicación cliente.

La utilización de esta arquitectura ofrece numerosas ventajas. No sólo aporta escalabilidad, centralización del control y facilidad de mantenimiento; sino que además disminuye los requerimientos computacionales de la máquina cliente. Esto se debe a que las herramientas de simulación deben estar instaladas únicamente en la máquina servidora; siendo transparente al usuario el entorno en el cual se ejecuta el modelo.

2.2 Tecnología

Algunos entornos específicos para el desarrollo de LVR, como por ejemplo Easy Java Simulations (EJS) junto con Java Internet Matlab (JIM) [7- 9], no fueron de utilidad al momento de seleccionar la tecnología con la cual implementar la plataforma propuesta. Esto se debió a que este tipo de herramientas facilita la definición de un único

laboratorio sobre un modelo y entorno de simulación específico; no permitiendo alcanzar la generalidad pretendida en el diseño arquitectónico. Por este motivo, la versatilidad en relación a la manipulación del conjunto de LVR debió modelarse mediante la utilización de un lenguaje de alto nivel en combinación con librerías intermedias que faciliten la comunicación con los simuladores.

Un análisis de las características de los lenguajes de programación más populares en relación a los requerimientos de la plataforma, fue el punto de partida para la selección de la tecnología. La Tabla 1 esquematiza el análisis realizado, junto con la ponderación resultante para cada lenguaje [10].

Tabla 1. Análisis de los lenguajes de programación.

CRITERIO \ LENGUAJE	PESO	C++	C#	JAVA
PORTABILIDAD	10	6	8	10
CAPACIDAD 2D/3D	8	6	4	6
MODULARIDAD	10	8	8	8
MATEMÁTICA DE PRECISIÓN	3	6	4	4
VELOCIDAD	7	10	8	7
LICENCIA	10	10	0	10
INTERACCIÓN CON SIMULADORES	10	8	7	9
TOTAL	58	-	-	-
SUMA PONDERADA	-	456	330	479
MEDIA PONDERADA	-	7.86	5.68	8.25

Como puede observarse, Java es el lenguaje que reúne todos los requisitos planteados con un valor aceptable de madurez. Permite interactuar con múltiples entornos de simulación por medio de librerías especiales, lo que presenta una ventaja puesto que la plataforma requiere de un fuerte intercambio de datos e información con este tipo de componentes. Brinda además un alto nivel de portabilidad, lo que fomenta la usabilidad y contribuye a la realización de un desarrollo independiente del entorno de ejecución. La baja capacidad de cálculo no presenta un problema, puesto que las simulaciones no se llevarán a cabo en el lenguaje sino que lo harán en las herramientas de simulación. Teniendo en cuenta el resultado de la ponderación realizada, se decidió trabajar con tecnología Java para la implementación de la plataforma.

2.3 Gestor de Base de Datos

Dada la necesidad de implementar una base de datos que almacene la información relevante de los LVR, se realizó una evaluación de los gestores existentes con el objetivo de determinar cuál de ellos es el más apropiado para la plataforma.

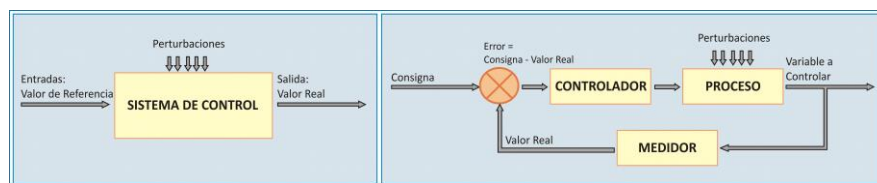
La elección de un Sistema de Gestión de Base de Datos queda determinada por las necesidades específicas de la aplicación a implementar. En la actualidad, MySQL y PostgreSQL son las opciones más difundidas; radicando su diferencia en la imple-

mentación de SQL y la velocidad con la que realizan las consultas. Expertos han señalado que si se demanda rapidez y características sencillas la elección es MySQL; mientras que, para sistemas complejos en los que el rendimiento es una cuestión secundaria debe optarse por PostgreSQL [2, 17].

Debido a que el conjunto de datos a resguardar en la plataforma no es significativo ni complejo, puesto que sólo referencia a las características generales de los modelos de simulación que representan a los laboratorios, se decidió utilizar MySQL.

2.4 Estándar para los Modelos de Simulación

Teniendo en cuenta que la plataforma debe dar soporte a LVR relacionados al área de Control Automático, se realizó un estudio de las configuraciones existentes en los LT de esta disciplina. Se analizaron dos posibles configuraciones para sistemas de control: lazo abierto (Figura 3(a)) y lazo cerrado (Figura 3(b)) [15, 21].



(a) Lazo abierto.

(b) Lazo cerrado.

Fig. 3. Posibles configuraciones de plantas de experimentación.

En lazo abierto, la acción de control se calcula conociendo la dinámica del sistema, las entradas y estimando las perturbaciones. No existe retroalimentación hacia el controlador para que ajuste la acción de control. Esta estrategia suele ser insuficiente debido a los errores del modelo y a los errores en la estimación de las perturbaciones.

En lazo cerrado, la acción de control se calcula en función del error medido entre la variable controlada y la consigna deseada, no requiriendo que ambas señales sean de la misma naturaleza. Las perturbaciones, aunque sean desconocidas, son consideradas indirectamente mediante sus efectos sobre las variables de salida. De esta manera, esta estrategia permite capturar señales del estado del sistema bajo control, y actuar sobre el mismo mediante la utilización de sensores y actuadores.

Comparando la utilización de ambas configuraciones, se observó que mientras el lazo abierto es aconsejable para controlar sistemas en los que se conocen con anticipación las entradas y en los cuales no existen perturbaciones; los sistemas de control en lazo cerrado tienen numerosas ventajas cuando se presentan perturbaciones y/o variaciones impredecibles en los componentes del sistema. Además, estos últimos pueden aplicarse en cualquier tipo de variable controlada, siendo la gran mayoría de los sistemas de control que se desarrollan en la actualidad configurados en lazo cerrado. Por estos motivos, se optó por diseñar la implementación de la plataforma en base a plantas de experimentación configuradas a lazo cerrado. Sin embargo, una adecuada configuración de los componentes permitirá transformar su accionar en lazo abierto.

Componentes del Modelo

Aunque la plataforma restringe los modelos de simulación a sistemas de control en lazo cerrado, no limita la definición de sus componentes. Diversos tipos de señales de entrada, controladores, actuadores, medidores y perturbaciones pueden modelarse como parte del mismo, quedando exento el aplicativo cliente de este conocimiento. Un LVR, presentará al usuario una planta de experimentación con el conjunto de posibles dispositivos a ser utilizados para controlar el sistema. Será el operador quien defina la configuración deseada para la experimentación indicando posteriormente el valor de los parámetros para cada componente. De esta manera, la plataforma es apta para simular tanto sistemas continuos como discretos, con o sin retardos, etc.; ya que la naturaleza de los mismos se verá reflejada en el modelo de simulación y no en los aplicativos cliente/servidor.

2.5 Implementación

En base a la arquitectura previamente definida, se implementaron en Java los módulos correspondientes a las aplicaciones cliente y servidor. Se detallan a continuación los módulos cuyo accionar es clave en el funcionamiento de la plataforma.

Módulo 1: “Configuración de Modelos/LVR”

La definición de este módulo se realizó en término de un conjunto de clases genéricas que representan los componentes de una configuración válida (Figura 4). Éstas, permiten al aplicativo cliente instanciar una configuración específica y posteriormente determinar la validez del conjunto de parámetros seteados por el usuario.

Módulo 2: “Ejecución de LVR”

Con el objetivo de atender las solicitudes de simulación, este módulo crea un hilo de ejecución para cada LVR a correr. Dicho hilo es el encargado de invocar al módulo responsable de la simulación, habiéndose definido su accionar en función de la instancia de la clase *Configuracion.java* seteada en el aplicativo cliente. Es decir, teniendo en cuenta que dicha instancia contiene la información de los componentes y parámetros del LVR a ejecutar, se redirecciona dicha información al módulo dedicado que manipula el modelo de simulación correspondiente. Existe una clase abstracta llamada *ModuloSimulador.java* de la cual deben heredar los módulos dedicados para garantizar consistencia con la plataforma.

La estructura de hilos planteada permite atender múltiples pedidos de forma concurrente. Si la herramienta de simulación permite trabajar con ejecuciones en paralelo, podrán invocarse múltiples motores desde diferentes hilos. Esto permitirá atender distintas solicitudes en un mismo momento. En caso contrario, el módulo dedicado será quien administre (por medio de la utilización de una cola) las solicitudes pendientes, invocando al motor de ejecución según en turno de cada petición.

Finalizada la ejecución de la corrida, la aplicación servidora devuelve (a la aplicación cliente) en la instancia de *Configuracion.java* los resultados de cada una de las variables medidas durante la simulación (conjunto de instancias de *Resultado.java*).

Módulo 3: “Visualización de Resultados”

Este módulo define el formato de presentación de los resultados obtenidos a lo largo del proceso de simulación. Permite visualizar dicha información bajo dos esquemas: tablas y gráficas.

No existe un conjunto de gráficas predeterminadas, sino que el usuario debe definir sus propias gráficas en términos de dos variables (x e y). De esta manera se flexibiliza el estudio de los resultados, puesto que el operador tiene la capacidad de definir el conjunto de gráficas a estudiar en función de las variables que le sean de interés. Además, el módulo brinda la posibilidad de visualizar en forma tabulada el conjunto de datos asociado a la gráfica. Sin embargo, también permite obtener una única tabla con todos los resultados de simulación (de forma independiente a la generación de gráficas). Esto posibilita al usuario la realización de otro tipo de estudios, en los cuales no es suficiente el trabajo con gráficas en ejes cartesianos.

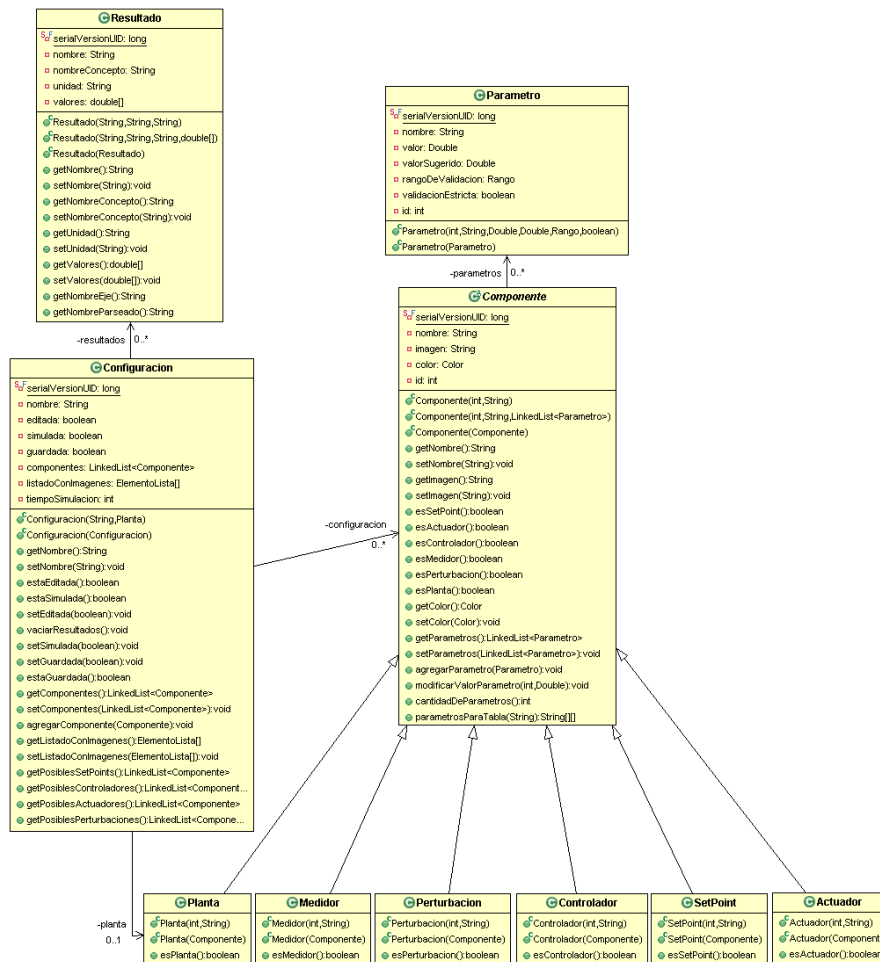


Fig. 4. Diagrama de clases del módulo 1.

2.6 Funcionamiento

En conjunto con los módulos funcionales se desarrollaron interfaces que facilitan la utilización de ambos aplicativos (cliente y servidor).

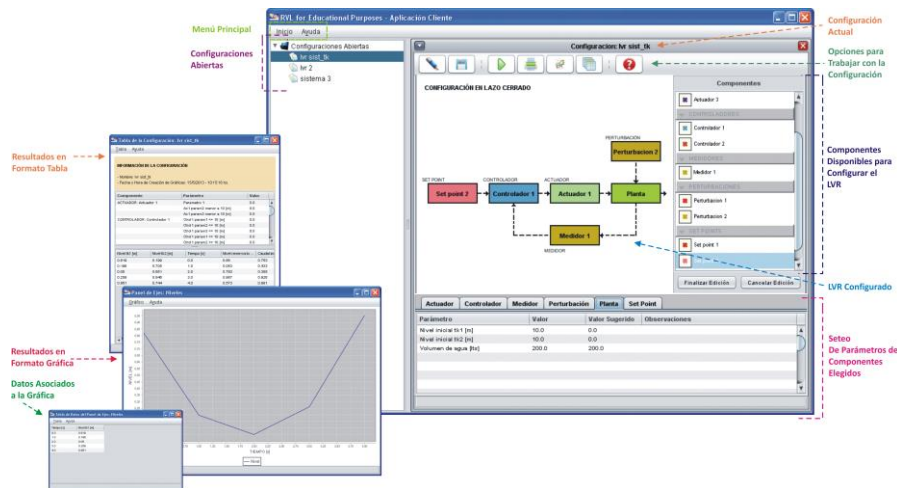


Fig. 5. Interfaz principal (aplicativo cliente).

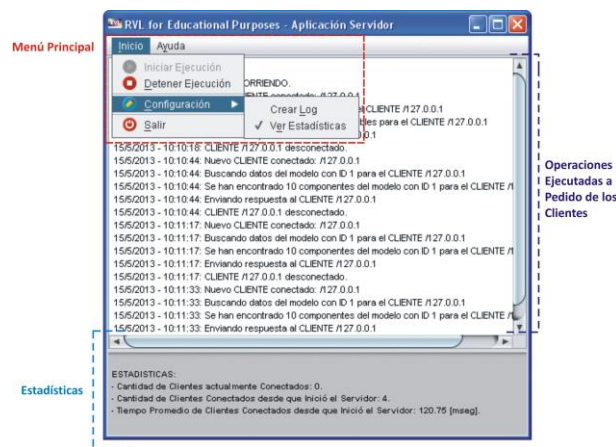


Fig. 6. Aplicativo servidor.

Además se incorporaron funcionalidades extras, entre las cuales se destacan:

1. Para el aplicativo cliente: Almacenamiento de configuraciones, generación de reportes PDF sobre la simulación, exportación de gráficas en formato PNG y almacenamiento de tablas en formato TXT.
2. Para el aplicativo servidor: Almacenamiento de un archivo de log en formato TXT, generación de estadísticas referentes a tiempos de transacción y cantidad de clientes conectados, y detalle de las operaciones ejecutadas a pedido de los clientes.

De esta manera, el acceso a la plataforma por parte de un usuario requiere únicamente de acceso a Internet y de la instalación de la aplicación cliente en su máquina; mientras que, el aplicativo servidor junto con la base de datos se encuentran ubicados en una máquina servidora cuya IP puede obtenerse dinámicamente.

3 Aplicación

El primer LVR incorporado a la plataforma corresponde a un laboratorio real disponible en el Laboratorio de Sistemas de Control de la Facultad Regional Santa Fe - Universidad Tecnológica Nacional. Consiste en un sistema de control de niveles de tanques, que se utiliza habitualmente para prácticas de control realimentado en asignaturas de Control Automático.

3.1 Descripción del Modelo

El modelo utilizado describe el funcionamiento de una planta de dos tanques interconectados, en donde uno de ellos presenta sección transversal no uniforme. Incluye además, el modelado correspondiente a la instrumentación requerida para la construcción del lazo cerrado: sensores de nivel por presión (con simulación de ruido), controlador PID y sistema de alimentación, entre otros. Este último manipula el volumen total de líquido disponible en el sistema (el cual opera en circuito cerrado a partir de un tercer tanque acumulador); vinculándose con los niveles de los tanques principales y la presión de admisión de la bomba centrífuga que impulsa el flujo de alimentación. Un equipo variador de frecuencia de alimentación eléctrica, permite regular la velocidad de operación de la bomba; consiguiendo así, la función de "actuador".

Las actividades referentes al modelado se desarrollaron en un trabajo previo [13], en el cual se analizan las alternativas para la estimación de parámetros desconocidos y se valida el modelo de simulación utilizado en el LVR.

La implementación de este modelo se realizó utilizando Matlab/Simulink^{®1} como herramienta de desarrollo.

3.2 Incorporación del LVR a la Plataforma

La inclusión del LVR correspondiente al sistema de control de niveles se realizó definiendo un módulo dedicado en el aplicativo servidor, el cual interactúa con Matlab/Simulink al momento de ejecutar la simulación. Este módulo es el responsable de garantizar, previo a la realización de la simulación, la validez de los valores seteados en cada uno de los parámetros. Utiliza la librería JMatLink² como intermediaria entre ambas tecnologías (Java y Matlab/Simulink), ya que provee el conjunto de métodos necesarios para la inicialización de variables, ejecución de la simulación, y retorno de los resultados. De esta forma, se simplifica la comunicación entre la plata-

¹ <http://www.mathworks.com/products/simulink/>

² <http://jmatlink.sourceforge.net/>

forma y la herramienta de simulación requerida. Finalizada la simulación, empaqueta el conjunto de valores retornados en instancias de la clase *Resultado.java*; y posteriormente envía esta información al cliente para su visualización.

De esta manera, se observa que la incorporación de un LVR no afecta al aplicativo cliente ya que sólo se requiere modificación en el extremo servidor.

4 Conclusiones

Se ha presentado una plataforma desarrollada en Java que da soporte al trabajo con diferentes LVR en el área de Control Automático. Esta herramienta abstrae al usuario del modelo de simulación subyacente, facilitándole la configuración de los experimentos y permitiéndole extraer sus propias conclusiones de los resultados obtenidos.

El nivel de generalidad logrado evita la necesidad de que el alumno conozca la implementación del modelo de simulación desarrollado, ya que el usuario configura las simulaciones mediante la manipulación de una única interfaz. Además, presenta beneficios frente a la experimentación en laboratorios reales; entre los que se destacan: posibilidad de acceso concurrente para la realización de ensayos en forma simultánea, reproducción de ensayos realizados sobre el laboratorio real sin costo de puesta a punto de la infraestructura, y modalidad de acceso independiente de la presencia física en las instalaciones. De esta manera, la plataforma implementada puede ser utilizada como herramienta educativa, ya que contribuye al estudio de conceptos normalmente vinculados a cursos introductorios de sistemas dinámicos y de control automático; presentando numerosas ventajas tanto para alumnos como para docentes.

Teniendo en cuenta que en la actualidad el control a lazo cerrado es una de las configuraciones más utilizadas, la plataforma también admite su aplicación como simulador de entrenamiento para la sintonía de lazos de control basados en modelos que representen otro tipo de plantas (incluso de interés industrial).

El trabajo con LVR conlleva a un nuevo paradigma y una nueva cultura de la ciencia. No se considera que vayan a suplantarse a los laboratorios tradicionales; sino que deben ser vistos como una extensión de los mismos, como generadores de nuevas perspectivas que no pueden ser exploradas en un laboratorio real a un costo accesible y con un riesgo aceptable. En este contexto, el aplicativo desarrollado constituye un avance tecnológico que complementa la experimentación sobre laboratorios reales.

Referencias

1. Bass, L.; Clements, P.; Kazman, R.: *Software Architecture in Practice*. Addison-Wesley (1998).
2. Boullón Garzón, R.; Rodríguez Laredo, P.: *Comparativas SQL*. Proyecto Final de Carrera, Ing. Téc. en Telecomunicaciones, Esp. en Sistemas Electrónicos, Universidad Politécnica de Catalunya (2006).
3. Calvo, I., Zulueta, E., Gangoiti, U., López, J.: *Laboratorios Remotos y Virtuales en Enseñanzas Técnicas y Científicas*. Ikastorrata, Revista electrónica de Didáctica, Núm. 3 (2008).

4. Domínguez M., Reguera P. Fuertes J.J. Laboratorio Remoto para la Enseñanza de la Automática en la Universidad de León. *Revista Iberoamericana de Automática e Informática Industrial*, Vol. 2, Núm. 2, pp. 36-45 (2005).
5. Dorf, R.; Bishop, R.: *Sistemas de Control Modernos*. Ed. Pearson, Prentice Hall, 10^o Edición (2005).
6. Farias, G.; Esquembre, F.; Dormido, S.: *Análisis, Desarrollo y Publicación de Laboratorios Virtuales y Remotos para la Enseñanza de la Automática*. Congreso Español de Informática: Simposio EIWISA, pp. 1-6 (2005).
7. Farias, G.; Esquembre, F.; Sanchez, J.; Dormido, S.; Vargas, H.; Dormido-Canto, S.; Canto, M.: *Desarrollo de laboratorios virtuales, interactivos y remotos utilizando Easy Java Simulations y Modelos Simulink*. 12th Congreso Latinoamericano de Control Automático, Salvador de Bahía, Brasil (2006).
8. Farias, G.; Dormido, S.; Esquembre, F.; Santos, M.; Dormido-Canto, S.: *Laboratorio virtual de reconocimiento de patrones usando Easy Java Simulations y Matlab*. Actas XXIX Jornadas de Automática (2008).
9. Farias, G.; Esquembre, F.; Sanchez, J.; Dormido, S.; Vargas, H.; Dormido-Canto, S.; Duro, N.: *Laboratorios Virtuales Remotos Usando Easy Java Simulations y Simulink*. Proc. XXVII Jornadas de Automática, pp. 926-933 (2006).
10. Gámez Batistal, Y.; Moreno Vega, V.; Martínez Márquez, Y.: *Consideraciones Generales para el Diseño de una Herramienta Interactiva de Simulación de Procesos*. XIV Congreso Argentino de Ciencias de la Computación (2008).
11. Gómez, A.; Lopez, F.: *Control de un Laboratorio Remoto de Instrumentación Industrial Didáctico por Conexión IP*. Universidad Distrital Francisco José de Caldas (2011).
12. Guzmán, J.; Vargas, H.; Sánchez, J.; Berenguel, M.; Dormido, S.; Rodríguez, F.: *Education research in engineering studies: Interactivity, virtual and remote labs*. Distance Education Research Trends, Nova Science Publisher (2007).
13. Loyarte, A; Diaz, G.; Rosa, J.: *Modelado y Parametrización de una Planta para Control de Niveles*. AADECA 2010, Semana del Control Automático (2010).
14. Morales, M.: *Clasificación del Software Libre Orientado a la Automatización Integral de Bibliotecas según el Nivel de Complejidad de la Biblioteca: Bibliotecas Simples, Bibliotecas de Mediana Complejidad y Bibliotecas de Alta Complejidad*. E-Ciencias de la Información, Revista electrónica semestral, Vol. 3, Núm. 1, Art. 1 (2013).
15. Ogata, K.: *Ingeniería de Control Moderna*. 3^o Edición, Prentice Hall, México (1998).
16. Rosado, L.; Herreros, J. R.: *Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza de la Física*. Recent Research Developments in Learning Technologies, pp. 1-5 (2005).
17. Rosenthal, A.; Martínez, R.; Morillo, D.; Miranda, D.: *PostgreSQL Y SQLServer: Una Comparativa de Rendimiento*. República Bolivariana de Venezuela, Ministerio del Poder Popular para la Educación Universitaria, Colegio Universitario de Los Teques Cecilio Acosta, Vol. 15, pp. 13 (2010).
18. Salcedo-Díaz, R.; Ruiz-Femenia, R.; Gomez-Rico, M.: *An interactive tool for Chemical Engineers students using Easy Java Simulations*. Department of Chemical Engineering, University of Alicante, Spain (2010).
19. Shaw, M.; Garlan, D.: *Software Architecture, Perspectives on an Emerging Discipline*. Prentice-Hal (1996).
20. Surgenor, B.; Firth, K.: *The Role of the Laboratory in Design Engineering Education*. CDEN 2006 - 3rd CDEN/RCCI International Design Conference (2006).
21. Umez-Eronini, E.: *Dinámica de Sistemas y Control*. Thompson Learning, México (2001).