



UNIVERSIDAD
DE PIURA

REPOSITORIO INSTITUCIONAL
PIRHUA

DESARROLLO DE UNA PLATAFORMA WEB BAJO EL FRAMEWORK SPRING DE JAVA PARA LABORATORIOS VIRTUALES

César Belupú-Amaya

Piura, agosto de 2018

FACULTAD DE INGENIERÍA

Máster en Ingeniería Mecánico-Eléctrica con Mención en Automática y
Optimización

Belupú, C. (2018). *Desarrollo de una plataforma web bajo el framework spring de java para laboratorios virtuales* (Tesis de Máster en Ingeniería Mecánico-Eléctrica con Mención en Automática y Optimización). Universidad de Piura. Facultad de Ingeniería. Piura, Perú.



Esta obra está bajo una licencia

[Creative Commons Atribución-NoComercial-SinDerivar 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)

[Repositorio institucional PIRHUA – Universidad de Piura](https://repositorio.institucional.pirhua.edu.pe/)

UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERÍA



**DESARROLLO DE UNA PLATAFORMA WEB BAJO EL FRAMEWORK
SPRING DE JAVA PARA LABORATORIOS VIRTUALES**

**Tesis para optar el Grado de Máster en Ingeniería Mecánico-Eléctrica
con mención en Automática y Optimización**

CÉSAR IVÁN BELUPÚ AMAYA

2018

UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERÍA



**Desarrollo de una plataforma web bajo el framework spring de java para
laboratorios virtuales**

**Tesis para optar el Grado de Máster en Ingeniería Mecánico-Eléctrica
con mención en Automática y Optimización**

César Iván Belupú Amaya

Asesor: Dr. William Ipanaqué Alama

Piura, agosto 2018

A Dios por brindarme la fortaleza de haber podido
llegar hasta aquí

A mis queridos padres, por su amor y confianza, a mi
hermana, Gaby Belupú Amaya, por su incondicional
apoyo

Prólogo

El desarrollo de software computacional es de suma importancia como herramienta para crear métodos de enseñanza, aprendizaje virtual y remoto en diferentes empresas, universidades, centros tecnológicos. Esta tendencia está relacionada con la creciente demanda del aprendizaje en los últimos años, gracias al auge en el desarrollo de las tecnologías de la información y comunicación (TIC).

En el Perú aún no hay una investigación relevante sobre estos temas es por eso que en este trabajo se presenta el desarrollo de una plataforma Web implementada bajo el *framework* de programación Spring de Java que ayude al aprendizaje virtual y remoto en tiempo real.

El uso de estas herramientas tecnológicas logró ser implementada también en la industria para la supervisión de una planta y capacitación del personal, con lo cual ayudaron a ahorrar costo, tiempo, mejorar la gestión y supervisión de sus procesos contribuyendo a aumentar la calidad del producto final.

Con esto se logró cumplir una de las principales motivaciones de este trabajo que era ofrecer una nueva solución al sector tecnológico e industrial que les permita realizar sus operaciones de manera más eficiente, esta investigación abre las puertas a un sinfín de aplicaciones empresariales que se pueden desarrollar con el “*know How*” alcanzado.

Finalmente quisiera agradecer de una manera especial al CONCYTEC (Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica) por su valioso apoyo al brindarme una beca para la realización de esta Maestría en Automática y Optimización, al Dr. Ing. William Ipanaqué Alama por su tiempo y dedicación en la asesoría de esta investigación, así mismo a la Universidad de Piura y al Laboratorio de Sistemas Automáticos de Control que me brindo incondicionalmente todas las facilidades en el desarrollo de esta tesis.

Resumen

Este trabajo está enfocado en el área de sistemas informáticos orientados a la enseñanza, llegando a desarrollar una plataforma Web para poder hacer experimentación de laboratorios a través de internet.

Para desarrollar esta plataforma se requirió aprender el *framework* Spring de Java de esta manera se logró un software robusto y confiable en cuanto a estabilidad; en cuanto al desarrollo de las simulaciones se usó el *toolbox* de Matlab llamado Matlab *Builder JA*, que permite convertir el código de Matlab en Java y así poder correr simulaciones a través de la plataforma Web hecha en Java sin la necesidad de que el usuario final tenga instalado Matlab en su ordenador.

El presente documento describe el proceso que se siguió para desarrollar la plataforma, la investigación bibliográfica y estado del arte de los laboratorios virtuales y remotos, la arquitectura y diseño del sistema, la explicación para realizar las simulaciones de los procesos a través del *toolbox Builder JA* de Matlab, los materiales y métodos utilizados para la programación de la plataforma Web.

Se ha podido experimentar y probar esta plataforma con un grupo de estudiantes en los cursos de Sistemas Automáticos de Control y Control Industrial del programa de Ingeniería Mecánica Eléctrica de la Facultad de Ingeniería de la Universidad de Piura.

El conocimiento alcanzado para esta plataforma no solamente ha servido para fines didácticos o académicos, sino que también ha permitido ser la base para el desarrollo de otras aplicaciones empresariales como una plataforma de gestión y producción en el sector industrial.

Índice general

Introducción	1
Capítulo 1 Antecedentes y estado del arte.....	3
1.1. La ingeniería del control automático y los laboratorios.....	3
1.2. Laboratorios físicos, virtuales y remotos	4
1.3. Diferencias entre un laboratorio virtual y un laboratorio remoto	5
1.4. Problemática.....	6
1.5. Trabajos relevantes acerca de laboratorios virtuales.....	6
1.6. Aporte de la investigación realizada	10
Capítulo 2 Descripción del sistema	13
2.1. Arquitectura del sistema.....	13
2.2. Modelado del sistema para la plataforma virtual	15
2.3. Modelado UML.....	16
2.4. Diagrama de clases de la plataforma virtual	17
2.5. Diagrama de casos de uso	18
2.5.1. Especificaciones de los casos de uso.....	19
2.6. Base de datos de la plataforma Web para laboratorios virtuales	23
2.6.1. Definición de base de datos.....	23
2.6.2. Componentes principales	23
2.6.3. Ventajas del uso de base de datos	23
2.6.4. Diagrama entidad – relación para el sistema de la plataforma virtual	24
2.6.5. Descripción de las tablas de la base de datos diseñada	24
Capítulo 3 Herramientas software utilizadas en la plataforma Web	27
3.1. Matlab <i>Builder</i> JA	27
3.2. Desarrollo de las simulaciones en Matlab.....	28
3.3. Comparación del código convertido a Java y el código Matlab	31
3.4. Introducción al Framework Spring de Java	36
3.4.1. Módulos que conforman Spring	38
3.4.2. Arquitectura de un sistema hecho en Spring	39

3.5.	Características principales del Framework Spring de Java	39
3.5.1.	Inyección de dependencias.....	39
3.5.2.	Contenedor Spring	42
3.5.3.	Uso de los Beans	42
3.5.4.	La inyección de datos a través de constructores	43
3.5.5.	Almacenamiento de listas	45
3.5.6.	Inyección automática con Autowiring	45
3.5.7.	Soporte para acceso a datos	47
3.5.8.	El uso de JDBC Framework.....	48
3.5.9.	El uso Jdbc Template	51
3.5.10.	Soporte para Spring DAO	53
3.5.11.	Manejo de las transacciones.....	54
Capítulo 4	Programación y funcionamiento de la plataforma Web.....	57
4.1.	Materiales y métodos	57
4.1.1.	Compilador Java	57
4.1.2.	Entorno de desarrollo integrado de programación.....	58
4.1.3.	Servidor Web	58
4.1.4.	Base de datos Mysql	58
4.2.	Funcionamiento de la plataforma Web desde un navegador	59
4.2.1.	Interfaz gráfica de entrada.....	59
4.2.2.	Funcionamiento de los roles existentes en la plataforma.....	61
4.2.3.	Ejemplo de laboratorio virtual	72
Capítulo 5	Resultados de la experimentación.....	75
Conclusiones		81
Bibliografía		83
Anexos		87
Anexo A.	Artículo científico presentado en el Congreso Latinoamericano de Control Automático CLCA 2014, Cancún, México	89
Anexo B.	Artículo científico presentado en el Congreso Salesiano de Ciencia, Tecnología e Innovación para la Sociedad, CITIS 2015, Guayaquil, Ecuador	95

Índice de Tablas

Tabla 1.	Ventajas y desventajas de los tipos de laboratorios	5
Tabla 2.	Especificación: Crear usuarios	19
Tabla 3.	Especificación: Crear ciclos	19
Tabla 4.	Especificación: Crear cursos	20
Tabla 5.	Especificación: Crear evaluaciones	20
Tabla 6.	Especificación: Asignación de preguntas a los alumnos	21
Tabla 7.	Especificación: Iniciar Sesión	21
Tabla 8.	Especificación: Realizar Test	22
Tabla 9.	Especificación: Realizar Laboratorio	22
Tabla 10.	Encuesta online	75
Tabla 11.	Resultados de la encuesta	80

Índice de Figuras

Figura 1.	Esquema de la conexión remoto de MATLAB-EJS	7
Figura 2.	Interfaz en Labview del sistema bola y aro	7
Figura 3.	Esquema de comunicación usando Moodle y Blackboard	8
Figura 4.	Sistema SCADA Lookout.....	8
Figura 5.	Esquema de la operación del laboratorio remoto.....	9
Figura 6.	Encuesta aplicada a la pedagogía.....	9
Figura 7	Esquema de trabajo del laboratorio teniendo un PLC para cada persona	10
Figura 8.	Esquema cliente – servidor llevado a la plataforma virtual.....	14
Figura 9.	Esquema de comunicación a través de Internet	15
Figura 10.	Diagrama de clases del sistema para la plataforma virtual	17
Figura 11.	Diagrama de casos de uso para el administrador	18
Figura 12.	Diagrama de casos de uso para el alumno	18
Figura 13.	Diagrama entidad - relación.....	25
Figura 14.	Funcionamiento del toolbox Matlab Builder JA.....	28
Figura 15.	Pasos generales Matlab Builder JA.....	28
Figura 16.	Código .m en Matlab.....	29
Figura 17.	Ventana de comandos Matlab	29
Figura 18.	Ventana Deployment Tool	29
Figura 19.	Ventana New Deployment Project.....	30
Figura 20.	Agregar al proyecto los archivos .m	30
Figura 21.	Compilar proyecto	31
Figura 22.	Paquete jar.....	31
Figura 23.	Framework Spring	37
Figura 24.	Módulos Spring.....	38
Figura 25.	Interface ReportGenerator	40
Figura 26.	Clase HtmlReportGenerator.....	40
Figura 27.	Clase PdfReportGenerator	40
Figura 28.	Clase ReportService.....	40

Figura 29.	Diagrama de clases	41
Figura 30.	Diagrama de clases con inyección.....	41
Figura 31.	Ciclo de vida de los Beans en Spring	42
Figura 32.	Interface ReportGenerator	43
Figura 33.	Clase HtmlReportGenerator	44
Figura 34.	ApplicationContext.xml	44
Figura 35.	Clase Main	44
Figura 36.	Clase HtmlReportGenerator	45
Figura 37.	ApplicationContext.xml	45
Figura 38.	ApplicationContext.xml	46
Figura 39.	applicationContext.xml.....	47
Figura 40.	Patrón DAO	47
Figura 41.	DataSource definido con drive JDBC	49
Figura 42.	DataSource definido por JNDL	50
Figura 43.	DataSource para un pool de Conexiones	50
Figura 44.	Típica aplicación JDBC.....	51
Figura 45.	JDBC Template	52
Figura 46.	JDBC Template	52
Figura 47.	JdbcTemplate.query()	52
Figura 48.	NamedParameterJdbcTemplate	53
Figura 49.	Spring DAO Interface.....	53
Figura 50.	Diagrama de clases implementando una clase padre.....	54
Figura 51.	Class JDBCAppDAO	54
Figura 52.	ApplicationContext.xml	54
Figura 53.	Clase con manejo transaccional.....	55
Figura 54.	Declaración de una transacción JDBC	55
Figura 55.	Declaración de una transacción JTA	56
Figura 56.	Página inicial de la plataforma virtual	60
Figura 57.	Vista de la página inicial en un dispositivo móvil.....	60
Figura 58.	Pantalla de configuración	61
Figura 59.	Configuración de usuario.....	62
Figura 60.	Ventana que lista los usuarios	62
Figura 61.	Formulario para la creación de ciclos	63
Figura 62.	Ventana que lista todos los ciclos creados.....	63

Figura 63.	Formulario para la creación de cursos	63
Figura 64.	Ventana que lista todos los cursos creados	64
Figura 65.	Formulario para la creación de tipos de evaluaciones	64
Figura 66.	Ventana que lista todos tipos de evaluaciones creados.....	64
Figura 67.	Formulario para la creación de nuevos tipos de preguntas	65
Figura 68.	Ventana que lista todos tipos de preguntas	65
Figura 69.	Ventana para la elección del tipo de evaluación.....	66
Figura 70.	Formulario para la creación de la evaluación	66
Figura 71.	Ventana que lista la creación de laboratorios	67
Figura 72.	Ventana para la elección del tipo de pregunta	67
Figura 73.	Ventana para la elección del tipo de pregunta	68
Figura 74.	Ventana para la elección del tipo de pregunta	68
Figura 75.	Ventana para la elección del alumno al que se le va asignar la pregunta.....	69
Figura 76.	Ventana para la elección del tipo de pregunta	69
Figura 77.	Formulario para la asignación de preguntas a los alumnos	70
Figura 78.	Ventana que muestra la asignación de preguntas a los alumnos	70
Figura 79.	Ejemplo de un test de tipo práctico.....	71
Figura 80.	Ejemplo de un test de tipo práctico.....	71
Figura 81.	Gráfica con datos procesados.....	72
Figura 82.	Estados del péndulo invertido a lazo abierto	73
Figura 83.	Salida del péndulo invertido a lazo abierto.....	73
Figura 84.	Estados del sistema a lazo cerrado.....	74
Figura 85.	Salida del sistema a lazo cerrado	74
Figura 86.	Resultados de la pregunta sobre la interfaz de la plataforma Web	76
Figura 87.	Resultados de la pregunta sobre el uso de la plataforma Web.....	77
Figura 88.	Resultados de la pregunta sobre si se disfruto usando la interfaz Web	77
Figura 89.	Resultados de la pregunta sobre si la plataforma es accesible y manejable.....	78
Figura 90.	Resultados de la pregunta sobre si se requiere mucho esfuerzo para llevar a cabo el laboratorio.....	78
Figura 91.	Resultados de la pregunta sobre si los laboratorios virtuales permiten entender mejor el curso	79
Figura 92.	Resultados de la pregunta sobre si los laboratorios virtuales motivan a aprender más sobre el control automático	79

Introducción

Las nuevas tecnologías de comunicación e información (Molina, 2013) han hecho crecer la demanda del aprendizaje virtual, importantes universidades en el extranjero ya han empezado a desarrollar soluciones (Guzmán et al., 2010), sin embargo en el Perú hay muy poco conocimiento y experiencia en software distribuido para supervisión y operación remota, virtual y aplicaciones industriales, de esto se deduce que existen escasas plataformas en tiempo real en el país y consecuentemente en la región Piura para aplicaciones en laboratorios virtuales, remotos y en aplicaciones empresariales.

La mayoría de universidades en el Perú implementan laboratorios convencionales y estos requieren la presencia física de los alumnos y profesores en un mismo lugar, esto se puede convertir en un problema de flexibilidad de horarios, número de salas de cómputo insuficientes, aulas llenas, etc., llevando a los alumnos a no tener un óptimo aprendizaje, de aquí es que se deriva la necesidad de una solución para que los alumnos puedan desarrollar laboratorios virtuales sin la necesidad de estar todos en un mismo sitio y un mismo horario.

Para cubrir esa necesidad es que se planteó una solución Web para que los alumnos universitarios puedan realizar laboratorios de forma virtual, el principal objetivo es dinamizar la captación cognitiva y mejorar las enseñanzas en ciencias de la ingeniería con el desarrollo de esta plataforma para realizar laboratorios virtuales.

Otro punto importante que motivó el estudio de estas nuevas herramientas tecnológicas es que con arquitecturas distribuidas usando herramientas remotas y virtuales se pueden resolver muchos problemas de previsión, optimización de tiempos, en capacitación, operación, seguridad, flexibilidad en una planta (Salman et al., 2013), mejorando la interacción del control y operación de procesos industriales. Es importante que esta tecnología sea asimilada, es decir que el aprendizaje y adaptación sea efectivo para el usuario. Por esto es necesario evaluar no solo técnicamente sino también a nivel pedagógico el aprovechamiento y adaptación de esta tecnología.

Se pueden aprovechar las ventajas que brinda el desarrollo de laboratorios virtuales utilizando herramientas software y hardware de última generación, estas son en dos ámbitos: en el sector educativo, porque permite generar una nueva tecnología para el aprendizaje, y, en el campo industrial, porque usando esta arquitectura se mejoran las condiciones de supervisión, operación y control de los procesos industriales en las plantas.

En cuanto a la metodología aplicada en esta investigación abarcó un aspecto teórico y uno práctico, en cuanto al aspecto teórico se estudiaron los resultados de investigaciones realizadas en otros países sobre laboratorios virtuales implementados como una solución Web.

La investigación bibliográfica se enfocó en la arquitectura de sistemas distribuidos y paralelos, así como también en el estudio del estado del arte en laboratorios virtuales y remotos, así mismo se estudió los lenguajes de programación a utilizar para el desarrollo de la plataforma como son el *framework* Spring, Java *Enterprise Edition* (Java Empresarial) para el desarrollo Web y Matlab para la simulación y control de procesos.

En la parte de diseño se definieron los procesos industriales que se simularon en la plataforma de laboratorios virtuales, también se realizó el diseño del modelo UML (Lenguaje Unificado de Modelado) para la plataforma Web basada en una arquitectura distribuida para entornos virtuales empleando casos de uso y diagrama de clases.

En cuanto a la configuración de hardware se instalaron dos servidores en los ambientes del Laboratorio de Sistemas Automáticos de Control, para el servidor de aplicaciones y para el servidor de base de datos.

Por último, se hicieron pruebas de evaluación con los alumnos de los cursos de Sistemas Automáticos de Control y Control Industrial de la Universidad de Piura, evaluando el nivel de aprendizaje del usuario usando la herramienta implementada frente al que se obtendría con el desarrollo de un laboratorio convencional.

Es importante resaltar que durante los estudios de esta maestría se han realizado dos publicaciones en eventos internacionales, un artículo científico presentado en el Congreso Latinoamericano de Control Automático CLCA 2014, en Cancún, México y otro presentado en el Congreso Salesiano de Ciencia, Tecnología e Innovación para la Sociedad, CITIS 2015, en Guayaquil, Ecuador, durante la maestría el autor también ha formado parte del equipo que desarrolló como producto industrial un sistema de automatización de balanzas para el pesado de uva, aprovechando el conocimiento sobre el *framework* de programación estudiado en esta tesis, sumando esto, a los dos registros de software que posee como propiedad intelectual en Indecopi, como son el “Software para Control de Humedad en Harina De Pescado” con partida registral 0017-2013 y el “Software SCADA para Supervisar Y Monitorear Procesos Industriales” con partida registral 1254-2011.

Capítulo 1

Antecedentes y estado del arte

En este capítulo se explican los conceptos previos, antecedentes y estado del arte de la presente investigación, se hace un resumen de la literatura encontrada en cuanto a laboratorios virtuales, la problemática y una comparación con la propuesta presentada en esta tesis.

1.1. La ingeniería del control automático y los laboratorios

En la Ingeniería del Control Automático, un laboratorio es una parte central en el proceso de aprendizaje convencional. La interacción entre el alumno y un elemento de un laboratorio, como una planta industrial, proporciona un conocimiento que es difícil de asimilar si no es mediante la utilización y la práctica del mismo (Farías, et al., 2016).

Es importante decir también que en el mundo del Control Automático se requiere el estudio constante y profundo de una enorme cantidad de sistemas y modelos de diferentes tecnologías. Es por ello que la simulación, viene a convertirse en un soporte principal para el aprendizaje de los mismos, apoyando de sobremanera a la teoría suministrada en las clases universitarias.

Por otra parte, para lograr desarrollar estas plataformas experimentales se aprovechan los avances en las tecnologías de la información y la comunicación (TIC) (Farías et al., 2006), resaltando el impacto positivo que éstas logran en la enseñanza del control automático, es así que empleando el desarrollo de estas nuevas tecnologías se implementan los laboratorios remotos (Conrado et al., 2012) (Casini et al., 2007) y virtuales (Barrios et al., 2013).

La realización de prácticas experimentales en la automatización, son de suma importancia, ya que beneficia la asimilación de los conceptos teóricos de una mejor manera; estas prácticas permiten resolver futuros problemas de previsión, optimización de tiempos, capacitación, operación, seguridad, flexibilidad en una planta (Barrios et al., 2013).

1.2. Laboratorios físicos, virtuales y remotos

El concepto de laboratorio físico es un concepto claro y tradicional para toda la comunidad docente y científica, sin embargo, en la literatura los conceptos de simulación, instrumento virtual, instrumento remoto, laboratorio remoto y laboratorio virtual no terminan de tener definiciones claras que los distingan.

Como ejemplo, estos términos son explicados de forma distinta en (Candelas et al., 2005) (Calvo et al., 2008) y (Dormido et al., 2007a).

Al ser el objetivo principal de este trabajo presentar una plataforma nueva de experimentación virtual a través de entorno Web es necesario profundizar en las definiciones de los laboratorios y realizar una clasificación de las diferentes formas existentes (Andújar et al., 2010).

- **Instrumento virtual**

Es un sistema que debe ser modelado con todas sus capacidades de proceso, sensores y controles que pueden estar contenidos en uno o más ordenadores, y que permite el acceso local a los recursos ya sean reales o simulados.

- **Instrumento remoto**

Es el instrumento virtual o físico que tiene la capacidad de comunicarse con otros dispositivos a través de la red, permitiendo que los usuarios puedan comunicarse desde cualquier punto con los recursos físicos o simulados.

- **Laboratorio remoto**

Los laboratorios remotos conectan un proceso real a distancia con un operario mediante un interfaz de alta interactividad (Orduña et al., 2012).

Es la plataforma o entorno que está diseñado para realizar el control de un proceso industrial en forma física a distancia, el principal objetivo es el de operar un sistema real, realizar experimentos y acceder a los datos a través de internet para poder analizarlos.

- **Laboratorio virtual**

Un laboratorio virtual es una plataforma que tiene todos los módulos necesarios para realizar experimentos, y que está instalado en la base de una aplicación computacional.

Los laboratorios virtuales interfazan simuladores de procesos con un supervisor emulando una planta real.

En la industria, este concepto se ha utilizado desde hace mucho tiempo para la experimentación de procesos industriales y el conocimiento de los mismos. En el contexto académico, ha surgido la necesidad de apoyar a los alumnos creando sistemas que permitan mejorar el aprendizaje del estudiante en relación a las prácticas de laboratorio, con el objetivo de optimizar el tiempo que éste emplea en la realización de dichas prácticas y la demanda de recursos e infraestructura.

- **Laboratorio virtual y remoto**

Es la plataforma que combina el control de un sistema real y un sistema simulado siempre accesible desde Internet con capacidades de gestión, aprendizaje de contenido y reservas de recursos compartidos.

1.3. Diferencias entre un laboratorio virtual y un laboratorio remoto

Un laboratorio virtual proporciona un entorno simulado; a través de los años se han desarrollado muchos paquetes de software para la simulación de experimentos reales, pero siempre teniendo que instalar un software adicional por cada cliente que requiera usarlo.

Algunas ventajas de estos simuladores son:

- Mejor captación y aprendizaje de los conceptos teóricos.
- Realización de simulaciones y experimentos con una guía detallada.
- Entornos de alta interactividad.
- Algunos softwares son sencillos y fáciles de usar.
- Es una alternativa de bajo costo.
- No existen restricciones en cuanto a instalaciones físicas ni de horario.

En cuanto a los laboratorios remotos el desarrollo ha ido aumentando también debido también a los avances tecnológicos y a las nuevas herramientas que existen para su diseño.

Éstos pueden ofrecer a los estudiantes:

- Una tele-presencia en el laboratorio para ver la evolución del proceso real.
- Realización de experimentos sobre equipos reales.
- Debe tener siempre la ayuda y la colaboración de un guía de laboratorio.
- Aprendizaje a prueba y error.
- Toma de datos experimentales reales.
- Flexibilidad en la elección del lugar para la realización del laboratorio y no tanto del horario.

Tabla 1. Ventajas y desventajas de los tipos de laboratorios

Tipo de laboratorio	Ventajas	Desventajas
Real	<ul style="list-style-type: none"> - Datos experimentales reales - Interacción y conocimiento de un equipo real - Trabajo en equipo - Interacción con el guía del laboratorio. 	<ul style="list-style-type: none"> - Restricciones de tiempo y lugar - Requiere la programación de horarios fijos. - Instalación cara de equipos. - Requiere supervisión
Virtual	<ul style="list-style-type: none"> - Permite entender mejor los conceptos teóricos. - No hay restricciones de tiempo ni lugar - Es un medio interactivo - Es de bajo costo. 	<ul style="list-style-type: none"> - Los datos son idealizados - No hay interacción con un equipo real
Remoto	<ul style="list-style-type: none"> - Interacción con equipo real - Datos experimentales reales - No hay restricciones de tiempo ni lugar - Costo medio. 	<ul style="list-style-type: none"> - Sólo hay presencia virtual en el laboratorio

Fuente: Elaboración propia.

1.4. Problemática

Las nuevas tecnologías de Internet sumado al crecimiento en cuanto a velocidad de los medios de comunicación digitales hacen que el uso de sistemas de software distribuido para el acceso en forma remota a laboratorios virtuales o físicos sea importante para llevar a cabo actividades de aprendizaje a distancia.

Los laboratorios como espacios físicos, son lugares que han posibilitado a los usuarios entender de mejor manera el comportamiento de los sistemas que estudian y más si se trata del Control Automático. Sin embargo, como se ha mencionado antes gracias a los adelantos en materia de nuevas tecnologías de la información, surgen otras opciones que pueden ayudar o reemplazar a un laboratorio tradicional es por eso que el desarrollo de estas nuevas plataformas remotas y virtuales son temas de actualidad en la Ingeniería de la Automatización (Barrios et al., 2013).

La sociedad actual necesita sistemas de enseñanza más flexibles, interactivos, accesibles debido a las limitaciones de tiempo, espacio, altos costos. Es por eso que este tipo de sistemas distribuidos ha evolucionado rápidamente.

Actualmente la problemática de la organización de las prácticas dentro de los estudios de ingeniería se centra en los horarios fijos, necesidad de personal estable y organización del espacio físico. En muchos casos no ha sido posible una buena organización, lo que suele repercutir negativamente en el alumno generando su frustración y llevándolo a dejar de usar los equipos del laboratorio.

En el laboratorio de Sistemas Automáticos de Control se propuso como objetivo descentralizar los laboratorios y que los horarios no sean un freno para los alumnos. Teniendo en cuenta también que el futuro contexto educativo, nos muestra a un alumno que tendrá mucha más libertad para organizar sus horarios, la enseñanza será menos rígida en cuanto a tiempos establecidos para los laboratorios y por lo tanto la organización estos deberán modificarse pudiéndose convertir en algo más complicado.

1.5. Trabajos relevantes acerca de laboratorios virtuales

Son varios los ejemplos de laboratorios virtuales y remotos que se muestran en la actualidad, tenemos a (Fábregas et al., 2011) que utiliza EJS (Easy Java Simulation) (Esquembre, 2004) y Simulink para controlar un sistema Ball and Hoop, usando el software adicional JIM server (Farías et al., 2006), que es un software libre que permite ejecutar la simulación de forma remota (Farías, 2010) (Figura 1).

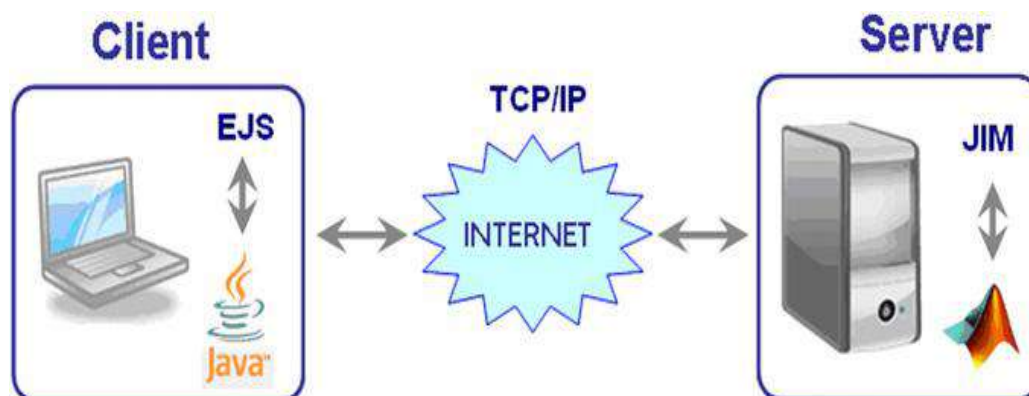


Figura 1. Esquema de la conexión remota de MATLAB-EJS
Fuente: (Fábregas et al., 2011)

En (Calvo et al., 2009) usan la arquitectura cliente – servidor (Jin, et al., 2004); Labview se utiliza en el lado servidor para adquirir los datos, mientras que la tecnología OPC (Mahmoud, et al., 2015) se utiliza para conectar el servidor remoto con el lado cliente donde está el operador remoto (interacción hombre-máquina), en esta investigación controlan un sistema bola y aro (Figura 2).

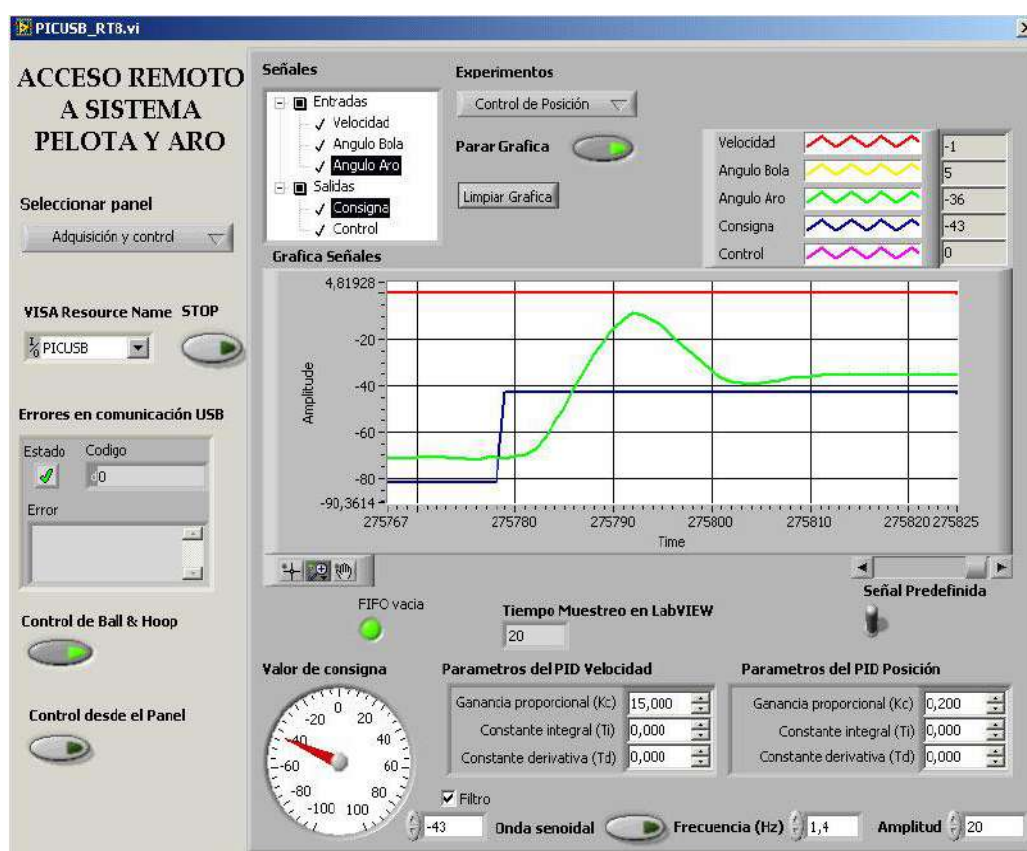


Figura 2. Interfaz en Labview del sistema bola y aro
Fuente: (Fábregas et al., 2009)

En (Barrios et al., 2013) han realizado un proyecto usando Labview (Luna-Moreno, et al., 2015), conectándose al EJS a través del software JIL Server (Java Internet Labview). Usa una arquitectura multiusuario que permite a las aplicaciones del lado del cliente como Moodle (Caputi et al., 2015) y Blackboard (Ahmed et al., 2015) para acceder a los experimentos remotos (Figura 3).

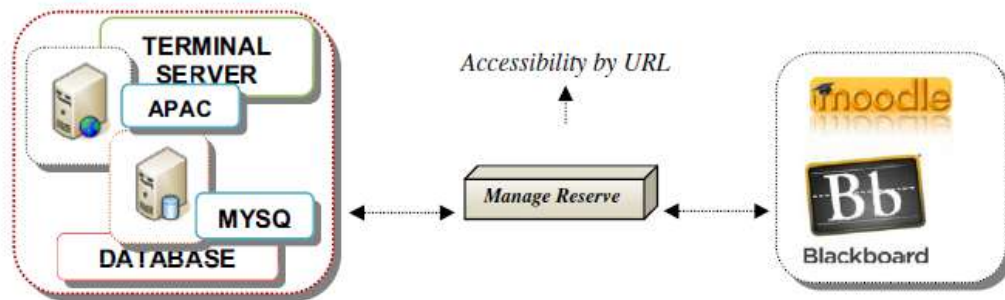


Figura 3. Esquema de comunicación usando Moodle y Blackboard
Fuente: (Barrios et al., 2013)

(Lazar et al., 2008) presentó un avance para la creación de sistemas de control en red utilizando el software Lookout SCADA (Figura 4). Se basa en aplicaciones cliente - servidor que operan plantas piloto reales vía Internet.

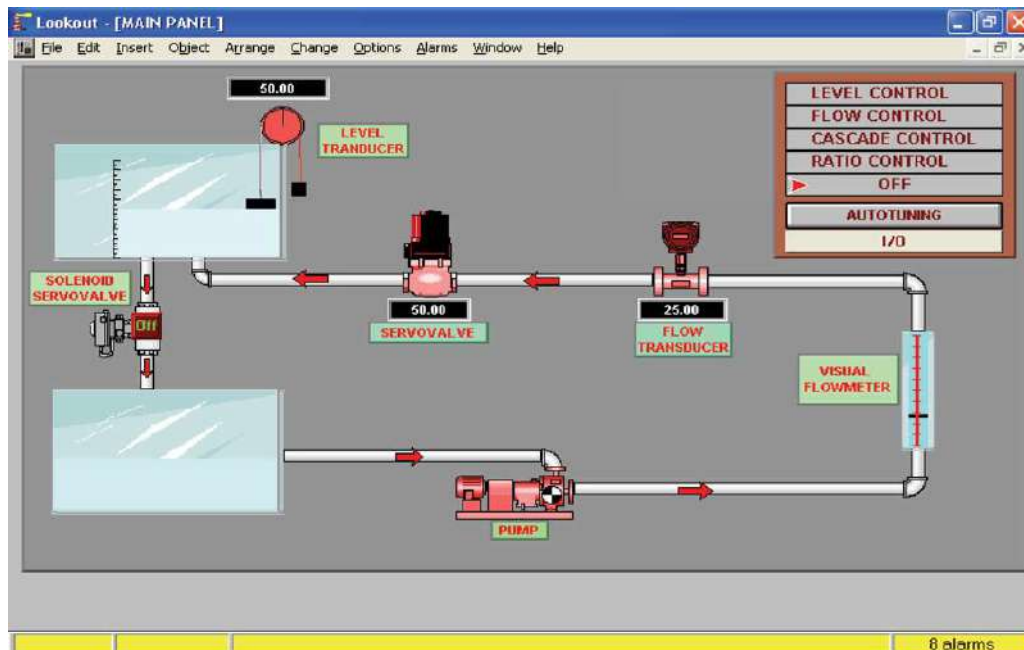


Figura 4. Sistema SCADA Lookout
Fuente: (Lazar et al., 2008)

En (Hercog et al., 2007) han realizado laboratorios remotos utilizando Matlab, Labview Virtual Instrument (VI) y DSP (Figura 5). Este experimento se compone de sistemas de control DSP-2, una PC, un servidor para páginas Web y un sistema de reserva. Al mismo tiempo, el servidor de Labview se ejecuta en la PC para permitir el control remoto.

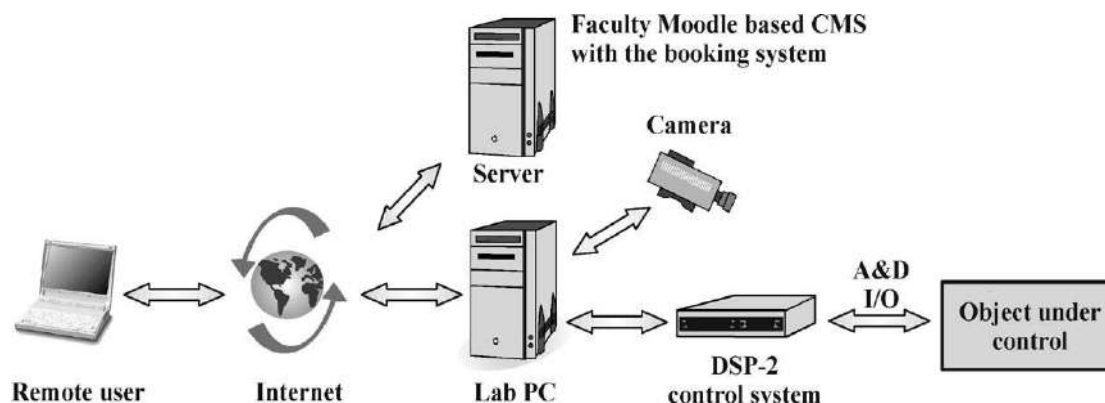


Figura 5. Esquema de la operación del laboratorio remoto

Fuente: (Hercog et al., 2007)

(Gardel et al., 2010) explica que una mejor enseñanza se daría si cada persona cuenta con su propio PLC en la realización de laboratorios, sin embargo, el llevarlo a cabo tendría un costo elevado, por lo que plantean un laboratorio remoto como solución a su problema.

(Sivakumar et al., 2005) centra su investigación en ayudar en el campo de la educación, no sólo demostrando como controlar un proceso, sino analizando las ventajas, desventajas, el uso e impacto en las personas (Figura 6), ayudando de esta manera en la pedagogía.

	On a scale of 1 to 5 rate: (1=Very poor, 2 = Poor, 3= Satisfactory, 4 = Good, 5= Very Good)
Q1	Was the INWK laboratory equipment easy to use ?
Q2	Was the level of interaction with laboratory components adequate?
Q3	How was the response time of laboratory components?
Q4	Could the router/switches and other networking gear be remotely accessed on entering the userID/password)?
Q5	Was the operation of router/switches and other networking gear reliable?
Q6	Did the response from router/switches and other networking gear help you better understand networking concepts and theories?
Q7	Was the feedback from the laboratory instructor useful?
Q8	Were laboratory handouts and extra online information useful?
Q9	Was the online wiring diagram information (cabling between networking gear) useful?
Q10	What was the level of "hands on" feeling experienced when configuring/troubleshooting networks with equipment in Internetworking laboratories?
Q11	Was the networking equipment used in the INWK laboratory similar to the equipment used in a real-world, networking environment?
Q12	Were the laboratory components / networking gear in the INWK laboratory (i.e., the router/switches and other networking gear) "state-of-the-art"?

Figura 6. Encuesta aplicada a la pedagogía

Fuente: (Sivakumar et al., 2005)

Para el desarrollo de la plataforma Web se eligió el *framework* Spring de Java que permite crear aplicaciones Web empresariales robustas, estables y confiables. De nuestro conocimiento, no se ha usado la combinación de estas herramientas como alternativa a los laboratorios virtuales.

Capítulo 2

Descripción del sistema

En este capítulo se explica el diseño y arquitectura del sistema propuesto para esta investigación. Se hará una breve explicación del propósito del sistema, la plataforma planteada, las partes que componen al sistema, el diseño UML, el diseño de base de datos, así como las diversas etapas que desarrolla cada componente para procesar los datos requeridos.

2.1. Arquitectura del sistema

Un sistema informático necesita de un *framework* de comunicación siguiendo un lenguaje en común, poder enviar y recibir datos entre los elementos; esta estructura es conocida como la arquitectura de un sistema.

La arquitectura de un sistema se define también como la suma de elementos computacionales que siguen diferentes patrones con el fin de unir distintos servicios informáticos; estos elementos se comunican por medio de mensajes de solicitud y mensajes de entrega para conseguir el intercambio de la información (Xiaocong, 2015).

Existe más de un estilo arquitectónico: cliente-servidor (Gardel et al., 2010), organización o modelos por capas, filtros, sistemas jerárquicos por niveles, etc.

En este trabajo se ha usado la arquitectura cliente-servidor, en la Figura 8 y Figura 9 se puede apreciar la comunicación que se da entre dos elementos: uno o varios clientes (usuarios finales); que pueden ser computadoras de propósito general que determinan los requerimientos de información que se solicitan; y uno o varios servidores que son computadoras con mayores prestaciones (procesador, memoria, disco duro, etc.), pues se encargan de la lógica del sistema, además deben tener la capacidad de dar respuesta a varios clientes, procesando las peticiones recibidas.

Algunas características que complementan y resumen la noción de arquitectura cliente – servidor (Gardel et al., 2010):

- El cliente interactúa con el usuario (interfaz) y el servidor lo hace con recursos compartidos (comunicación directa con el proceso).

- Posibilidad de compartir recursos lógicos y físicos: varios clientes hacen uso de un solo servidor.
- El cliente es emisor cuando solicita y el servidor es el receptor, siendo la función del último de carácter pasivo, pues solo espera las solicitudes del cliente. Inmediatamente, cambian de posición, el servidor envía lo requerido (emisor) y el cliente lo recibe (receptor).
- Las prestaciones necesarias de una computadora son diferentes si actuará como cliente o como servidor.
- Funciones determinadas van dirigidas a cada uno con el fin de lograr un mejor aprovechamiento del ancho de banda de red.
- Se aplica el concepto de escalabilidad horizontal (adicionar clientes) y vertical (mejorar características del servidor o adiciona más servidores).
- Permite que sistemas distintos puedan compartir una fácil integración.
- Diferentes procesos pueden llevarse a cabo en una misma computadora o en varias que se encuentran distribuidas geográficamente a lo largo de una misma red.
- Exigen una verificación tanto en el cliente como en el servidor para evitar riesgos en la seguridad del esquema cliente - servidor.

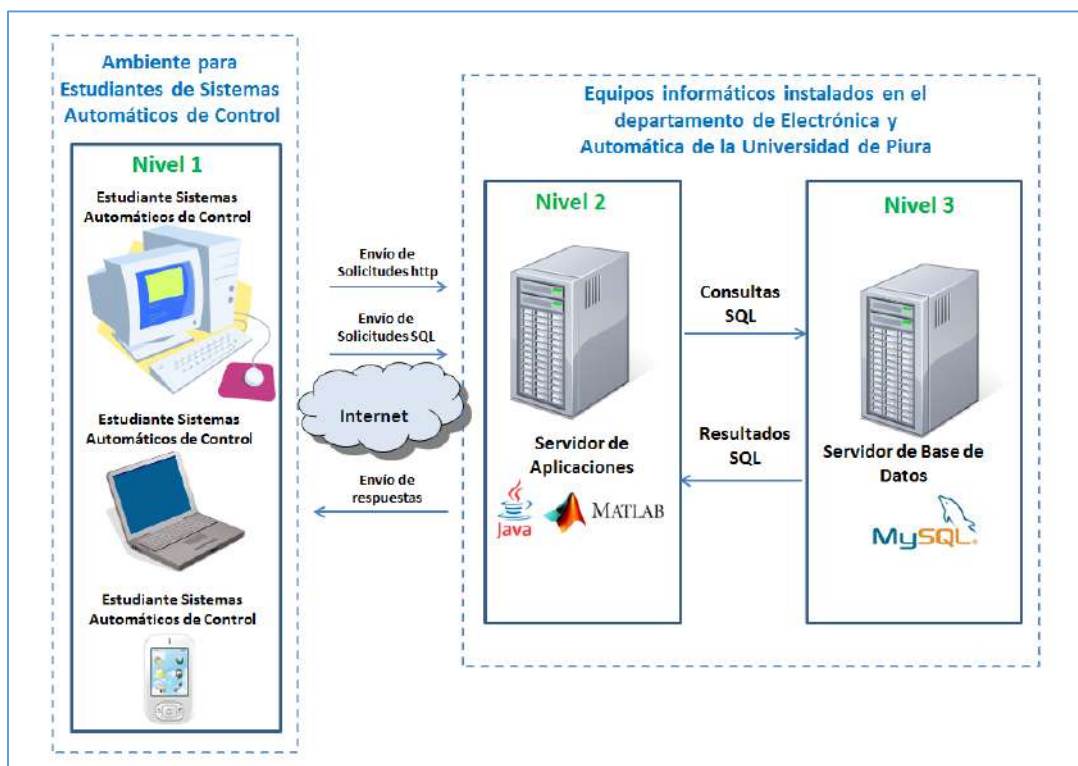


Figura 8. Esquema cliente – servidor llevado a la plataforma virtual
Fuente: Elaboración propia.

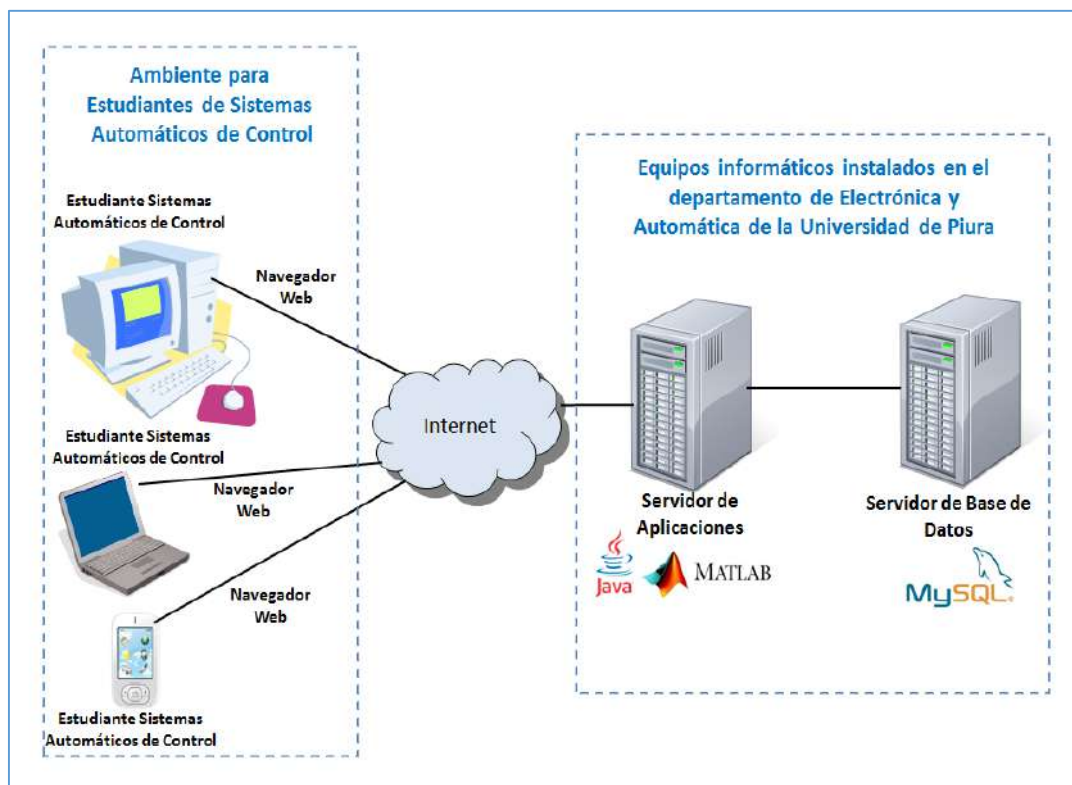


Figura 9. Esquema de comunicación a través de Internet

Fuente: Elaboración propia.

2.2. Modelado del sistema para la plataforma virtual

Los sistemas informáticos actualmente llegan a ser muy complejos, tanto que aquellos de mayores requerimientos son comparables en dificultad y tamaño con grandes obras de otras ramas de la ingeniería.

Esta complejidad tiene dos aspectos fundamentales. Por un lado, es medianamente difícil desarrollar un sistema sofisticado si no se tiene alguna experiencia previa ni la información clara y básica del total de los requerimientos de la aplicación. Por otro lado, también es difícil establecer a priori si el sistema funcionará de manera correcta una vez terminado, lo que es especialmente grave en aquellas aplicaciones cuyo coste es muy elevado, o son especialmente difíciles de modificar, o llevan a cabo tareas muy delicadas o peligrosas.

Es para eso que es necesario el uso de modelos que permitan un análisis previo de los requerimientos, características y funcionamiento del sistema.

Un buen modelo debe tener varias características (Selic, 2003):

- Debe permitir modificar la abstracción, para poder obviar detalles irrelevantes si se quiere para el análisis de ciertas propiedades concretas del sistema o aumentar el grado para analizar con mayores detalles si es necesario.

- Se deben usar notaciones claras, para que cualquier lector pueda entender el sistema. Si la notación es difícil de entender el modelo no será de mucha utilidad, incluso para sistemas con un mínimo de complejidad.
- Debe mostrar todos los requerimientos que va tener la aplicación final.

2.3. Modelado UML

UML, (Unified Modeling Language) (Booch, et al., 1999), surge en los años noventa como fusión de tres métodos importantes para el desarrollo de software orientados a objetos, el método de Booch (Booch, 1994), el método OOSE (Jacobson, et al., 1992) y el método OMT (Rumbaugh, et al., 1991).

Cada método era utilizado para una fase del desarrollo y se quiso generar un único método que fuera usado durante toda la etapa del desarrollo y eliminara los problemas de contar con diferentes nomenclaturas.

UML es un conjunto de lenguajes, no es una metodología, su objetivo es especificar, documentar los elementos que integran una aplicación software. Los lenguajes definidos en UML son gráficos, para facilitar el estudio y comprensión de los diferentes elementos.

A continuación, se definen los diagramas más importantes en UML:

- Diagramas de clases: Es un diagrama que muestra las clases involucradas en el sistema, es un diagrama estático.
- Diagramas de objetos: Relacionado con el diagrama de clases, muestra una visión de las instancias reales de las clases que se ejecutan en la aplicación en un momento determinado.
- Diagramas de casos de uso: Muestra los actores y sus relaciones en el sistema.
- Diagramas de secuencia: Muestra la parte dinámica del sistema, en especial la interacción entre un subconjunto de objetos, básicamente a través del envío de mensajes entre ellos.
- Diagramas de estados: Determinan el funcionamiento de los objetos de una clase.
- Diagramas de actividad: Son un tipo especial de diagrama de estados que resaltan el flujo de actividad entre los objetos.
- Diagramas de componentes. Son diagramas estáticos que muestran la organización y las dependencias entre los componentes. Un componente suele corresponder a varias clases o interfaces.

Para esta plataforma virtual se ha hecho el modelado en dos de los diagramas más importantes como son el diagrama de clases y los casos de uso.

Es un sistema diseñado para que se puedan realizar no sólo laboratorios virtuales, sino también practicas virtuales y exámenes, el objetivo es dejar el sistema abierto, es decir que no sólo lo usen los cursos de Sistemas Automáticos de Control y Control de Industrial sino cualquier otro curso de Ingeniería.

2.4. Diagrama de clases de la plataforma virtual

En la Figura 10 se muestra el diagrama de clases para el sistema de la plataforma virtual.

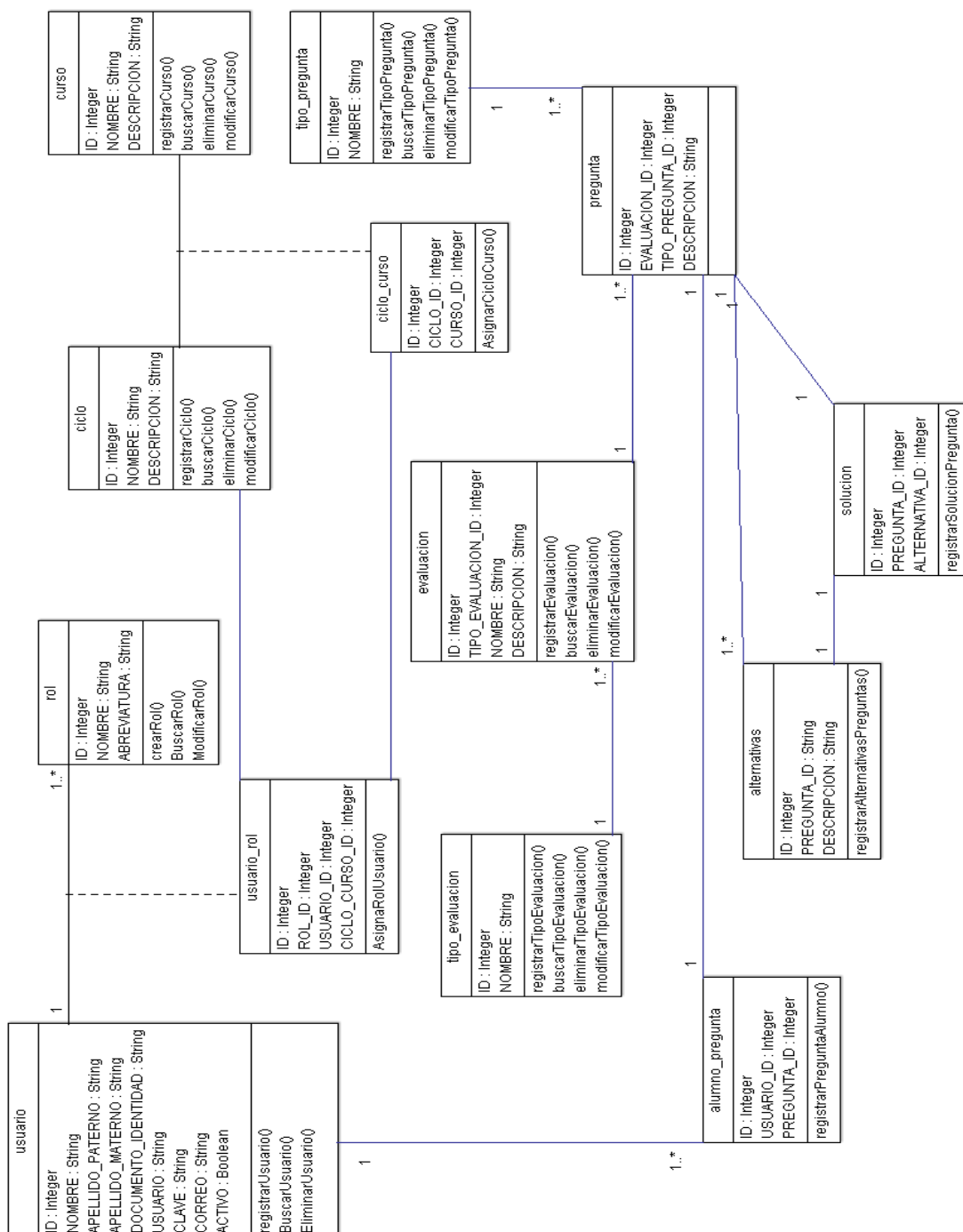


Figura 10. Diagrama de clases del sistema para la plataforma virtual
Fuente: Elaboración propia.

2.5. Diagrama de casos de uso

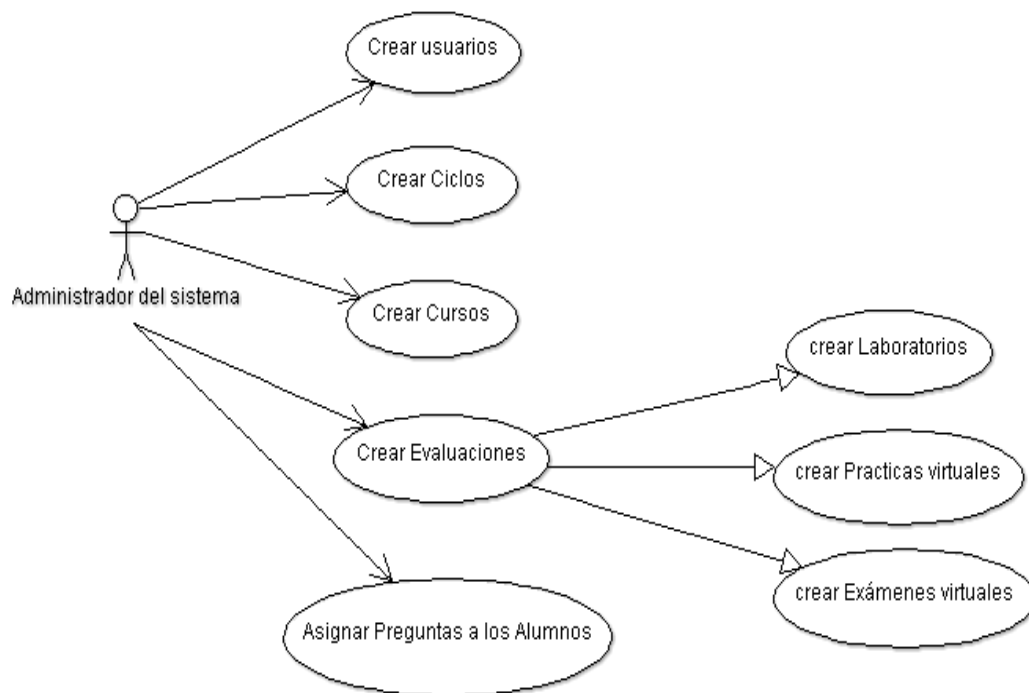


Figura 11. Diagrama de casos de uso para el administrador
Fuente: Elaboración propia.

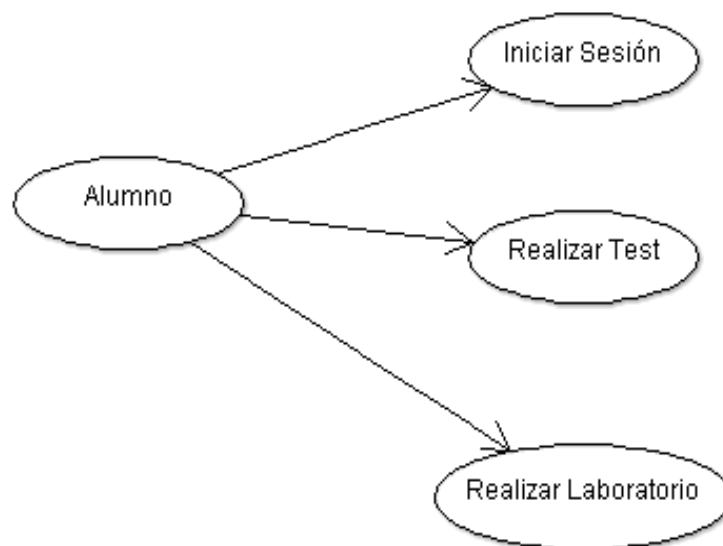


Figura 12. Diagrama de casos de uso para el alumno
Fuente: Elaboración propia.

2.5.1. Especificaciones de los casos de uso

A continuación, se describe un resumen de los casos de uso para el sistema de la plataforma virtual.

Casos de uso para el administrador

Tabla 2. Especificación: Crear usuarios

	CREAR USUARIOS										
Fuentes	Administrador										
Objetivos Asociados	Este caso de uso tiene la finalidad de registrar datos de los usuarios (nombres y apellidos, DNI, dirección), así mismo registrar que rol tiene en el sistema y a que ciclo y curso pertenece.										
Descripción	Este caso de uso empieza cuando se inicia sesión para el administrador, cada ciclo debe crear los usuarios que van a usar el sistema.										
Precondición	Inicio de sesión										
Secuencia Normal	<table> <tr> <td><u>PASO</u></td><td><u>ACCION</u></td></tr> <tr> <td>1</td><td>Inicio sesión.</td></tr> <tr> <td>2</td><td>Ingresar datos de los usuarios.</td></tr> <tr> <td>3</td><td>Se registra el usuario.</td></tr> <tr> <td>4</td><td>Actualiza base de datos.</td></tr> </table>	<u>PASO</u>	<u>ACCION</u>	1	Inicio sesión.	2	Ingresar datos de los usuarios.	3	Se registra el usuario.	4	Actualiza base de datos.
<u>PASO</u>	<u>ACCION</u>										
1	Inicio sesión.										
2	Ingresar datos de los usuarios.										
3	Se registra el usuario.										
4	Actualiza base de datos.										
Postcondición	Usuario nuevo activo										

Fuente: Elaboración propia.

Tabla 3. Especificación: Crear ciclos

	CREAR CICLOS										
Fuentes	Administrador										
Objetivos Asociados	Este caso de uso tiene la finalidad de registrar datos de los ciclos en los que se van a realizar las evaluaciones.										
Descripción	Este caso de uso empieza cuando se inicia sesión para el administrador, que debe crear el ciclo donde se van a tomar las evaluaciones.										
Precondición	Inicio de sesión										
Secuencia Normal	<table> <tr> <td><u>PASO</u></td><td><u>ACCION</u></td></tr> <tr> <td>1</td><td>Inicio sesión.</td></tr> <tr> <td>2</td><td>Ingresar datos del ciclo</td></tr> <tr> <td>3</td><td>Se registra el ciclo.</td></tr> <tr> <td>4</td><td>Actualiza base de datos.</td></tr> </table>	<u>PASO</u>	<u>ACCION</u>	1	Inicio sesión.	2	Ingresar datos del ciclo	3	Se registra el ciclo.	4	Actualiza base de datos.
<u>PASO</u>	<u>ACCION</u>										
1	Inicio sesión.										
2	Ingresar datos del ciclo										
3	Se registra el ciclo.										
4	Actualiza base de datos.										
Postcondición	Ciclo nuevo activo										

Fuente: Elaboración propia.

Tabla 4. Especificación: Crear cursos

	CREAR CURSOS										
Fuentes	Administrador										
Objetivos Asociados	Este caso de uso tiene la finalidad de registrar datos de los cursos en los que se van a realizar las evaluaciones.										
Descripción	Este caso de uso empieza cuando se inicia sesión para el administrador, que debe crear el curso donde se van a tomar las evaluaciones.										
Precondición	Inicio de sesión										
Secuencia Normal	<table> <tr> <td><u>PASO</u></td><td><u>ACCION</u></td></tr> <tr> <td>1</td><td>Inicio sesión.</td></tr> <tr> <td>2</td><td>Ingresar datos del curso</td></tr> <tr> <td>3</td><td>Se registra el curso.</td></tr> <tr> <td>4</td><td>Actualiza base de datos.</td></tr> </table>	<u>PASO</u>	<u>ACCION</u>	1	Inicio sesión.	2	Ingresar datos del curso	3	Se registra el curso.	4	Actualiza base de datos.
<u>PASO</u>	<u>ACCION</u>										
1	Inicio sesión.										
2	Ingresar datos del curso										
3	Se registra el curso.										
4	Actualiza base de datos.										
Postcondición	curso nuevo activo										

Fuente: Elaboración propia.

Tabla 5. Especificación: Crear evaluaciones

	CREAR EVALUACIONES										
Fuentes	Administrador										
Objetivos Asociados	Este caso de uso tiene la finalidad de registrar datos de las evaluaciones que se van a tomar en determinado ciclo y determinado curso.										
Descripción	Este caso de uso empieza cuando se inicia sesión para el administrador, él es el único facultado para crear las evaluaciones. Los tipos de evaluaciones a crear son: laboratorios, prácticas virtuales, exámenes virtuales.										
Precondición	Inicio de sesión										
Secuencia Normal	<table> <tr> <td><u>PASO</u></td><td><u>ACCION</u></td></tr> <tr> <td>1</td><td>Inicio sesión.</td></tr> <tr> <td>2</td><td>Ingresar datos de las evaluaciones</td></tr> <tr> <td>3</td><td>Se registra la evaluación</td></tr> <tr> <td>4</td><td>Actualiza base de datos.</td></tr> </table>	<u>PASO</u>	<u>ACCION</u>	1	Inicio sesión.	2	Ingresar datos de las evaluaciones	3	Se registra la evaluación	4	Actualiza base de datos.
<u>PASO</u>	<u>ACCION</u>										
1	Inicio sesión.										
2	Ingresar datos de las evaluaciones										
3	Se registra la evaluación										
4	Actualiza base de datos.										
Postcondición	evaluación nueva activo										

Fuente: Elaboración propia.

Tabla 6. Especificación: Asignación de preguntas a los alumnos

	ASIGNACIÓN DE PREGUNTAS A LOS ALUMNOS										
Fuentes	Administrador										
Objetivos Asociados	Este caso de uso tiene la finalidad de asignar las preguntas de determinada evaluación a los alumnos inscritos en determinado curso.										
Descripción	Este caso de uso empieza cuando se inicia sesión para el administrador, él es el único facultado para asignar las preguntas a los alumnos.										
Precondición	Inicio de sesión, alumnos registrados										
Secuencia Normal	<table> <tr> <td><u>PASO</u></td><td><u>ACCION</u></td></tr> <tr> <td>1</td><td>Inicio sesión.</td></tr> <tr> <td>2</td><td>Asignar preguntas a los alumnos.</td></tr> <tr> <td>3</td><td>Se registra las preguntas</td></tr> <tr> <td>4</td><td>Actualiza base de datos.</td></tr> </table>	<u>PASO</u>	<u>ACCION</u>	1	Inicio sesión.	2	Asignar preguntas a los alumnos.	3	Se registra las preguntas	4	Actualiza base de datos.
<u>PASO</u>	<u>ACCION</u>										
1	Inicio sesión.										
2	Asignar preguntas a los alumnos.										
3	Se registra las preguntas										
4	Actualiza base de datos.										
Postcondición	Evaluación nueva para que el alumno pueda desarrollar.										

Fuente: Elaboración propia.

Casos de uso para el Alumno

Tabla 7. Especificación: Iniciar Sesión

	INICIAR SESIÓN								
Fuentes	Alumno								
Objetivos Asociados	Este caso de uso tiene la finalidad de comprobar el usuario y password del alumno.								
Descripción	Este caso de uso empieza se carga la interfaz gráfica para el alumno, ayuda en la validación del usuario y password ingresados.								
Precondición	Interfaz cargada								
Secuencia Normal	<table> <tr> <td><u>PASO</u></td><td><u>ACCION</u></td></tr> <tr> <td>1</td><td>Cargar Interfaz gráfica</td></tr> <tr> <td>2</td><td>Ingresar datos</td></tr> <tr> <td>3</td><td>Validar los datos ingresados</td></tr> </table>	<u>PASO</u>	<u>ACCION</u>	1	Cargar Interfaz gráfica	2	Ingresar datos	3	Validar los datos ingresados
<u>PASO</u>	<u>ACCION</u>								
1	Cargar Interfaz gráfica								
2	Ingresar datos								
3	Validar los datos ingresados								
Postcondición	Inicio de sesión para que el alumno pueda realizar su test virtual								

Fuente: Elaboración propia.

Tabla 8. Especificación: Realizar Test

	REALIZAR TEST										
Fuentes	Alumno										
Objetivos Asociados	Este caso de uso tiene la finalidad mostrar el test creado al alumno en determinada evaluación.										
Descripción	Este caso de uso empieza cuando se inicia sesión para el alumno, después de iniciar sesión se le mostrará la interfaz gráfica del test.										
Precondición	Inicio de sesión, alumnos registrados										
Secuencia Normal	<table> <tr> <td><u>PASO</u></td><td><u>ACCION</u></td></tr> <tr> <td>1</td><td>Inicio sesión.</td></tr> <tr> <td>2</td><td>Cargar Interfaz del test</td></tr> <tr> <td>3</td><td>Se registra las respuestas</td></tr> <tr> <td>4</td><td>Actualiza base de datos.</td></tr> </table>	<u>PASO</u>	<u>ACCION</u>	1	Inicio sesión.	2	Cargar Interfaz del test	3	Se registra las respuestas	4	Actualiza base de datos.
<u>PASO</u>	<u>ACCION</u>										
1	Inicio sesión.										
2	Cargar Interfaz del test										
3	Se registra las respuestas										
4	Actualiza base de datos.										
Postcondición	Interfaz de la evaluación a desarrollar										

Fuente: Elaboración propia.

Tabla 9. Especificación: Realizar Laboratorio

	REALIZAR LABORATORIO										
Fuentes	Alumno										
Objetivos Asociados	Este caso de uso tiene la finalidad mostrar el laboratorio creado para desarrollar.										
Descripción	Este caso de uso empieza el alumno pasa el test de conocimientos, al aprobar dicho test deberá seguir a la interfaz del laboratorio implementado.										
Precondición	Inicio de sesión, alumnos registrados, test aprobado										
Secuencia Normal	<table> <tr> <td><u>PASO</u></td><td><u>ACCION</u></td></tr> <tr> <td>1</td><td>Test aprobado</td></tr> <tr> <td>2</td><td>Cargar Interfaz del laboratorio</td></tr> <tr> <td>3</td><td>Se registra las simulaciones</td></tr> <tr> <td>4</td><td>Mostrar gráficas de las simulaciones</td></tr> </table>	<u>PASO</u>	<u>ACCION</u>	1	Test aprobado	2	Cargar Interfaz del laboratorio	3	Se registra las simulaciones	4	Mostrar gráficas de las simulaciones
<u>PASO</u>	<u>ACCION</u>										
1	Test aprobado										
2	Cargar Interfaz del laboratorio										
3	Se registra las simulaciones										
4	Mostrar gráficas de las simulaciones										
Postcondición	Interfaz del laboratorio desarrollar										

Fuente: Elaboración propia.

2.6. Base de datos de la plataforma Web para laboratorios virtuales

2.6.1. Definición de base de datos

Una base de datos es un conjunto de datos organizados y conexos entre sí, los cuales son recolectados y utilizados por los sistemas de información.

Las bases de datos brindan los componentes necesarios a las aplicaciones para apoyar en la toma de decisiones. Un software empresarial explota la información contenida en las bases de datos y eso permite lograr ventajas considerables. Es por eso que es importante conocer el funcionamiento y estructura de una base de datos para saber manejarla.

2.6.2. Componentes principales

- Los datos o registros: son la base de datos en sí.
- El hardware: Son los dispositivos de almacenamiento donde está instalada la base de datos, también se pueden nombrar todos los dispositivos periféricos que se necesitan para su uso.
- Software: Está formado por un conjunto de programas que manejan la base de datos. Este software gestiona todas las solicitudes que llegan a la base de datos por parte de los diferentes usuarios.
- Existen tres clases de usuarios que interactúan con una base de datos:
 - El programador de aplicaciones, que es quien crea el software que va interactuar con la base de datos.
 - El usuario final, es el usuario que utiliza el programa desarrollado y accede a la base de datos por medio de un lenguaje de consulta que en la mayoría de las veces es el lenguaje SQL.
 - El administrador de la base de datos que es quien se encarga del control general de la plataforma de base de datos.

2.6.3. Ventajas del uso de base de datos

- Permite globalizar la información ya que diferentes usuarios pueden considerarla como un recurso colectivo de la empresa careciendo de dueños concretos.
- Permite compartir información ya que múltiples usuarios y sistemas pueden utilizar la misma base de datos para cruzar información que necesiten.
- Permite mantener la integridad en la información ya que se debe almacenar la información correcta sin duplicidad.
- Para un buen modelo de base de datos es importante la independencia de los mismos. Esto implica un que los registros sean abstractos al programa o software que interactúa con él; es decir, que se puedan hacer cambios en la información sin hacer cambios en el software desarrollado.

2.6.4. Diagrama entidad – relación para el sistema de la plataforma virtual

La base de datos para la plataforma virtual está conformada por las siguientes tablas (Figura 13):

- “usuario”
- “rol”
- “usuario_rol”
- “tipo_evaluacion”
- “evaluación”
- “alternativas”
- “banco_preguntas”
- “curso”
- “ciclo”
- “ciclo_curso”
- “curso”
- “tipo_pregunta”
- “solución”
- “pregunta”
- “alumno_pregunta”

2.6.5. Descripción de las tablas de la base de datos diseñada

- “Tabla Rol”: En esta tabla se definen todos los roles del sistema. El sistema tiene cargado por defecto los roles de administrador, alumnos, profesor.
- “Tabla usuario”: En esta tabla se guardan los datos de todos los usuarios del sistema ya sean alumnos, administradores o profesores.
- “Tabla usuario_rol”: Es una tabla de enlace, determina la asignación de un determinado rol con un determinado usuario.
- “Tabla ciclo”: En esta tabla se guardan los datos del ciclo donde se van a tomar las evaluaciones.
- “Tabla curso”: Guarda el curso donde se toma la evaluación.
- “Tabla ciclo_curso”: Es una tabla de enlace, determina la asignación de un determinado curso con un determinado ciclo.



Figura 13. Diagrama entidad - relación

Fuente: Elaboración propia.

- “Tabla tipo_evaluacion”: En esta tabla se guarda el tipo de evaluación, el sistema tiene cargado por defecto los siguientes tipos de evaluación: laboratorios, prácticas, exámenes.

- “Tabla evaluación”: En esta tabla se guarda la descripción de la evaluación, por ejemplo: nombre, día, hora.
- “Tabla tipo_pregunta”: En esta tabla se guarda los tipos de pregunta que puede haber en una evaluación, el sistema tiene cargado por defecto los siguientes tipos de pregunta: teóricas, prácticas, con alternativas.
- “Tabla pregunta”: En esta tabla se guarda las preguntas registradas para determinada evaluación.
- “Tabla banco_preguntas”: En esta tabla se guarda un histórico de las preguntas registradas en el sistema.
- “Tabla alternativas”: En esta tabla se guardan las alternativas de una determinada pregunta si esta es de tipo con alternativas.
- “Tabla solución”: En esta tabla se guarda la alternativa correcta de una pregunta de tipo alternativa.
- “Tabla alumno_pregunta”: En esta tabla se guarda la asignación de determinadas preguntas a determinados alumnos por curso y ciclo.

Capítulo 3

Herramientas software utilizadas en la plataforma Web

En este capítulo se presentan las herramientas software usadas para el desarrollo en sí de la plataforma Web para laboratorios virtuales. Se empieza por describir el *toolbox* Matlab Builder JA que sirve para convertir las simulaciones hechas en Matlab a Java, también se explica los pasos necesarios para realizar las simulaciones con el uso de este *toolbox*, por último se hace una descripción de la otra herramienta utilizada para la plataforma como es el *framework* Spring de Java, los aspectos teóricos y técnicos, presentando algunos ejemplos para entender su funcionamiento y características.

3.1. Matlab Builder JA

Matlab *Builder* JA es una herramienta que permite crear clases Java a partir del código escrito en Matlab (Perutka, et al., 2015). Estas clases de Java son integradas en las aplicaciones de Java a través del compilador *Matlab Compiler Runtime* (MCR), desplegándose gratuitamente en las computadoras de escritorio o servidores Web que no necesariamente tienen Matlab instalado.

Matlab *Builder* JA crea componentes desplegables que hacen cálculos basados en Matlab, visualizaciones e interfaces gráficas accesibles a los usuarios finales de los programas Java, estas clases son portables (Figura 14).

Una de las principales características de Matlab *Builder* JA es que proporciona una sólida conversión de datos, conservando la flexibilidad de Matlab cuando se llama desde código Java. Esto es posible con la clase *MWArray* implementada en Matlab, esta clase permite convertir matrices nativas en Matlab a Java y viceversa; es así como se logra la conversión automática de los datos.

Cuando el programa Java se implementa en la red, varios usuarios pueden acceder a él a través de un navegador Web.

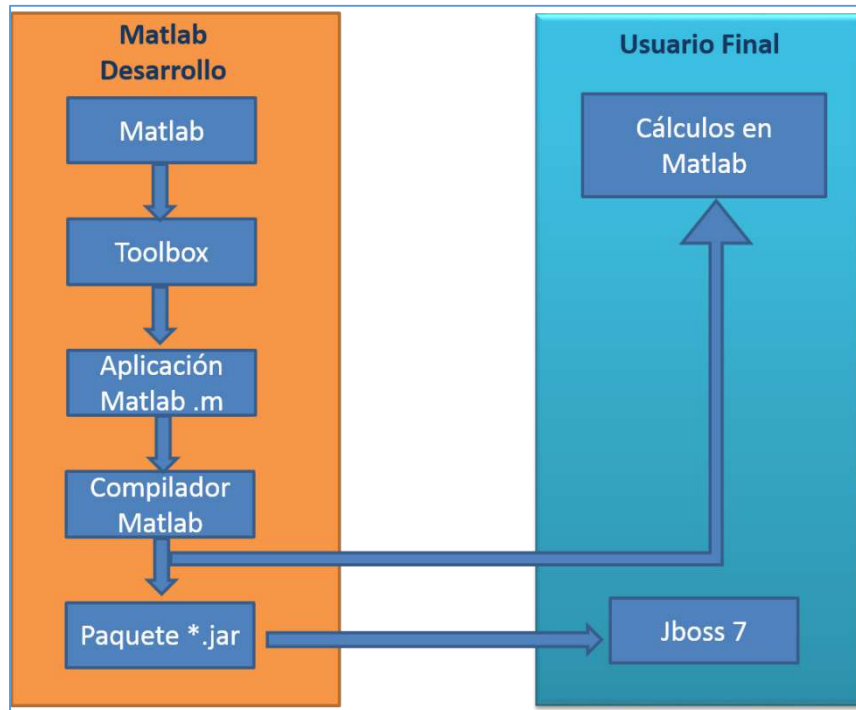


Figura 14. Funcionamiento del toolbox Matlab Builder JA
Fuente: Elaboración propia.

En este caso el programa Java se ha implementado en un servidor Web para que la aplicación pueda ser accesible desde cualquier punto remoto.

3.2. Desarrollo de las simulaciones en Matlab

Los códigos para implementar los procesos para simulación se deben desarrollar en código .m de Matlab, después se debe crear el archivo independiente de java (archivo .jar) con el *toolbox* Matlab Builder JA siguiendo los siguientes pasos generales (Figura 15):



Figura 15. Pasos generales Matlab Builder JA
Fuente: Elaboración propia.

A continuación se muestra un ejemplo de la creación de un paquete java a partir de código Matlab para su aplicación en el sistema Web para laboratorios virtuales.

- Escribir el ejercicio del laboratorio en código .m (Figura 16).

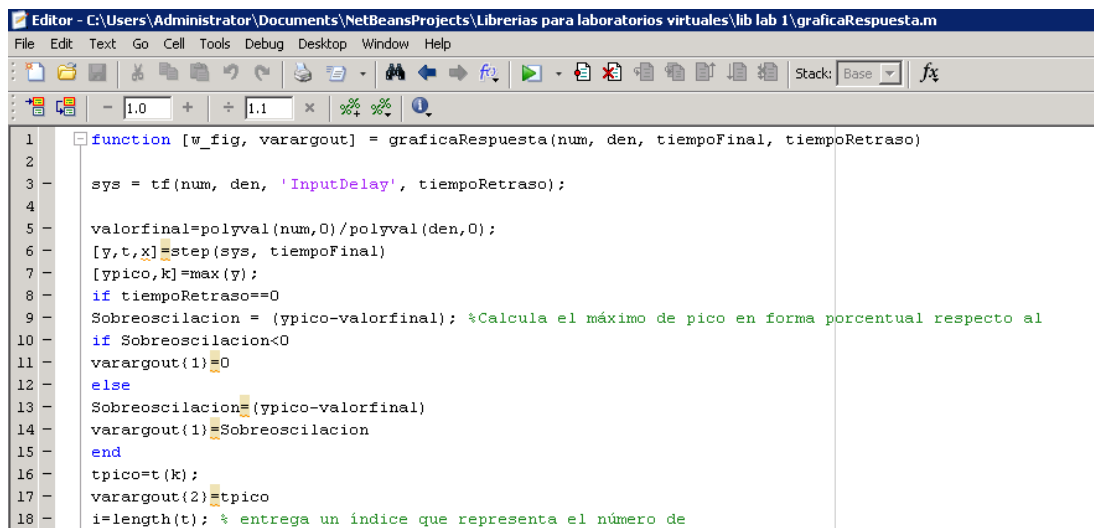


Figura 16. Código .m en Matlab

Fuente: Elaboración propia

- Escribir el comando "deploytool" en la ventana de comandos de Matlab (Figura 17) para abrir el toolbox Matlab Builder JA.

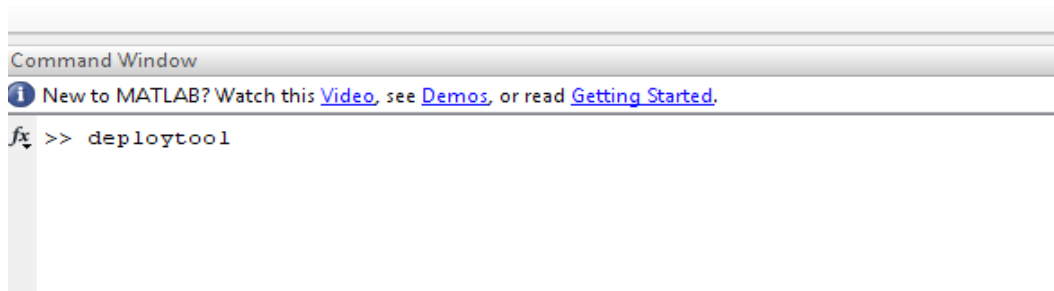


Figura 17. Ventana de comandos Matlab

Fuente: Elaboración propia

- Se abrirá la ventana Deployment Tool (Figura 18)

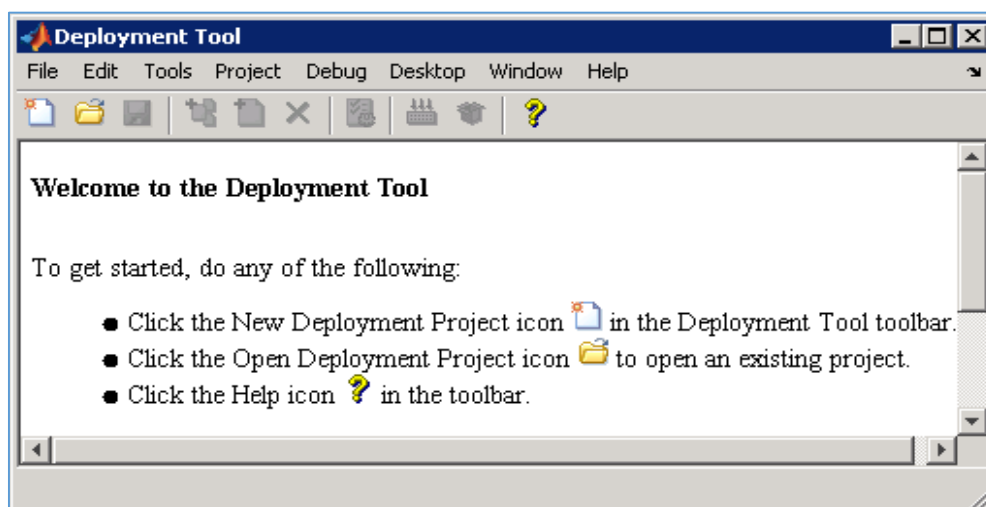


Figura 18. Ventana Deployment Tool

Fuente: Elaboración propia

- Se procede a crear un nuevo proyecto: File - New Deployment Project (Figura 19).

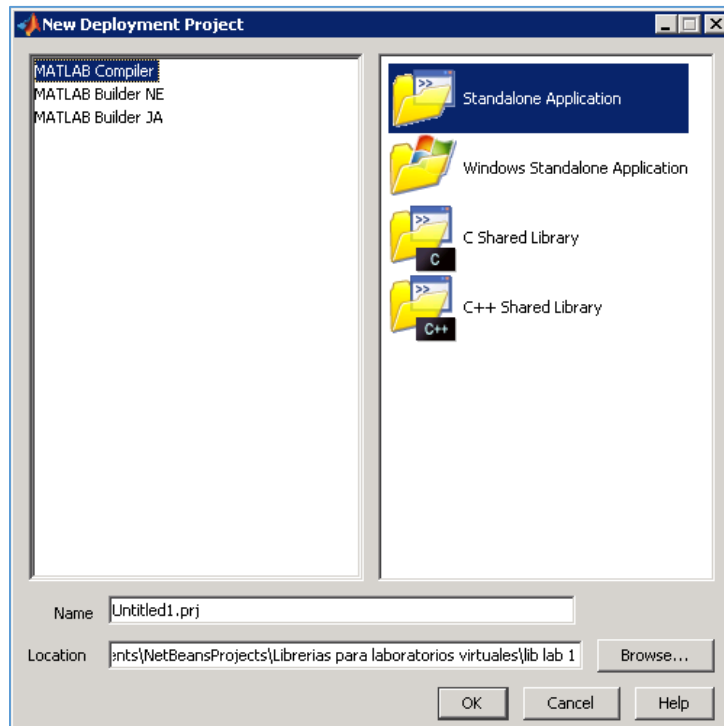


Figura 19. Ventana New Deployment Project
Fuente: Elaboración propia

- Se elige la opción Matlab Builder JA y se le asigna un nombre al proyecto (Figura 19).
- El siguiente paso es asignar los archivos .m a nuestro proyecto con la opción Add File (Figura 20).

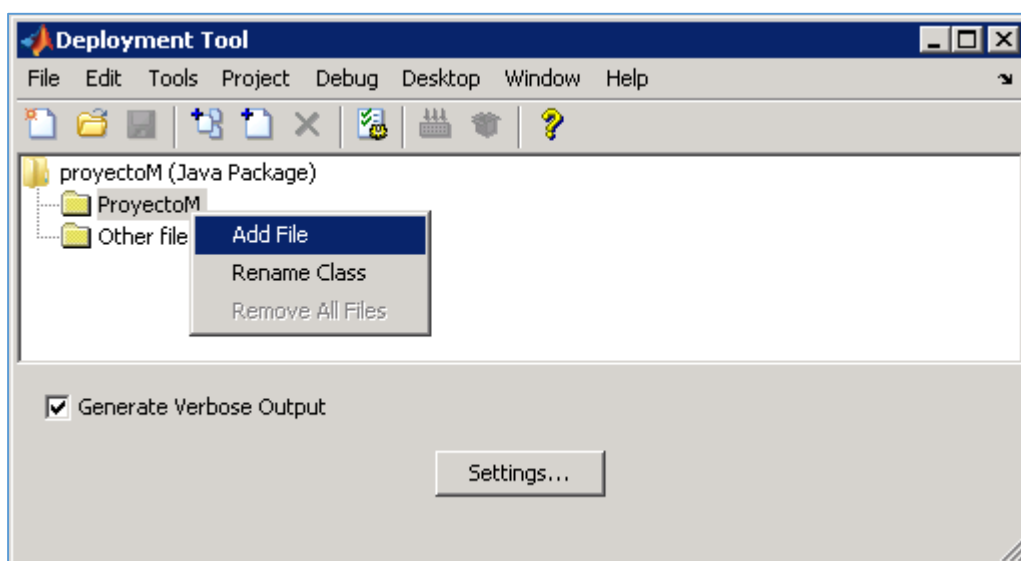


Figura 20. Agregar al proyecto los archivos .m
Fuente: Elaboración propia

- Una vez asignados los archivos .m sólo queda compilar el proyecto con la opción Build the Project (Figura 21).

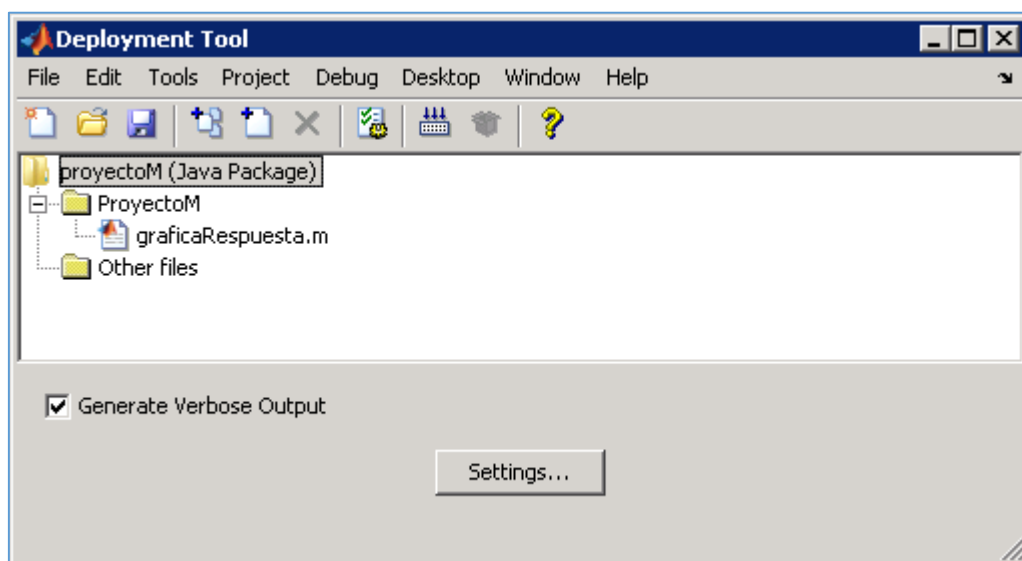


Figura 21. Compilar proyecto
Fuente: Elaboración propia

- Al finalizar ya tenemos el código .m de matlab encapsulado en un paquete .jar para poder usarlo en la programación Web en Java (Figura 22).

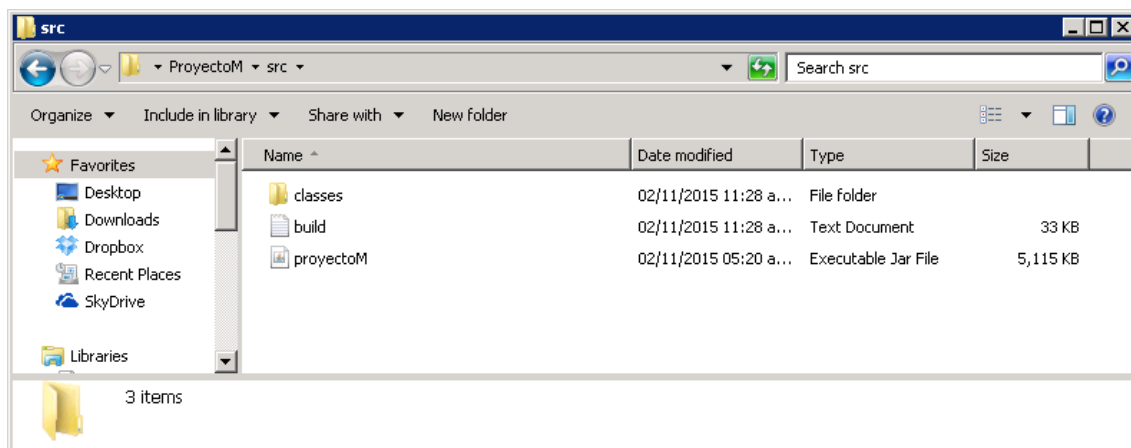


Figura 22. Paquete jar
Fuente: Elaboración propia

3.3. Comparación del código convertido a Java y el código Matlab

Como se ha explicado en párrafos anteriores, para usar el *toolbox* Matlab *Builder* JA se necesita desarrollar el laboratorio en código .m como se muestra en el ejemplo a continuación:

```

function [w_fig, varargout] = graficaRespuesta(num, den, tiempoFinal,
tiempoRetraso)

sys = tf(num, den, 'InputDelay', tiempoRetraso);

valorfinal=polyval(num,0)/polyval(den,0);
[y,t,x]=step(sys, tiempoFinal)
[ypico,k]=max(y);
if tiempoRetraso==0
Sobreoscilacion = (ypico-valorfinal); %Calcula el máximo de pico en forma
porcentual respecto al
if Sobreoscilacion<0
varargout{1}=0
else
Sobreoscilacion=(ypico-valorfinal)
varargout{1}=Sobreoscilacion
end
tpico=t(k);
varargout{2}=tpico
i=length(t); % entrega un índice que representa el número de

while(y(i)>0.98*valorfinal)&(y(i)<1.02*valorfinal)
i=i-1;
end
Tiempo_de_establecimiento = t(i);
varargout{3}= Tiempo_de_establecimiento
n=1;
while(y(n)<0.10*valorfinal)
n=n+1;
end
m=1;
while(y(m)<0.9*valorfinal)
m=m+1;
end
Tiempo_de_subida=t(m)-t(n);
varargout{4}=Tiempo_de_subida
k1=1;
while(y(k1)<0.5*valorfinal)
k1=k1+1;
end
Tiempo_de_retardo=t(k1-1)
varargout{5}=Tiempo_de_retardo
end
if tiempoRetraso>0
Sobreoscilacion=(ypico-valorfinal); %Calcula el máximo de pico en forma
porcentual respecto al
if Sobreoscilacion<0
varargout{1}=0
else
Sobreoscilacion = (ypico-valorfinal);
varargout{1}=Sobreoscilacion
end
tpico=t(k)+tiempoRetraso;
varargout{2}=tpico
i=length(t); % entrega un índice que representa el número de

```

```

while (y(i)>0.98*valorfinal) & (y(i)<1.02*valorfinal)
i=i-1;
end
Tiempo_de_establecimiento= t(i)+tiempoRetraso;
varargout{3}=Tiempo_de_establecimiento
n=1;
while (y(n)<0.10*valorfinal)
n=n+1;
end
m=1;
while (y(m)<0.9*valorfinal)
m=m+1;
end
Tiempo_de_subida=t(m)-t(n);
varargout{4}=Tiempo_de_subida
k1=1;
while (y(k1)<0.5*valorfinal)
k1=k1+1;
end
Tiempo_de_retardo=t(k1-1)+tiempoRetraso;
varargout{5}=Tiempo_de_retardo
end
h_fig = figure('visible','off', 'Menubar', 'none', ...
'PaperPositionMode','auto', 'Numbertitle', 'off', ...
'Name', 'VarArg Example');
set(h_fig,'Visible','off');
h_plot = plot(t,y);
grid on;
w_fig = webfigure(h_fig);
close(h_fig);
end

```

Al finalizar la conversión se obtiene una clase Java como la que se muestra a continuación:

```

// Decompiled by DJ v3.12.12.100 Copyright 2015 Atanas Neshkov Date: // Home
Page: http://www.neshkov.com/dj.html - Check often for new version!
// Decompiler options: packimports(3)
// Source File Name: ClaseLab2.java

```

```

package paqueteLab2;

```

```

import com.mathworks.toolbox.javabuilder.*;
import com.mathworks.toolbox.javabuilder.internal.*;
import java.util.*;

```

```

// Referenced classes of package paqueteLab2:
//     PaqueteLab2MCRFactory

```

```

public class ClaseLab2 extends MWComponentInstance
{

```

```

private ClaseLab2(MWMCR mwmcr)
    throws MWException
{
    super(mwmcr);
    synchronized(paqueteLab2/ClaseLab2)
    {
        sInstances.add(this);
    }
}

public ClaseLab2()
    throws MWException
{
    this(PaqueteLab2MCRFactory.newInstance());
}

private static MWComponentOptions getPathToComponentOptions(String s)
{
    MWComponentOptions mwcomponentoptions = new
MWComponentOptions(new Object[] {
    new MWCtfExtractLocation(s), new MWCtfDirectorySource(s)
});
    return mwcomponentoptions;
}

/**
 * @deprecated Method ClaseLab2 is deprecated
 */

public ClaseLab2(String s)
    throws MWException
{
    this(PaqueteLab2MCRFactory.newInstance(getPathToComponentOptions(s)));
}

public ClaseLab2(MWComponentOptions mwcomponentoptions)
    throws MWException
{
    this(PaqueteLab2MCRFactory.newInstance(mwcomponentoptions));
}

public void dispose()
{
    super.dispose();
}

```

```

    synchronized(paqueteLab2/ClaseLab2)
    {
        sInstances.remove(this);
    }
    break MISSING_BLOCK_LABEL_67;
    Exception exception1;
    exception1;
    synchronized(paqueteLab2/ClaseLab2)
    {
        sInstances.remove(this);
    }
    throw exception1;
}

public static void main(String args[])
{
    try
    {
        MWMCR mwmcr = PaqueteLab2MCRFactory.newInstance();
        mwmcr.runMain(sGraficaRespuestaSignature, args);
        mwmcr.dispose();
    }
    catch(Throwable throwable)
    {
        throwable.printStackTrace();
    }
}

public static void disposeAllInstances()
{
    synchronized(paqueteLab2/ClaseLab2)
    {
        Disposable disposable;
        for(Iterator iterator = sInstances.iterator(); iterator.hasNext();
disposable.dispose())
            disposable = (Disposable)iterator.next();

        sInstances.clear();
    }
}

public void graficaRespuesta(List list, List list1)
    throws MWException
{
    fMCR.invoke(list, list1, sGraficaRespuestaSignature);
}

```

```

    public void graficaRespuesta(Object aobj[], Object aobj1[])
        throws MWException
    {
        fMCR.invoke(Arrays.asList(aobj), Arrays.asList(aobj1),
sGraficaRespuestaSignature);
    }

    public transient Object[] graficaRespuesta(int i, Object aobj[])
        throws MWException
    {
        Object aobj1[] = new Object[i];
        fMCR.invoke(Arrays.asList(aobj1), MWMCR.getRhsCompat(aobj,
sGraficaRespuestaSignature), sGraficaRespuestaSignature);
        return aobj1;
    }

    private static final Set sInstances = new HashSet();
    private static final MWFunctionSignature sGraficaRespuestaSignature = new
MWFunctionSignature(2, true, "graficaRespuesta", 4, false);

}

```

La función `graficaRespuesta` está definida primero en Matlab, después de la obtención del paquete `.jar` se “convierte” en una función Java para que ya pueda ser utilizada en cualquier programa Java pasándole por parámetro la lista de valores que se necesitan para realizar la simulación.

3.4. Introducción al Framework Spring de Java

Este apartado está basado en la investigación de (Arauco, 2012) que ha permitido entender de forma teórica el marco de trabajo Spring Empresarial.

Spring es un marco de trabajo (*framework*) (Zschaler et al., 2014) utilizado para el desarrollo de aplicaciones *open source*, escritas en lenguaje de programación Java. Fue creado por Rod Johnson y Jürgen Höller que aprovecharon su experiencia en lenguajes de programación para basar este framework en el patrón MVC (Modelo Vista Controlador) (Pop et al., 2014).

Este patrón de programación tiene por finalidad facilitar la construcción de aplicaciones empresariales, eso con Spring es posible gracias a la utilización de sencillos *Java Beans* (Praehofer, et al., 2001) dejando de lado la utilización de los EJBs (*Enterprise Java Beans*) que hasta antes de la creación de Spring se venían usando.

El dejar de lado los EJBs es un enfoque por el cual Spring ha ganado mucha popularidad ya que simplifica el desarrollo de aplicaciones J2EE. (Walls, 2011).

(Walls, 2011) explica que anteriormente los *Enterprise Java Beans* (EJBs) permitían resolver problemas complejos. Y muchas veces se tenía la obligación de utilizar EJB, aunque el proyecto no tuviera un gran grado de complejidad, como una solución a esta dificultad apareció Spring, donde la complejidad de la aplicación es proporcional a la complejidad del problema que se requiere resolver. Sin embargo, esto no le resta crédito a EJB, ya que sigue ofreciendo a los desarrolladores servicios muy útiles para resolver tareas un tanto complicadas, la diferencia con Spring está en que este *framework* de trabajo brinda los mismos servicios, pero simplificando aún más el desarrollo en cuanto a la programación.

A continuación, se detalla brevemente las características más resaltantes de Spring (Arauco, 2012).

- Ligero: En cuanto a tamaño y memoria Spring es un marco de trabajo pequeño y rápido. Las librerías de Spring pueden darse en un archivo de tamaño ligero aproximado de 2.5 mb.
- No intrusivo: Las aplicaciones desarrolladas en Spring a través de capas (Figura 23) no contienen dependencias hacia clases del marco de trabajo, lo que permite mantener la independencia de las mismas.
- Inyección de dependencias: Spring promueve el desacoplamiento entre clases conocido como inyección de dependencias.
- Contenedor: Spring es un contenedor automático que se encarga del ciclo de vida y configuración de los objetos utilizados por las aplicaciones.

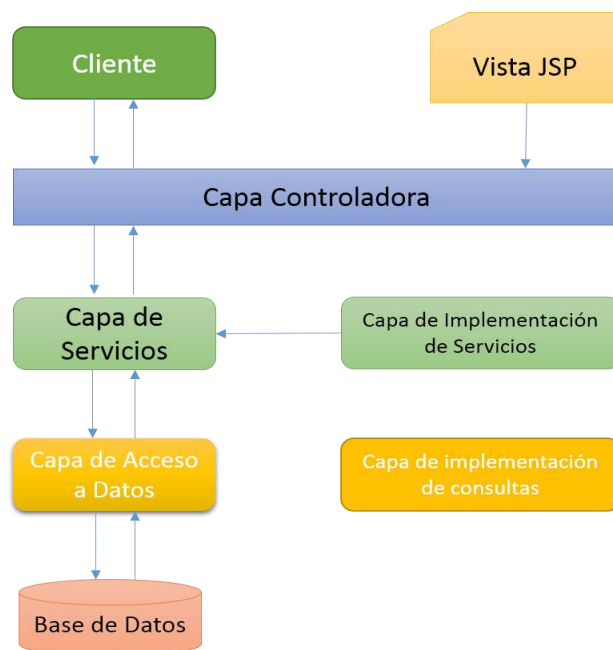


Figura 23. Framework Spring
Fuente: Elaboración propia

3.4.1. Módulos que conforman Spring

EL *framework* Spring está compuesto por una serie de módulos que brindan a los programadores los medios y servicios necesarios para desarrollar aplicaciones empresariales de acuerdo a sus necesidades. Es importante establecer que no necesariamente se tienen que utilizar todos los módulos, el programador tiene libre albedrío para que módulos utilizar para su aplicación.

En la Figura 24 podemos observar los módulos que componen el marco de trabajo Spring.

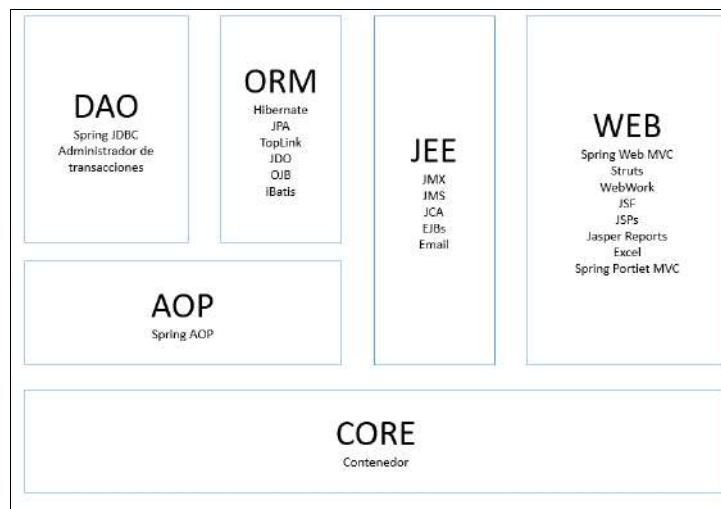


Figura 24. Módulos Spring
Fuente: Elaboración propia

A continuación, se detalla los módulos que conforman el marco de trabajo:

- **Contenedor Core:** Es el módulo que provee la funcionalidad fundamental del marco de trabajo. Contiene la factoría de Beans (*BeanFactory*), que representa al contenedor Spring y la base de la inyección de dependencias.
- **Spring AOP:** Provee el módulo para el soporte de la programación orientada a objetos de una manera muy similar a la inyección de dependencias, el módulo permite el desacoplamiento de los objetos java.
- **DAO (*Data Access Objects Module*):** Este es uno de los módulos más importantes de Spring ya que libera al programador de la interacción con los gestores de base de datos.
- **ORM (*Object Relational Module*):** Este módulo brinda el soporte para diversos marcos ORM como JPA, Hibernate, iBatis, etc.
- **JEE:** Módulo que ofrece el soporte a componentes empresariales como EJBs, Mensajería JMS, etc.
- **Web:** Módulo que brinda el soporte para la construcción de aplicaciones basadas en el patrón MVC.

3.4.2. Arquitectura de un sistema hecho en Spring

A continuación, se explica la estructura de una típica aplicación en Spring, este *framework* está diseñado para facilitar la flexibilidad de la arquitectura de una aplicación.

- Capa de presentación: Debe ser lo más ligera posible, dado que tiene que permitir alternar distintas capas de presentación.
- Capa de Negocios o Servicios: Es la capa responsable de las transacciones del sistema. Esta capa debe ser independiente de la capa de presentación es decir no debería tener conocimiento de la misma y debería ser lo más reutilizable posible.
- Interface DAO: Esta capa no contiene la lógica de negocios. Representa las interfaces (independientes de cualquier tecnología de acceso a datos). La implementación de estas interfaces normalmente usa cualquier tecnología O/R o JDBC, en este proyecto se ha elegido la tecnología JDBC.
- Objetos del dominio: Objetos persistentes que forman parte del modelo de datos.
- Bases de datos: Es el repositorio de información.

3.5. Características principales del Framework Spring de Java

Este apartado está basado en la investigación de (Arauco, 2012) que expone las principales características de Spring.

3.5.1. Inyección de dependencias

La inyección de dependencias es una de las características más importantes de Spring, el objetivo es separar el sistema en un grupo de objetos reutilizables. Sin un módulo central para la gestión de objetos, los mismos tendrán que crear y gestionar sus propias dependencias, y eso repercutiría en tener objetos altamente acoplados.

La solución para este caso pasa por la necesidad de poseer un contenedor encargado de administrar los objetos utilizados por el sistema e inyectarlos a quien los necesite.

Antes de proceder a analizar como contenedores como Spring utilizan esta característica tan importante, podemos analizar como separar las interfaces de la implementación.

En la programación de la plataforma Web se han creado diferentes interfaces, por ejemplo, una con la funcionalidad de generar diversos tipos de reportes tales como HTML ó PDF. Lo ideal es crear una interface que contenga lo siguiente (Figura 25):

```

    *
    * @author Iván Belupú
    */
    public interface ReportGeneratorAlumno {
        public void generate(String [][] table);
    }

```

Figura 25. Interface ReportGenerator
Fuente: Elaboración propia

A continuación se muestra las clases que implementan la interface en mención para la generación de los reportes en los formatos ya mencionados (Figura 26 y Figura 27).

```

    /**
     *
     * @author Iván Belupú
     */
    public class HtmlReportGenerator implements ReportGeneratorAlumno {
        public void generate (String[][] table) {
            System.out.println("Generating HTML report ...");
        }
    }

```

Figura 26. Clase HtmlReportGenerator
Fuente: Elaboración propia

```

    * @author Iván Belupú
    */
    public class PdfReportGenerator implements ReportGeneratorAlumno {
        public void generate (String[][] table) {
            System.out.println("Generating PDF report ...");
        }
    }

```

Figura 27. Clase PdfReportGenerator
Fuente: Elaboración propia

Ahora observemos la clase que muestra la Figura 28:

```

    * @author Iván Belupú
    */
    public class ReportService {

        private ReportGeneratorAlumno reportGenerator = new PdfReportGenerator();

        public void generateAnnualReport(int year) {

            String[][] statistics = null;
            //..... // Obteniendo las estadísticas del año.
            reportGenerator.generate(statistics);

        }

        public void generateMonthlyReport(int year, int month) {

            String[][] statistics = null;
            //..... // obteniendo las estadísticas
            reportGenerator.generate(statistics);

        }

    }

```

Figura 28. Clase ReportService
Fuente: Elaboración propia

La lógica de negocio está representada por la clase ReportService, el formato de salida de los reportes dependerá de cual clase ReportGenerator sea implementada.

Como complemento podemos observar el siguiente diagrama de clases de la Figura 29 que muestra la relación entre las clases ya mencionadas.

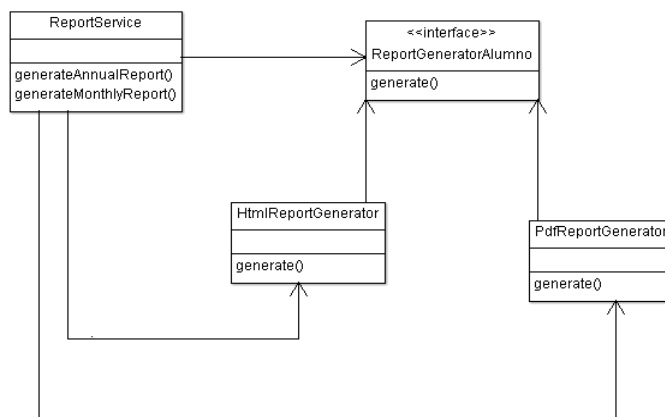


Figura 29. Diagrama de clases
Fuente: Elaboración propia

Analizando el diagrama de clases anterior, observamos que ReportService es el responsable de crear la instancia de tipo ReportGeneratorAlumno, la cual puede ser HtmlReportGenerator ó PdfReportGenerator, causando una dependencia directa aún entre ReportService y las clases mencionadas.

Ahora con Spring como encargado de gestionar el ciclo de vida de los objetos, la clase ReportService no tiene por qué ser la responsable de instanciar a las otras clases, sino que las mismas son “inyectadas” por el contenedor hacia la clase ReportService, por lo tanto, observaremos una gráfica como la que sigue:



Figura 30. Diagrama de clases con inyección
Fuente: Elaboración propia

En la Figura 30, se puede observar que la clase ReportService no debe realizar más la instancia de alguna clase, sino que las mismas son inyectadas por el contenedor Spring.

El principal beneficio de la inyección de dependencias, es el bajo acoplamiento entre clases. Si un objeto sólo conoce a sus dependencias a través de sus interfaces entonces las dependencias pueden ser cambiadas en cualquier momento sin traer como consecuencia cambios drásticos en la aplicación.

3.5.2. Contenedor Spring

La característica principal de Spring está representada por un contenedor ligero responsable de gestionar las funcionalidades que están ligadas a inversión de control e inyección de dependencias.

Spring trae consigo diversas implementaciones del contenedor de Beans que pueden ser categorizadas en dos tipos:

- La más básica conocida como “*BeanFactory*”
- La segunda conocida como “*ApplicationContext*” construida sobre la base del “*BeanFactory*” y con un conjunto mayor de funcionalidades.

Trabajar con la implementación “*BeanFactory*” resulta ideal para aplicaciones simples, pero si se quiere aprovechar todo el poder del marco de trabajo Spring, se tiene que implementar el segundo tipo de contenedor “*ApplicationContext*”.

3.5.3. Uso de los Beans

En una aplicación java tradicional, el ciclo de vida de un Bean es bastante simple; sólo basta con usar la instrucción “new” y está listo para ser usado. Una vez que el Bean no se encuentre siendo usado, puede pasar por el recolector de basura para que se destruya.

En cambio en un contenedor como Spring el ciclo de vida de los Beans gestionados es mucho más sofisticado (Figura 31).

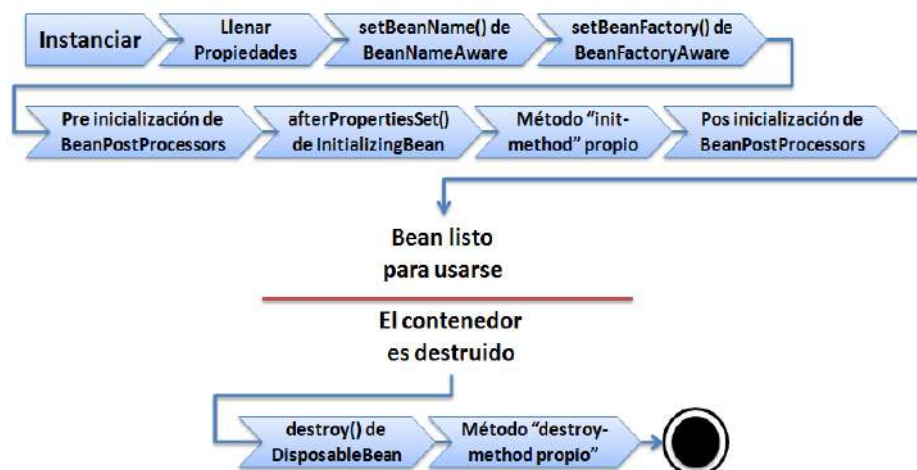


Figura 31. Ciclo de vida de los Beans en Spring

Fuente: Elaboración propia

Se detalla a continuación la imagen anterior:

- Instanciar: Los Beans son instanciados utilizando sus constructores.

- Llenar propiedades: Spring inyecta valores a las propiedades de los Beans.
- *Set Bean Name*: Si el Bean implementa la interface `BeanNameAware`, Spring envía el ID del Bean al método `setBeanName()`.
- *Set Bean Factory*: Si el Bean implementa la interface `BeanFactoryAware`, Spring envía la factoría de Beans al método `setBeanFactory()`.
- *BeanPostProcessor*: Si existe algún `BeanPostProcessor`, Spring ejecuta el método `postProcessBeforeInitialization()`.
- Inicializar Bean: Si el Bean implementa la interface `InitializingBean`, Spring ejecuta el método `afterPropertiesSet()`.
- *BeanPostProcessor*: Si existe algún `BeanPostProcessor`, Spring ejecuta el método `postProcessAfterInitialization()`.
- Bean listo para usarse: En este punto, el Bean está listo para ser usado por la aplicación, y permanecerá en la factoría hasta que deje de necesitarse.
- *Destroy Bean*: Si el Bean implementa la interface `DisposableBean`, el método `destroy()` es ejecutado.

3.5.4. La inyección de datos a través de constructores

También se puede modificar las clases para poder definir constructores dentro de ellas, por lo tanto, el contenedor Spring también puede enviar los valores necesarios para inicializar dichas clases.

El elemento **<constructor-arg>** es usado con la finalidad de enviar valores al constructor del Bean, en caso de no usar dicho elemento, el constructor por defecto del Bean será invocado.

Ejemplos en las Figura 32, Figura 33, Figura 34 y Figura 35:

```

2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package com.spring.interfaces;
7
8
9
10
11
12  public interface ReportGenerator {
13
14      public void generate();
15  }
16

```

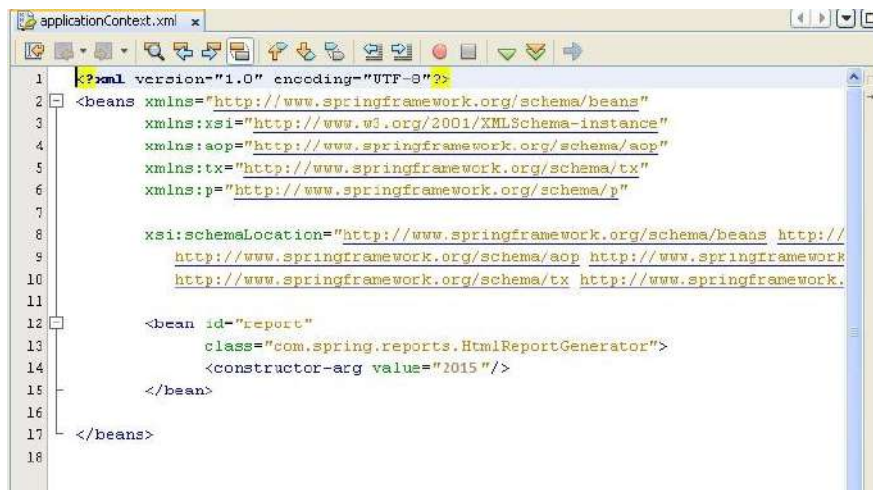
Figura 32. Interface ReportGenerator
Fuente: Elaboración propia

```

5
6 package com.spring.reports;
7
8 import com.spring.interfaces.ReportGenerator;
9
10
11
12
13
14 public class HtmlReportGenerator implements ReportGenerator {
15     private String anio;
16
17     public HtmlReportGenerator(String anio) {
18         this.anio = anio;
19     }
20
21     @Override
22     public void generate() {
23         System.out.println("Año: " + anio + " Generating HTML report ...");
24     }
25
26 }

```

Figura 33. Clase HtmlReportGenerator
Fuente: Elaboración propia

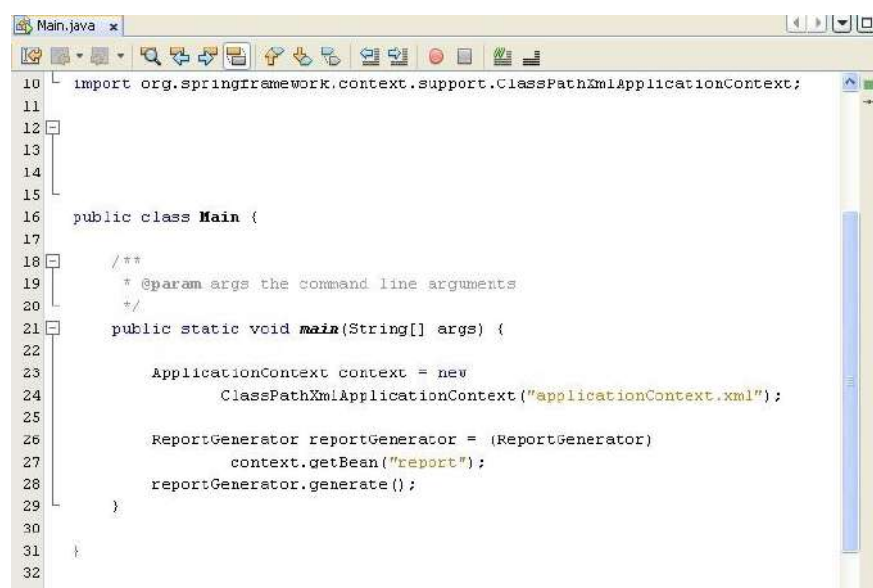


```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:aop="http://www.springframework.org/schema/aop"
5       xmlns:tx="http://www.springframework.org/schema/tx"
6       xmlns:p="http://www.springframework.org/schema/p"
7
8       xsi:schemaLocation="http://www.springframework.org/schema/beans http://
9       http://www.springframework.org/schema/aop http://www.springframework
10      http://www.springframework.org/schema/tx http://www.springframework.
11
12      <bean id="report"
13            class="com.spring.reports.HtmlReportGenerator">
14        <constructor-arg value="2015"/>
15      </bean>
16
17 </beans>
18

```

Figura 34. ApplicationContext.xml
Fuente: Elaboración propia



```

10 import org.springframework.context.support.ClassPathXmlApplicationContext;
11
12
13
14
15
16 public class Main {
17
18     /**
19      * @param args the command line arguments
20      */
21     public static void main(String[] args) {
22
23         ApplicationConte context = new
24             ClassPathXmlApplicationContext("applicationContext.xml");
25
26         ReportGenerator reportGenerator = (ReportGenerator)
27             context.getBean("report");
28         reportGenerator.generate();
29     }
30
31 }
32

```

Figura 35. Clase Main
Fuente: Elaboración propia

3.5.5. Almacenamiento de listas

Los elementos `<list>` y `<set>` son los elementos más utilizados en Spring que permiten almacenar un conjunto de valores determinados. El uso de estos elementos se puede observar en la figura 36 y Figura 37.



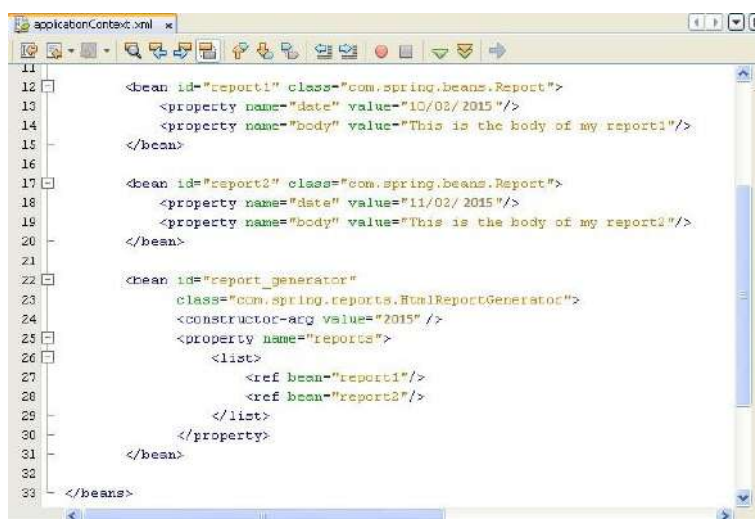
```

18 public class HtmlReportGenerator implements ReportGenerator {
19     private String anio;
20     private Collection<Report> reports;
21
22     public HtmlReportGenerator(String anio){
23         this.anio = anio;
24     }
25
26     @Override
27     public void generate() {
28         System.out.println("Año: " + anio + " Generating HTML report ...");
29         for (Report report: getReports()){
30             System.out.println("Report's Date: " + report.getDate() +
31                               "; Printing text: " + report.getBody());
32         }
33     }
34
35     /**
36      * Return the reports
37      */
38     public Collection<Report> getReports() {
39         return reports;

```

Figura 36. Clase HtmlReportGenerator

Fuente: Elaboración propia



```

12 <bean id="report1" class="com.spring.beans.Report">
13     <property name="date" value="10/02/2015"/>
14     <property name="body" value="This is the body of my report1"/>
15 </bean>
16
17 <bean id="report2" class="com.spring.beans.Report">
18     <property name="date" value="11/02/2015"/>
19     <property name="body" value="This is the body of my report2"/>
20 </bean>
21
22 <bean id="report_generator"
23     class="com.spring.reports.HtmlReportGenerator">
24     <constructor-arg value="2015" />
25     <property name="reports">
26         <list>
27             <ref bean="report1"/>
28             <ref bean="report2"/>
29         </list>
30     </property>
31 </bean>
32
33 </beans>

```

Figura 37. ApplicationContext.xml

Fuente: Elaboración propia

3.5.6. Inyección automática con Autowiring

Anteriormente se ha explicado como el *framework* Spring puede instanciar e inyectar las dependencias a través de elementos como `<property>` ó `<constructor-arg>`. Asimismo, también es posible indicarle a Spring la definición automática del objeto a instanciar y que pueda ser enviado a esa propiedad se le conoce como “*autowiring*”.

En la actualidad existen 4 tipos de autowiring.

- *byName*: Spring coloca el Bean cuyo nombre sea exactamente el mismo (incluido mayúsculas) que el de propiedad a la cual se le pasará la dependencia.
- *byType*: Spring coloca el Bean cuyo tipo de variable coincida con el tipo de propiedad a la cual se le pasará la dependencia. En caso que existan más de una coincidencia, se lanzará una excepción (*UnsatisfiedDependencyException*).
- Constructor: Spring intenta ubicar uno ó más Beans en el contenedor con los parámetros de uno de los constructores del Bean que está siendo declarado.
- *Autodetect*: Elige entre *autowiring* “*byType*” ó “constructor”.

En la Figura 38, observamos la declaración de un <list> cuyos elementos hacen referencia a las dos instancias de “reports” (report1 y report2).

Después cuando se declara la clase HtmlReportGenerator con un *autowire* “*byName*” se está indicando que las listas declaradas anteriormente se injecten en las propiedades “reports”.



```

13 <util:list id="reports">
14   <ref bean="report1"/>
15   <ref bean="report2"/>
16 </util:list>
17
18 <bean id="report1" class="com.spring.beans.Report">
19   <property name="date" value="10/02/2015"/>
20   <property name="body" value="This is the body of my report1"/>
21 </bean>
22
23 <bean id="report2" class="com.spring.beans.Report">
24   <property name="date" value="11/02/2015"/>
25   <property name="body" value="This is the body of my report2"/>
26 </bean>
27
28 <bean id="report_generator"
29   class="com.spring.reports.HtmlReportGenerator" autowire="byName">
30   <constructor-arg value="2015"/>
31 </bean>
32

```

Figura 38. ApplicationContext.xml

Fuente: Elaboración propia

En la figura 39 se muestra un ejemplo donde el *autowiring* es del tipo “*byType*”, similar al ejemplo anterior sólo que en este caso Spring busca las coincidencias por tipo de datos.

```

12
13 <util:list id="reports">
14   <ref bean="report1"/>
15   <ref bean="report2"/>
16 </util:list>
17
18 <bean id="report1" class="com.spring.beans.Report">
19   <property name="date" value="10/02/2015"/>
20   <property name="body" value="This is the body of my report1"/>
21 </bean>
22
23 <bean id="report2" class="com.spring.beans.Report">
24   <property name="date" value="11/02/2015"/>
25   <property name="body" value="This is the body of my report2"/>
26 </bean>
27
28 <bean id="report_generator"
29   class="com.spring.reports.HtmlReportGenerator" autowire="byType">
30   <constructor-arg value="2015"/>
31 </bean>
32
33 </beans>

```

Figura 39. applicationContext.xml

Fuente: Elaboración propia

3.5.7. Soporte para acceso a datos

El patrón DAO (Data Access Object) es uno de los más importantes utilizados en aplicaciones java empresariales, es así que Spring lo incluyó dentro de sus características dando el soporte para el mismo.

El patrón DAO usa El JDBC Framework que está construido sobre la base del J2SE JDBC Api; es un elemento considerado también importante dentro del marco de trabajo ya que el encargado del acceso a la información.

El propósito principal del patrón DAO es separar la capa de la lógica de negocios con la tecnología que se usa para acceder a los datos, para que si en algún momento se cambia el tipo de base de datos a utilizar no sea tan complicado hacer el cambio en la programación de la aplicación. En la Figura 40 se observa el uso del patrón DAO:

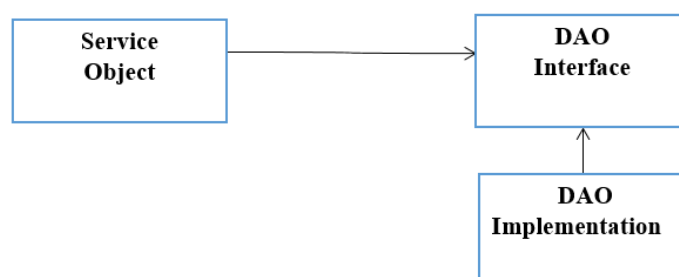


Figura 40. Patrón DAO

Fuente: Elaboración propia

En el caso de la figura anterior, se observa que los objetos de servicio acceden a los objetos DAO a través de las “interfaces”; esto trae como consecuencia que los objetos de servicio no dependan de una implementación para el acceso a datos en particular.

En la actualidad Spring, provee un conjunto de clases abstractas DAO que permiten simplificar el acceso a bases de datos; por ejemplo, para el caso de JDBC, existe “*JdbcDaoSupport*”, la cual provee métodos para definir la configuración de un “DataSource” y ofrece además plantillas preconfiguradas para la gestión de los datos.

3.5.8. El uso de JDBC Framework

JDBC ha estado disponible desde la versión 1.1 de Java y en la actualidad es una de las más importantes clases java que forman parte del entorno de desarrollo, tanto así que es casi improbable concebir el acceso a bases de datos sin utilizar JDBC.

En Spring se siguen usando las principales interfaces de JDBC tales como *Connection*, *DataSource*, *Statement*, *PreparedStatement*, *CallableStatement* y *ResultSet*; todas ellas de uso cotidiano cuando se trata de acceder a fuentes de datos; y también el control de excepciones a través de la clase *SQLException*.

Es importante hacer una mención especial a la interface *DataSource*, la cual es parte de la versión 2 de las librerías JDBC y que define una manera estándar para interactuar con un *pool* de conexiones sin tener que especificar parámetros específicos de base de datos como nombre del driver ó la conexión url.

El objetivo principal de Spring JDBC es seguir usando aquellas características JDBC que funcionan bien y abstraer o evitar que el programador se preocupe por algunos problemas que pueden aparecer producto del uso de JDBC especialmente relacionados con la gestión de las conexiones y excepciones.

A continuación, se describe el proceso para el acceso a una base de datos:

- Obtención del DataSource.
- Establecer la conexión con la base de datos.
- Creación del PreparedStatement.
- Si la consulta a ejecutar tiene parámetros, los mismos deben ser insertados.
- Ejecución de la consulta y obtención del ResultSet.
- Recorrido del ResultSet.
- Cierre del ResultSet.
- Cierre del PreparedStatement.
- Cierre de la conexión.

La primera ventaja que se observa con Spring JDBC frente a una conexión de base de datos tradicional es que no es necesario el uso de bloques *try/catch* para controlar las excepciones que pudieren aparecer, ya que Spring JDBC lo realiza. Spring captura y traduce toda excepción JDBC al tipo *DataAccessException*.

Spring permite varias opciones para la creación de un “*DataSource*” de cualquier tipo:

- *DataSource* definido por un driver JDBC (Figura 41).
- *DataSource* ubicado a través de JNDI (figura 42)
- *DataSource* que define un grupo de conexiones (*Pool* de conexiones) (figura 43).

La manera más simple de crear una conexión a una base de datos es a través de un *DataSource* definido por algún driver JDBC, para ello Spring ofrece dos clases que pueden ser utilizadas para definir el tipo de *DataSource* a utilizar:

- *DriverManagerDataSource* que retorna una nueva conexión cada vez que se solicita.
- *SingleConnectionDataSource* que retorna la misma conexión cada vez que se solicita.

Ambos tipos de *DataSource* son recomendables sólo para aplicaciones pequeñas, en el primer caso el costo de performance sería alto si intentamos utilizarlo en una aplicación más grande, y en el segundo caso al ofrecer sólo una conexión no sería lo ideal para una aplicación con múltiples hilos.

```
<bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName" value="sun.jdbc.odbc.JdbcOdbcDriver"/>
    <property name="url" value="jdbc:odbc:SpringODBC"/>
    <property name="username" value=""/>
    <property name="password" value=""/> </bean>
```

Figura 41. *DataSource* definido con drive JDBC

Fuente: Elaboración propia

La segunda forma de definir un *DataSource* es permitiendo su localización a través de un JNDI. Normalmente las aplicaciones Spring son desplegadas en un servidor de aplicaciones como Jboss, Tomcat, Glassfish, etc., la ventaja de estos servidores es que permiten a los desarrolladores configurar los *DataSources* y accederlos vía JNDI.

El beneficio de configurar *DataSources* bajo esta modalidad es que el mismo puede ser gestionado externamente y no en la aplicación, por lo tanto esta no debe preocuparse por la configuración de la conexión sino simplemente de la obtención de la información; asimismo es importante recalcar que los *DataSources* gestionados

por un servidor de aplicaciones tienen mejor performance que los gestionados localmente.

```
<bean id="dataSource" class="org.springframework.jndi.JndiObjectFactoryBean"
    scope="singleton">
    <property name="jndiName" value="/jdbc/MiDataSource"/>
    <property name="resourceRef" value="true"/>
</bean>
```

Figura 42. DataSource definido por JNDL
Fuente: Elaboración propia

La tercera forma de definir un *DataSource* es configurando un *Pooled DataSource* directamente en la aplicación Spring así es como se ha configurado en el servidor Jboss; para ello el proyecto Apache ofrece una alternativa a través del *Jakarta Commons Database Connections Pools* (DBCP).

```
<datasource jta="false" jndi-name="java:/labsacDatasource" pool-name="labsacDS" enabled="true" use-cm="false">
  <property name="driverClassName" value="org.hsqldb.jdbcDriver"/>
  <connection-url>jdbc:mysql://localhost:3306/labsac</connection-url>
  <driver-class>com.mysql.jdbc.Driver</driver-class>
  <driver>mysql</driver>
  <security>
    <user-name>root</user-name>
    <password>root</password>
  </security>
  <validation>
    <validate-on-match>false</validate-on-match>
    <background-validation>false</background-validation>
  </validation>
  <statement>
    <share-prepared-statements>false</share-prepared-statements>
  </statement>
</datasource>
```

Figura 43. DataSource para un pool de Conexiones
Fuente: Elaboración propia

A continuación se muestra en la Figura 44 el siguiente código tradicional utilizando JDBC para el ingreso de datos de los usuarios del sistema.

```

* @author Iván Belupú
*/
public class ConexionDatos {

    private String sql = "insert into usuario (id, email, password) values (?, ?, ?)";

    public void saveUsuario(Usuario usuario) {
        Connection conn = null;
        PreparedStatement stmt = null;
        try {
            conn = dataSource.getConnection();
            stmt = conn.prepareStatement(sql);
            stmt.setString(1, usuario.getId());
            stmt.setString(2, usuario.getEmail());
            stmt.setString(3, usuario.getPassword());
            stmt.execute();
        } catch (SQLException e) {
            //.....
        } finally {
            try {
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException e) {
            }
        }
    }
}

```

Figura 44. Típica aplicación JDBC
Fuente: Elaboración propia

De la figura anterior observamos muchas líneas de códigos para una simple inserción, esto debido a que JDBC requiere que el programador gestione adecuadamente las conexiones y sentencias a ejecutar, así como también el control de las excepciones.

En cambio, Spring ofrece tres tipos de plantillas para el uso de JDBC con la finalidad de facilitar las operaciones en aplicaciones que necesiten acceso a bases de datos; podemos mencionar:

- JdbcTemplate que provee un acceso simple a una base de datos a través de JDBC y consultas simples parametrizadas.
- NamedParameterJdbcTemplate que está enfocado principalmente ofrecer flexibilidad en los parámetros que están siendo pasados a una instrucción sql.

3.5.9. El uso Jdbc Template

Para utilizar templates JDBC lo único que se necesita es configurar previamente un *DataSource* y referenciarlo al atributo “*dataSource*” del template (Figura 45).

```
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
    <property name="dataSource" ref="dataSource"/>
</bean>
```

Figura 45. JDBC Template
Fuente: Elaboración propia

Posteriormente se puede insertar una referencia del “*jdbcTemplate*” a la clase y simplificar las operaciones JDBC (Figura 46).

```
private String sql = "insert into usuario (id, email, password) values (?, ?, ?)";

public void saveUsuario(Usuario usuario) {
    jdbcTemplate.update(sql,
        new Object[]{
            usuario.getId(), usuario.getEmail(), usuario.getPassword()});
}
```

Figura 46. JDBC Template
Fuente: Elaboración propia

```
private String sql = "select id, email, password from usuario where id = ?";

public Usuario getUsuarioById(String id) {
    List matches = jdbcTemplate.query(sql,
        new Object[]{id},
        new RowMapper() {
            public Object mapRow(ResultSet rs, int rowNum) throws SQLException, DataAccessException {
                Usuario usuario = new Usuario();
                usuario.setId(rs.getString("id"));
                usuario.setEmail(rs.getString("email"));
                usuario.setPassword(rs.getString("password"));
                return usuario;
            }
        });
    return matches.size() > 0 ? (Usuario) matches.get(0) : null;
}
```

Figura 47. JdbcTemplate.query()
Fuente: Elaboración propia

En la Figura 47 se puede observar que el método “*JdbcTemplate.query()*” es el encargado de obtener los registros de la base de datos, este método define tres parámetros:

- Un *String* que representa la consulta sql.
- Un arreglo de tipo *Object* que contiene los valores a ser insertados como parámetros en la consulta sql (En el ejemplo sólo existe un parámetro (id)).
- Un objeto de tipo *RowMapper* encargado de extraer los valores desde un *ResultSet* y construir el objeto de dominio (Usuario). Por cada registro que cumpla con la condición de la consulta sql, se crea un objeto de tipo Usuario y se llena la lista (*matches*) con las coincidencias encontradas.

En los ejemplos anteriores se muestra que el método “saveUsuario” utiliza ciertos parámetros que deben ser definidos en el mismo orden que se encuentra en la instrucción sql, por lo tanto, si cambiamos el orden de los mismos en la consulta también se debe tener en cuenta el orden en que son insertados los valores en el método “saveUsuario”.

Una opción para simplificar este caso es utilizar “*named parameters*”, los cuales permiten asignarle un nombre específico a cada parámetro en la cadena sql y referirnos a ellos desde cualquier parte de la aplicación. En la Figura 48 se observa cómo cambiaría el código para el caso de inserción de datos en una base de datos.

```
private String sql2 = "insert into usuario (id, email, password) values (:id, :email, :password)";
public void saveUsuario (Usuario usuario) {
    Map parameters = new HashMap();
    parameters.put("id", usuario.getId());
    parameters.put("email", usuario.getEmail());
    parameters.put("password", usuario.getPassword());
    jdbcTemplate.update (sql2, parameters);
}
```

Figura 48. NamedParameterJdbcTemplate

Fuente: Elaboración propia

3.5.10. Soporte para Spring DAO

Una aplicación típica en Spring tiene necesariamente que implementar una capa DAO que es responsable de la persistencia de los objetos en la base de datos sin importar la tecnología a utilizar; un ejemplo se muestra en la Figura 49.

```
/**
 *
 * @author Iván Belupú
 */

public class JdbcAppDao implements AppDao {
    //...
    private JdbcTemplate jdbcTemplate;

    public void setJdbcTemplate (JdbcTemplate jdbcTemplate) {
        this.jdbcTemplate = jdbcTemplate;
    }
    public void saveUsuario(Usuario usuario) {
        //...
    }
}
```

Figura 49. Spring DAO Interface

Fuente: Elaboración propia

Lo visto en la figura anterior no es un problema complicado si es que sólo poseemos un solo DAO, pero si se desea implementar múltiples DAO, el problema sería repetir varias líneas iguales de código.

Una solución para este inconveniente es la creación de una clase padre común para todos los objetos DAO donde resida la propiedad “JdbcTemplate”,

posteriormente todas las clases DAO que extiendan esta clase podrían usar esta propiedad para acceder a los datos (Figura 50).

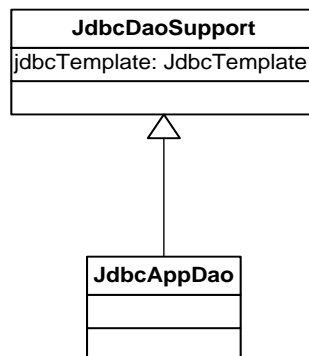


Figura 50. Diagrama de clases implementando una clase padre.
Fuente: Elaboración propia

La idea de crear una clase base DAO, es inicializar el valor del atributo “jdbcTemplate” y que el mismo pueda ser heredado por las clases hijas con tan sólo extender la clase padre (Figura 51).

```

* @author Iván Belupú
*/

public class JdbcAppDao extends JdbcDaoSupport implements AppDao {
    public void saveUsuario(Usuario usuario) {
        getJdbcTemplate().update(
            //...
        );
    }
}
  
```

Figura 51. Class JDBCAppDAO
Fuente: Elaboración propia

Ahora, la “applicationContext.xml” quedaría como se muestra en la Figura 52:

```

<bean id="jdbcAppDao" class="com.spring.myjdbc.JdbcAppDao">
    <property name="dataSource" ref="dataSource"/>
</bean>
  
```

Figura 52. ApplicationContext.xml
Fuente: Elaboración propia

Ahora, inyectando una referencia del “dataSource” a la clase, ella podrá invocar al método “getJdbcTemplate()” y obtener la instancia correspondiente. Esto elimina la necesidad de declarar un atributo de tipo “JdbcTemplate” en todas las clases.

3.5.11. Manejo de las transacciones

Las transacciones es otra de las características importantes de Spring, tienen un rol importante en el desarrollo de software, con la ayuda de sus servicios la aplicación se cerciora que los datos nunca van a estar en un estado inconsistente.

Una transacción puede ser concebida como el conjunto de operaciones a ejecutar; si todas son exitosas la transacción será exitosa, pero si alguna falla, la transacción se anulará por completo.

Una aplicación típica Java que utiliza JDBC como método de persistencia de información, puede definir implementar un manejo transaccional en sus operaciones tal como se observa en la siguiente Figura 53:

```
public void GetUsuario (String codigo, String usuario) {
    Connection conn = null;
    try {
        conn = dataSource.getConnection();
        conn.setAutoCommit(false);
        //...
        conn.commit();
    } catch (SQLException e) {
        if (conn != null) {
            try { conn.rollback();
            } catch (SQLException e1) { }
        }
        //...
    }
}
```

Figura 53. Clase con manejo transaccional
Fuente: Elaboración propia

Un gestor de transacciones es el encargado de asegurar las características de una transacción: atomicidad, durabilidad y aislamiento. En caso de error en la ejecución de la transacción la base de datos debe volver a su estado anterior.

Las transacciones en J2EE normalmente son gestionadas por el *Java Transaction Service* (JTS) en conjunto con el Java Transaction Api (JTA). Aquí se definen cinco tipos de actores a mencionar:

- Un Gestor de transacciones, responsable de la demarcación, gestión de recursos, sincronización y propagación de las transacciones.
- Un Gestor de recursos que provee acceso a recursos transacciones, por ejemplo: servidor de base de datos, de mensajería, etc.
- Un Servidor de aplicaciones, que provee el entorno de ejecución para las aplicaciones y gestiona el estado de las transacciones.
- Una Aplicación que opera normalmente en un servidor de aplicaciones.
- Un gestor de comunicaciones, que facilita la propagación de la transacción entre múltiples gestores de transacciones.

```
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource"/>
</bean>
```

Figura 54. Declaración de una transacción JDBC
Fuente: Elaboración propia

```

<bean id="transactionManager" class="org.springframework.transaction.jta. JtaTransactionManager">

    <property name ="transactionManagerName" value="java:/TransactionManager"/>

</bean>

```

Figura 55. Declaración de una transacción JTA

Fuente: Elaboración propia

Las transacciones pueden ser clasificadas en dos tipos en función a las necesidades de la aplicación a desarrollar:

- **Transacciones locales:** Si la aplicación sólo involucra a un simple gestor de recursos transaccionales (bases de datos, mensajería, etc.), entonces estamos al frente de una transacción local, por ejemplo, gestionamos las transacciones vía JDBC, todo lo que necesitamos es definir previamente la siguiente línea: `conn.setAutoCommit(false)`.
- **Transacciones globales:** Una transacción global involucra múltiples gestores de recursos y de transacciones que deben coordinar entre ellos para definir si la transacción es exitosa o no.

Las transacciones globales normalmente se basan en el protocolo “*two-phase commit*”, lo que significa que la primera fase consiste en solicitar a todos los recursos involucrados que se preparen para la ejecución de un “*commit*”; si la primera fase se completa y todos los recursos están listos para la ejecución, entonces se da paso a la segunda fase. Si algún recurso falla en la preparación de la primera fase, entonces todos los recursos deberán deshacer (*rollback*) todos los cambios durante la segunda fase.

Capítulo 4

Programación y funcionamiento de la plataforma Web

En este capítulo se presentan los materiales y métodos usados para la programación de la plataforma Web. También se hace una descripción del funcionamiento de la plataforma ya instalada en los servidores del laboratorio de Sistemas Automáticos de Control corriendo desde un navegador Web.

4.1. Materiales y métodos

4.1.1. Compilador Java

Java es un lenguaje de programación orientado a objetos que fue creado por Sun Microsystems en la década de los noventa, para poder funcionar en distintos tipos de procesadores.

La principal característica de Java es su máquina virtual esto permite que el código Java, una vez compilado, puede llevarse sin tener que modificarlo sobre cualquier computadora, y ejecutarlo, la máquina virtual es la encargada de interpretar el código que en este caso se trata de archivos .class y convertirlos a código que pueda entender la pc, donde esté corriendo el programa.

Java define tres plataformas para diversos entornos de aplicaciones como son:

- Java ME (*Java Platform Micro Edition*) o J2ME, es java para móviles está orientado a celulares de limitados recursos.
- Java SE (*Java Platform, Standard Edition*) o J2SE, se utiliza para desarrollar aplicaciones de escritorio para PC.
- Java EE (*Java Platform, Enterprise Edition*) o J2EE, es el entorno que se ha utilizado en esta investigación, está orientado a entornos distribuidos empresariales a través de Web.

La plataforma virtual Web está desarrollada bajo el compilador de Sun Microsystem (JDK1.7.0) para servidores de 64bits ya que el servidor en el que se implementó trabaja con sistema operativo CentOS para 64 bit. Esto no quiere decir que no se pueda implementar en servidores que trabajen bajo otros sistemas

operativos, de hecho, la característica multiplataforma de Java es una de sus mayores ventajas.

4.1.2. Entorno de desarrollo integrado de programación

Para la implementación del Entorno de Desarrollo Integrado (IDE) se ha utilizado el software NetBeans 8.0, el cual es un entorno de desarrollo multilenguaje muy utilizado tanto por programadores novatos como por profesionales.

Es *open source* y por lo tanto su uso es gratuito, fue creado por Sun la misma compañía que creó Java, esta plataforma IDE permite que las aplicaciones sean por secciones, que en programación se llaman paquetes.

Las aplicaciones que son construidas a partir de paquetes se hacen fácilmente extensibles si se les agregan nuevos paquetes. Es así que la aplicación desarrollada puede ser fácilmente ampliada por otros programadores de software.

Si evaluamos su eficiencia para programación Web, este IDE da soporte a frameworks comerciales como son Struts, Hibernate, y Spring, a parte de los servicios comunes que se ofrecen para las aplicaciones de escritorio.

4.1.3. Servidor Web

Como servidor de Aplicaciones Web se usó el Jboss 7. Es un servidor gratuito basado en patrones de J2EE (*Java Enterprise Edition*), potente y además multiplataforma.

Está implementado totalmente en Java es decir en Java puro. Al tener esta característica puede ser utilizado en cualquier sistema operativo que lo soporte dado la particularidad multiplataforma de Java.

JBoss implementa todo el paquete de servicios de J2EE además adopta una arquitectura de código abierto, es por ese motivo que es un servidor Web que tiene gran acogida entre los desarrolladores de software de grandes empresas.

Las principales características de JBoss son:

- Producto *open source*.
- Confiable para aplicaciones empresariales de gran tamaño.
- Soporte para diferentes servicios java.

4.1.4. Base de datos Mysql

Para el desarrollo de esta plataforma se decidió usar el gestor de base de datos Mysql (Egan et al., 2000). Su principal característica es que gestiona las bases de datos relacionales poniendo las tablas en ficheros diferentes, es multiusuario administrando varios hilos de conexión a la vez.

Mysql es un gestor de base de datos ligero, rápido y fácil de usar. Un buen ejemplo de ello es que uno de los motores de base de datos más buscados en internet, una de las posibles razones también es que es gratis para aplicaciones empresariales no comerciales.

Mysql es multiplataforma, mayoritariamente los desarrolladores lo trabajan sobre los sistemas operativos de Windows y Linux, pero también se puede instalar sobre otras plataformas como IOS, Hp-Ux, Sco.

Las características principales de Mysql son (Zikopoulos, et al., 2000):

- Es un gestor de base de datos: Como se ha mencionado en punto anteriores una base de datos es un conjunto de datos y necesita un gestor de base de datos que viene a ser la aplicación para que se puedan manejar esos datos de una manera eficaz y rápida.
- Las relaciones entre tablas son muy importantes en este gestor de base de datos, los datos almacenados en las tablas mantienen relaciones entre sí para manejar la información de una manera eficiente y segura. El lenguaje de programación SQL es el que se usa en la mayoría de gestores de bases de datos para administrarlo.
- Otra característica importante es que Mysql es *open source*, el código fuente de se puede descargar desde su página oficial siendo accesible para cualquier usuario, también usan licencia GPL para aplicaciones empresariales.
- Debido a la gran colaboración de varios programadores este gestor ha ido mejorando y cada vez aumentando y optimizando la velocidad de sus respuestas, por eso una de las bases de datos más usadas en Internet.
- Existe una gran cantidad de aplicaciones empresariales que lo usan y demuestran su robustez.

4.2. Funcionamiento de la plataforma Web desde un navegador

El funcionamiento de esta plataforma ha sido probado con los alumnos de los cursos de Sistemas Automáticos de Control y Control Industrial de la Facultad de Ingeniería de la Universidad de Piura.

4.2.1. Interfaz gráfica de entrada

Para acceder a la plataforma Web se debe abrir un navegador instalado en la computadora de escritorio, se puede usar Google Chrome, Firefox o Internet Explorer. Una vez abierto, en la barra de direcciones URL, se debe ingresar la siguiente dirección Web: [http:// 200.48.235.220:8080/labsac](http://200.48.235.220:8080/labsac)

Aquí aparecerá la siguiente interfaz gráfica que se muestra en la figura 68 para iniciar sesión, vale recalcar como se ha explicado en capítulos anteriores que la plataforma virtual tiene niveles de seguridad y cada alumno tiene asignado un usuario y clave para iniciar sesión (Figura 56 y Figura 57).



Figura 56. Página inicial de la plataforma virtual
Fuente: Elaboración propia



Figura 57. Vista de la página inicial en un dispositivo móvil
Fuente: Elaboración propia

4.2.2. Funcionamiento de los roles existentes en la plataforma

Se debe ingresar el usuario y contraseña de acuerdo al tipo de rol que se tenga en la plataforma virtual, en este caso existen tres roles creados por defecto:

Rol “administrador” general del sistema

Es el rol que permite la edición en todos los módulos de la plataforma, puede editar, guardar, eliminar.

La plataforma virtual también podrá ser usada para tomar exámenes o prácticas virtuales, un administrador de la plataforma podrá crear los registros para estas funcionalidades (Figura 58).

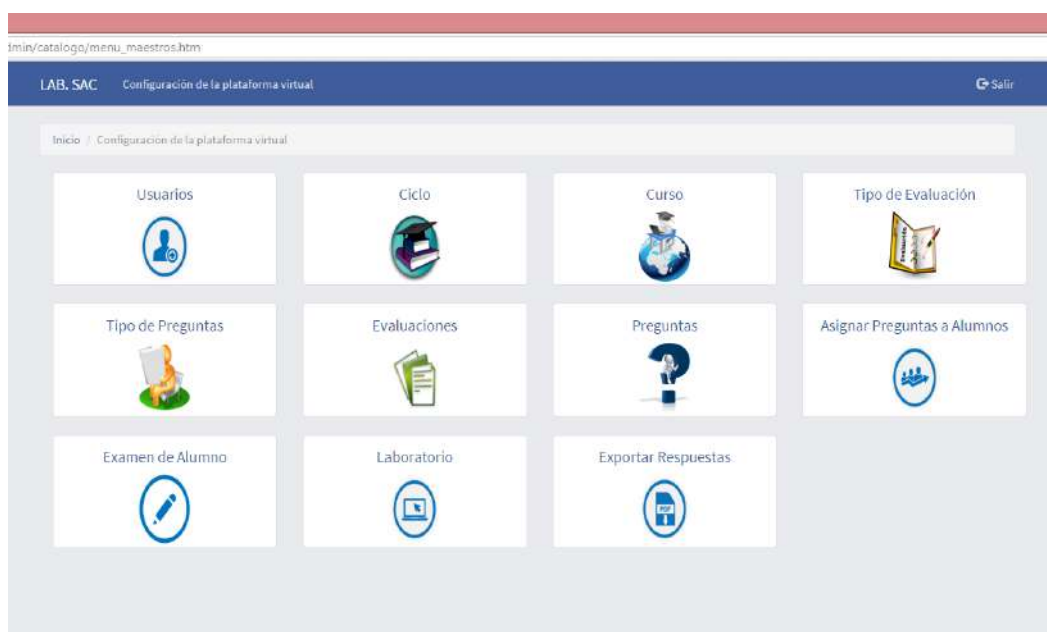


Figura 58. Pantalla de configuración
Fuente: Elaboración propia

Funcionalidades del administrador del sistema

Usuarios

En la Figura 59 se muestra el formulario para crear un nuevo usuario en la plataforma virtual, a este usuario se le debe indicar que tipo de rol tiene en el sistema y a que ciclo y curso va a pertenecer.

LAB. SAC Configuración de la plataforma virtual Salir

Inicio / Usuarios / Editar

Editar

Nombre: IVAN

Apellido Paterno: BELUPU

Doc. Identidad: 43757915

Usuario: admin

Contraseña: *****

Activo: ☒

Rol: administrador

Ciclo - Curso: CICLO 2015-I - SC

Guardar Cancelar

Figura 59. Configuración de usuario
Fuente: Elaboración propia

Después de guardar aparecerá la ventana que lista todos los usuarios creados en el sistema verificando que el nuevo usuario se ha creado (Figura 60).

Inicio / Usuarios Nuevo

Usuarios

10 registros por página Buscar:

#	Usuario	Nombres	Apellido Paterno	Apellido Materno	Doc. Identidad	Ciclo	Curso	Activo	Opción
1	admin	IVAN	BELUPU	AMAYA	43757915	CICLO 2015-I	CURSO SC	Si	
2	alumn	ALUMNO	USUARIO	ALUMNO	12345678	CICLO 2015-I	CURSO SC	Si	

Mostrando 1 hasta 2 de 2 registros

Anterior 1 Siguiente

Figura 60. Ventana que lista los usuarios
Fuente: Elaboración propia

Ciclos

Se debe crear el registro del ciclo en el que se está usando la plataforma, en la Figura 61 se muestra la creación del ciclo 20015-I y en la Figura 62 aparece la ventana que lista todos los ciclos creados en el sistema.



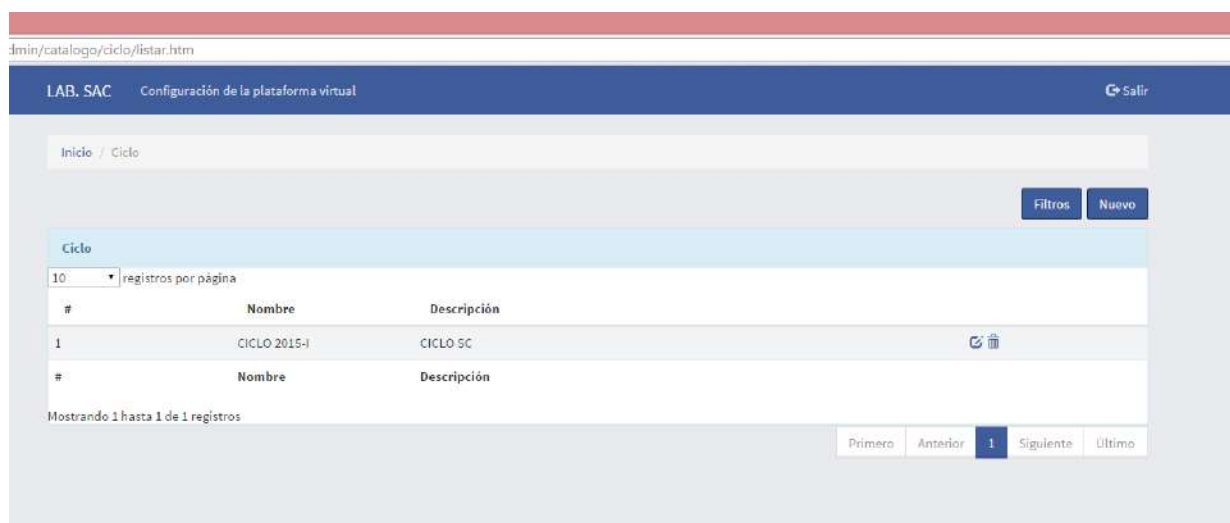
Inicio / Ciclo / Editar

Editar

Nombre:

Descripción:



Figura 61. Formulario para la creación de ciclos
Fuente: Elaboración propia



admin/catalogo/ciclo/listar.htm

LAB. SAC Configuración de la plataforma virtual Salir

Inicio / Ciclo Filtros Nuevo

#	Nombre	Descripción	
1	CICLO 2015-I	CICLO SC	 

Mostrando 1 hasta 1 de 1 registros

Primero Anterior 1 Siguiente Último

Figura 62. Ventana que lista todos los ciclos creados
Fuente: Elaboración propia

Cursos

Se debe crear el registro del curso en el que se está usando la plataforma, en la Figura 63 se muestra la creación del curso SC y en la Figura 64 aparece la ventana que lista todos los cursos creados en el sistema.



Inicio / Curso / Editar

Editar

Nombre:

Descripción:

Figura 63. Formulario para la creación de cursos
Fuente: Elaboración propia

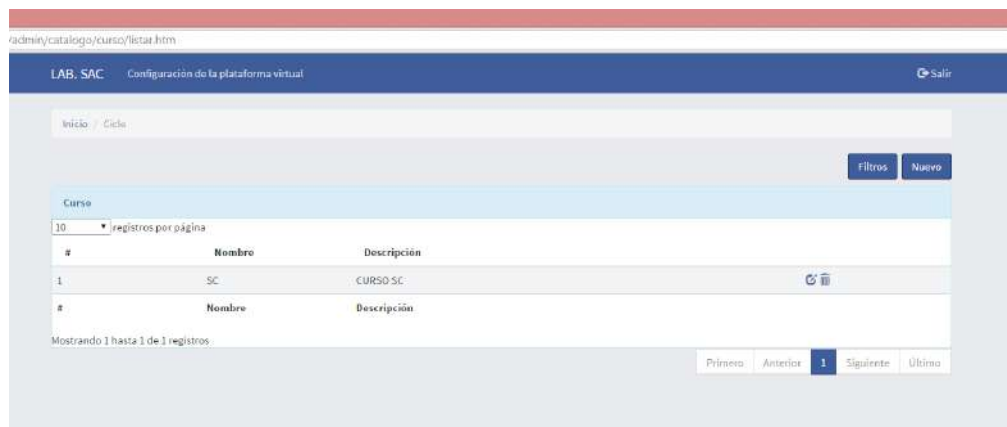


Figura 64. Ventana que lista todos los cursos creados
Fuente: Elaboración propia

Tipo de evaluaciones

Existen tres tipos de evaluaciones cargadas por defecto en el sistema como son: Práctica, Laboratorio y Examen, se ha creado un formulario para dejar abierta la creación de otro tipo de evaluación (Figura 65) y una ventana que lista todas las evaluaciones que soporta el sistema (Figura 66).

Figura 65. Formulario para la creación de tipos de evaluaciones
Fuente: Elaboración propia

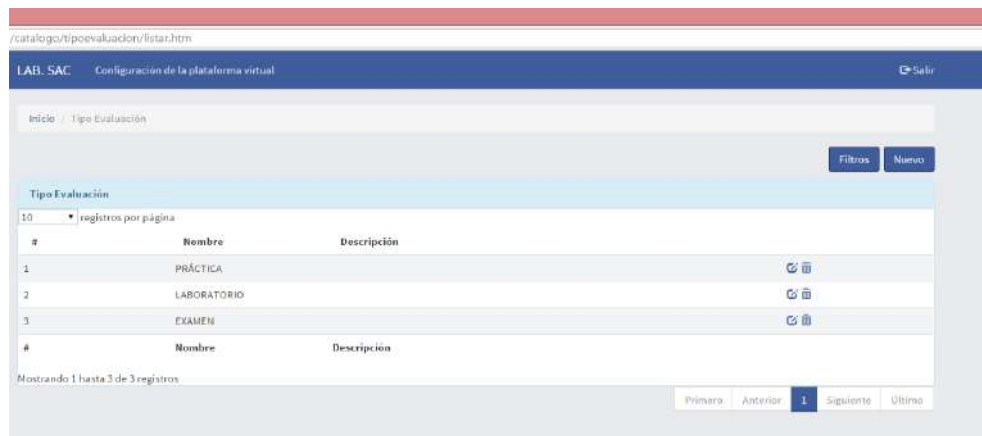


Figura 66. Ventana que lista todos tipos de evaluaciones creados
Fuente: Elaboración propia

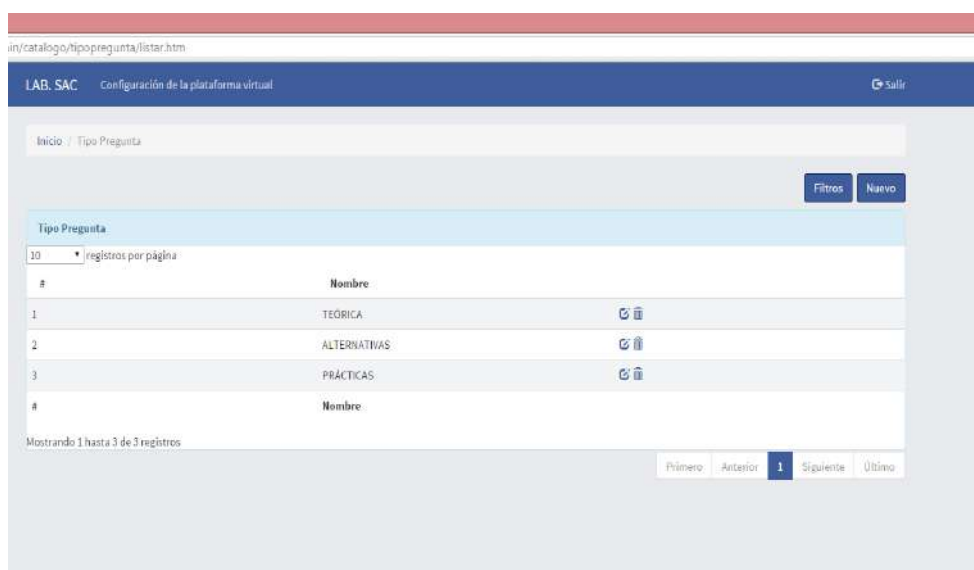
Tipo de preguntas

Existen tres tipos de preguntas cargadas por defecto en el sistema como son: pregunta teórica, pregunta con alternativas y preguntas prácticas, se ha creado un formulario para dejar abierta la creación de otro tipo de preguntas (Figura 67) y una ventana que lista todos los tipos de pregunta que soporta el sistema (Figura 68).









Formulario para la creación de nuevos tipos de preguntas. El formulario muestra un campo de texto con el valor "TEÓRICA" y dos botones: "Guardar" y "Cancelar".

Figura 67. Formulario para la creación de nuevos tipos de preguntas
Fuente: Elaboración propia



Ventana que lista todos los tipos de preguntas. La interfaz muestra una barra de navegación superior con "LAB. SAC" y "Configuración de la plataforma virtual". El título de la página es "Inicio / Tipo Pregunta". Hay botones "Filtros" y "Nuevo".

#	Nombre	
1	TEÓRICA	 
2	ALTERNATIVAS	 
3	PRÁCTICAS	 

Mostrando 1 hasta 3 de 3 registros

Paginación: Primero Anterior 1 Siguiendo Último

Figura 68. Ventana que lista todos tipos de preguntas
Fuente: Elaboración propia

Evaluaciones

Es uno de los módulos más importantes del sistema, desde aquí se empieza la creación de la evaluación que se le va a realizar al alumno en el sistema. En la Figura 69 se muestra la elección de una evaluación de tipo laboratorio.

Configuración de la plataforma virtual

Evaluaciones / Nuevo

Tipo Evaluación

10 registros por página

#	Nombre	Descripción	
1	PRÁCTICA		✓
2	LABORATORIO		✓
3	EXAMEN		✓
<input type="text"/>	<input type="text"/>	<input type="text"/>	

Mostrando 1 hasta 3 de 3 registros

Primero Anterior 1 Siguiendo Último

Cerrar

Figura 69. Ventana para la elección del tipo de evaluación
Fuente: Elaboración propia

Después de elegir el tipo de evaluación se procede a asignarle un nombre y a guardar el registro (Figura 70 y Figura 71).

Evaluaciones / Editar

Tipo Evaluación LABORATORIO

Nombre Análisis de respuesta en el tiempo

Descripción Laboratorio 1

Guardar Cancelar

Figura 70. Formulario para la creación de la evaluación
Fuente: Elaboración propia



Figura 71. Ventana que lista la creación de laboratorios
Fuente: Elaboración propia

Preguntas

En este módulo se crea las preguntas que se van a tomar en las evaluaciones creadas en el sistema. En la Figura 72 se muestra la elección de un tipo de pregunta para la evaluación.



Figura 72. Ventana para la elección del tipo de pregunta
Fuente: Elaboración propia

También se tiene que asignar a que evaluación va ser asignada esa pregunta (Figura 73).

Tipo Evaluación ×

10 registros por página

#	Tipo Evaluación	Evaluación	Descripción	
1	LABORATORIO	Análisis de respuesta en el tiempo	Laboratorio 1	✓
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Mostrando 1 hasta 1 de 1 registros

Primero
Anterior
1
Siguiente
Último

Figura 73. Ventana para la elección del tipo de pregunta
Fuente: Elaboración propia

Después de elegir el tipo de pregunta y la evaluación a donde se va asigna la pregunta, se procede a crear la pregunta y a guardar el registro (Figura 74).

Inicio / Preguntas / Editar

Editar

Evaluación

Tipo Pregunta

Se aplica una entrada constante a un sistema el sistema primero oscila pero luego converge a cero, entonces es estable.

Figura 74. Ventana para la elección del tipo de pregunta
Fuente: Elaboración propia

Asignación de preguntas a los alumnos

En este módulo se asignan las preguntas creadas a los alumnos que van a rendir la evaluación. En la Figura 75 se muestra la elección del alumno al que se le va asignar la pregunta.

10 registros por página

#	Nombre	Apellido Paterno	
1	IVAN	BELUPU	✓
2	ALUMNO	USUARIO ALUMNO	✓
#	Nombre	Apellido Paterno	

Mostrando 1 hasta 2 de 2 registros

Primero Anterior 1 Siguiente Último

Cerrar

Figura 75. Ventana para la elección del alumno al que se le va asignar la pregunta
Fuente: Elaboración propia

También se tiene que elegir la pregunta que le va ser asignada al alumno (Figura 76).

Preguntas

10 registros por página

#	Evaluación	Tipo de Pregunta	Descripción	
1	Análisis de respuesta en el tiempo	TEÓRICA	Se aplica una entrada constante a un sistema el sistema primero oscila pero luego converge a cero, entonces es estable	✓
2	Análisis de respuesta en el tiempo	TEÓRICA	Un sistema tiene dos polos , se desea aproximar a un sistema de primer orden indique las condiciones para ello.	✓
3	Análisis de respuesta en el tiempo	TEÓRICA	Puedo implementar un sistema feedforward sin sensor	✓
4	Análisis de respuesta en el tiempo	TEÓRICA	Un sistema automático de control no puede ser analógico	✓
5	Análisis de respuesta en el tiempo	TEÓRICA	Cual es la diferencia y semejanzas de dos sistemas cuyos diagramas de Nyquist son (líneas con flechas es el eje de coordenadas), ambos diagramas empiezan en $w=0$ en el eje real:	✓
#	Evaluación	Tipo de Pregunta	Descripción	

Mostrando 1 hasta 5 de 6 registros

Primero Anterior 1 2 Siguiente Último

Figura 76. Ventana para la elección del tipo de pregunta
Fuente: Elaboración propia

Después de elegir usuario y la pregunta asignada, se procede a guardar el registro (Figura 77).

Inicio / Asignación de Preguntas a Alumnos / Editar

Editar

Alumno: IVAN BELUPU

Pregunta: Se aplica una entrada constante a un sistema el sistema p...

Guardar Cancelar

Figura 77. Formulario para la asignación de preguntas a los alumnos
Fuente: Elaboración propia

Finalmente, después de guardar se muestra la ventana que lista todas las preguntas que se van asignando a los alumnos (Figura 78).

LAB. SAC Configuración de la plataforma virtual Salir

Inicio / Preguntas

Filtros Nuevo

Asignación de Preguntas a Alumnos

10 registros por página

#	Nombre Alumno	Apellido Alumno	Pregunta
1	IVAN	BELUPU	Se aplica una entrada constante a un sistema el sistema primero oscila pero luego converge a cero, entonces es estable
2	IVAN	BELUPU	Un sistema tiene dos polos, se desea aproximar a un sistema de primer orden indique las condiciones para ello.
4	IVAN	BELUPU	Puedo implementar un sistema feedforward sin sensor
5	IVAN	BELUPU	Un sistema automático de control no puede ser analógico
6	IVAN	BELUPU	Cual es la diferencia y semejanzas de dos sistemas cuyos diagramas de Nyquist son (líneas con flechas es el eje de coordenadas), ambos diagramas empiezan en $w=0$ en el eje real:
7	IVAN	BELUPU	Un sistema con retroalimentación positiva es inestable

Mostrando 1 hasta 6 de 6 registros

Primero Anterior 1 Siguiente Último

Figura 78. Ventana que muestra la asignación de preguntas a los alumnos
Fuente: Elaboración propia

Rol “alumno”

Es el rol creado para los alumnos, este rol sólo permite a los alumnos ver su evaluación y en determinado caso el laboratorio a desarrollar.

Para desarrollar el laboratorio virtual el alumno debe tener un mínimo de conocimientos sobre el tema del laboratorio a realizar, es por eso que después de iniciar la sesión se les pide solucionar un test o cuestionario para validar si están capacitados para desarrollar el laboratorio virtual (Figura 79).

min/catalogo/alumnopregunta/formAlumnoPregunta.htm?id=1

LAB. SAC Configuración de la plataforma virtual

Inicio / Asignación de Preguntas a Alumnos / Nuevo

Test para el Alumno: Iván Belupú

Editar

(1.) Se aplica una entrada constante a un sistema el sistema primero oscila pero luego converge a cero, entonces es estable

Se aplica una entrada constante a un sistema el sistema primero oscila pero luego converge a cero, entonces es estable

(2.) Un sistema tiene dos polos, se desea aproximar a un sistema de primer orden indique las condiciones para ello.

Un sistema tiene dos polos, se desea aproximar a un sistema de primer orden indique las condiciones para ello.

(3.) Puedo implementar un sistema feedforward sin sensor

Puedo implementar un sistema feedforward sin sensor

(4.) Un sistema automático de control no puede ser analógico

Un sistema automático de control no puede ser analógico

Figura 79. Ejemplo de un test de tipo práctico
Fuente: Elaboración propia

Si el alumno responde de forma correcta más del 80% del cuestionario, puede realizar el laboratorio de lo contrario se le brindará una segunda oportunidad con un test diferente, si no logra resolver el segundo test, no podrá realizar el laboratorio virtual.

Los alumnos que aprobaron el test cargarán la página del laboratorio con las respectivas cajas de texto para ingresar la información del proceso y empezar la simulación (Figura 80 y Figura 81).

/catalogo/laboratoriouno/form.htm

LAB. SAC Configuración de la plataforma virtual

Inicio / Nuevo

Nuevo

Numerador: Ingrese numerador

Denominador: Ingrese denominador

Retardo: Ingrese retardo

Tiempo de Simulación: Ingrese Tiempo de Simulación

Procesar Cancelar

Figura 80. Ejemplo de un test de tipo práctico
Fuente: Elaboración propia

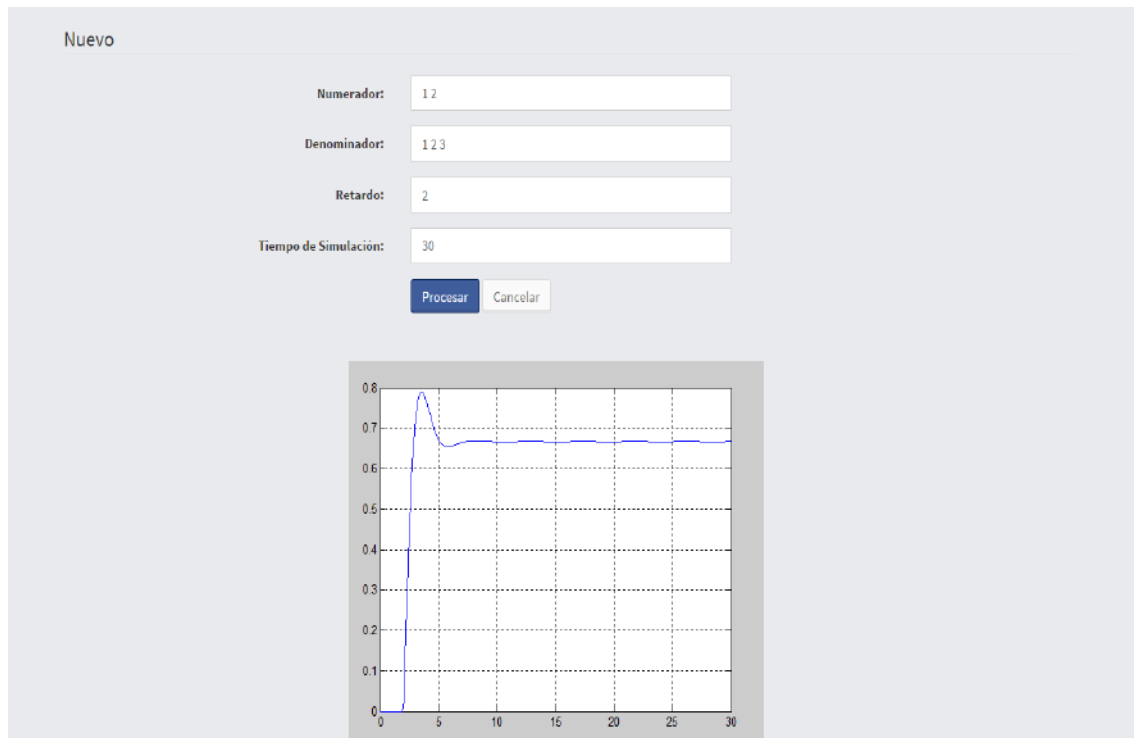


Figura 81. Gráfica con datos procesados
Fuente: Elaboración propia

4.2.3. Ejemplo de laboratorio virtual

Como ejemplo de los laboratorios virtuales implementados podemos citar el de control en variables de estados del sistema péndulo invertido. Inicialmente el usuario debe linealizar el modelo matemático del péndulo invertido a un modelo en espacio de estados para obtener las matrices de estado.

El usuario debe ingresar a la plataforma Web las matrices para graficar los estados y la salida del sistema a lazo abierto, calculando el determinante de las matrices de observabilidad y controlabilidad. De esta manera el usuario puede evaluar si el sistema es controlable y observable parcial o total. Así mismo, para la sintonización del controlador proporcional en espacio de estados, el usuario debe ingresar los polos del controlador y los polos del observador. Tomando en cuenta los polos, la plataforma Web grafica los estados y las salidas a lazo cerrado (ver Figura 82, Figura 83, Figura 84, Figura 85).

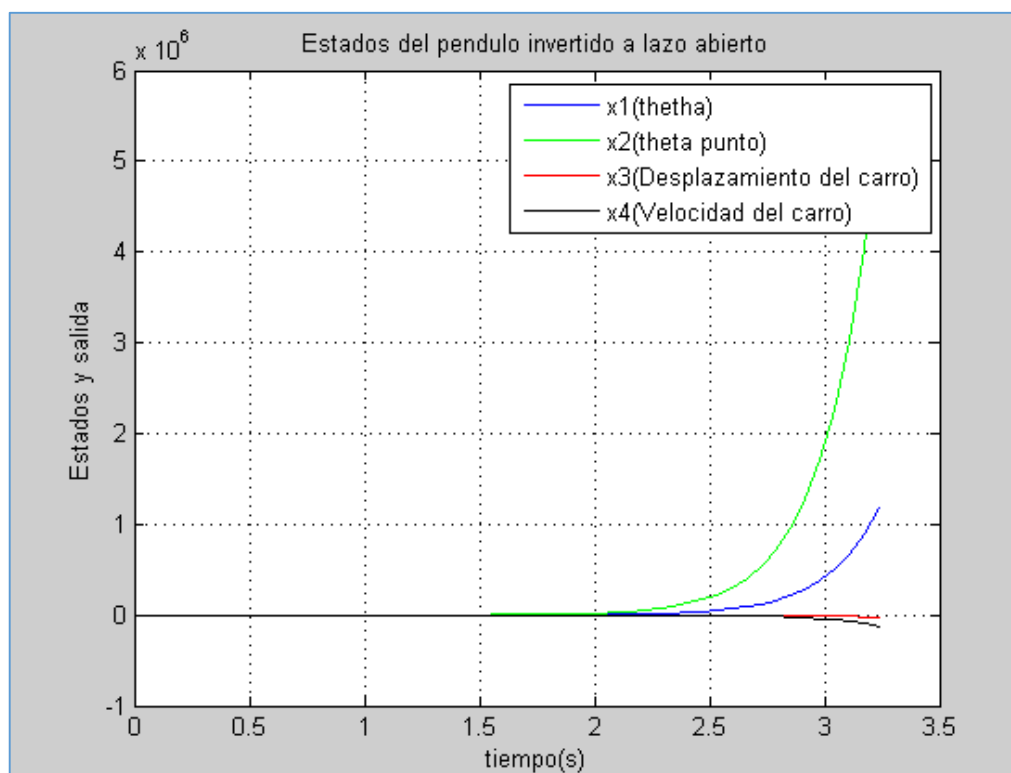


Figura 82. Estados del péndulo invertido a lazo abierto
Fuente: Elaboración propia

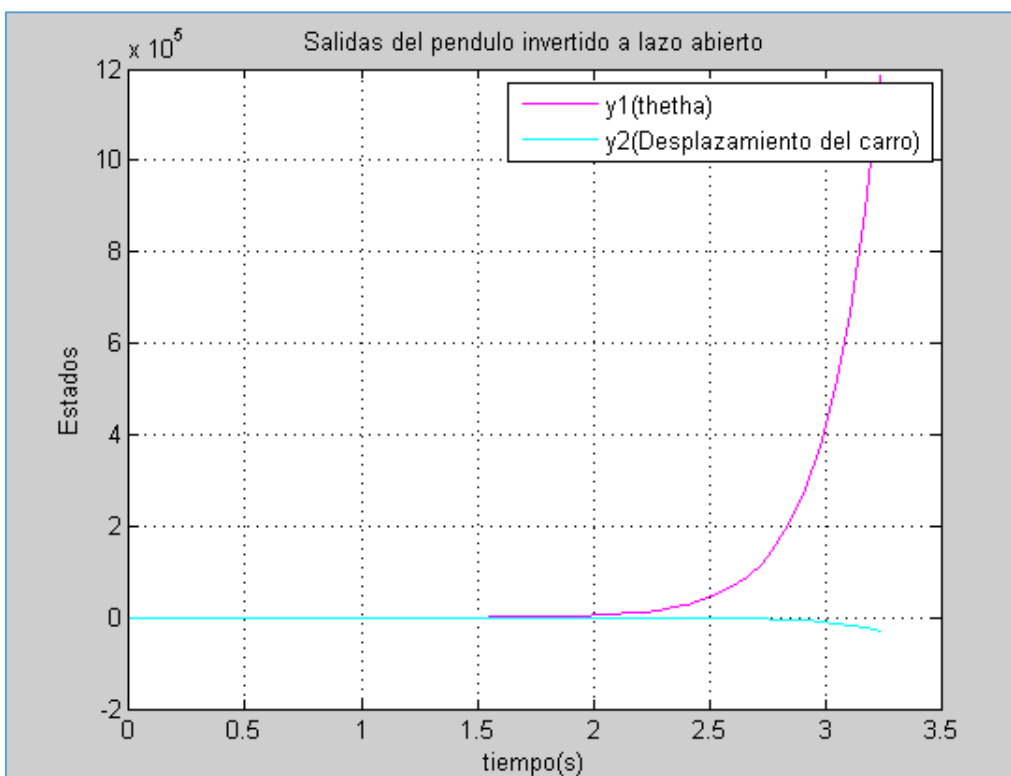


Figura 83. Salida del péndulo invertido a lazo abierto
Fuente: Elaboración propia

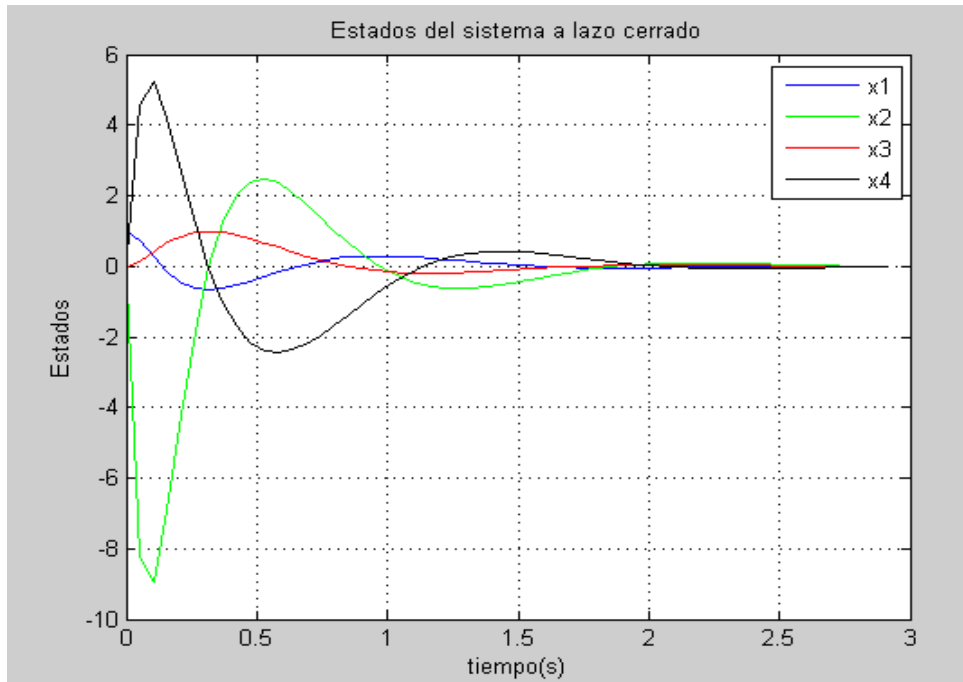


Figura 84. Estados del sistema a lazo cerrado
Fuente: Elaboración propia

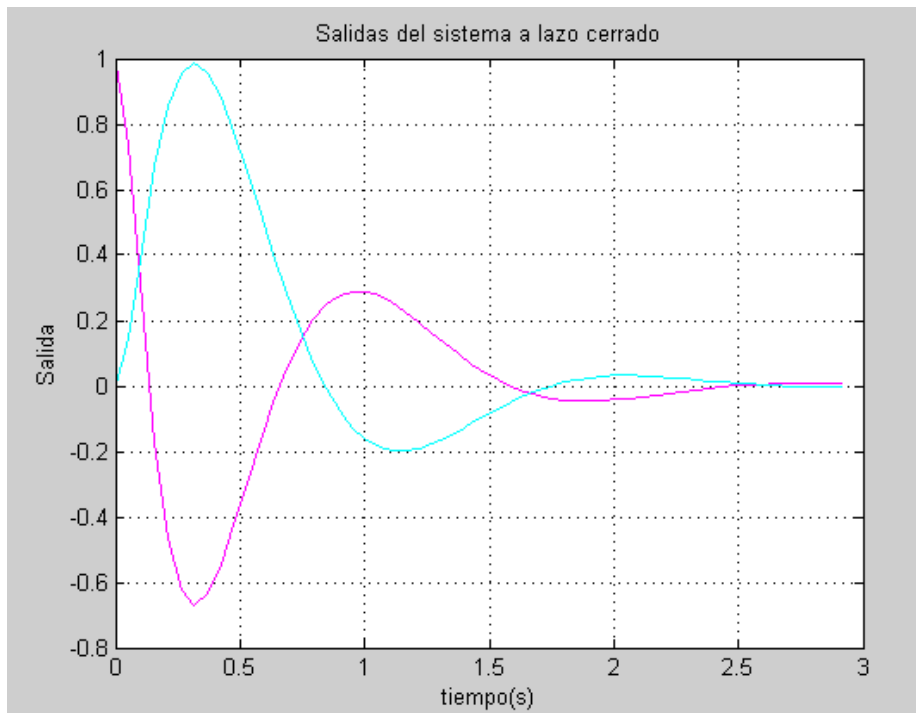


Figura 85. Salida del sistema a lazo cerrado
Fuente: Elaboración propia

Capítulo 5

Resultados de la experimentación

Los laboratorios virtuales y remotos ofrecen flexibilidad de horarios y comodidad a los alumnos, sin embargo, no sería una buena alternativa sino se logran los objetivos educativos planteados.

Por esta razón, los alumnos respondieron una encuesta online, basada en el trabajo de (Fábregas et al., 2011), acerca de su apreciación sobre los laboratorios virtuales, realizados durante el dictado de los cursos mencionados anteriormente para evaluar los resultados de usar esta plataforma.

Dos temas principales agruparon a las preguntas del cuestionario:

Visualización e interactividad: Incluye todo lo relacionado a la plataforma e interfaz creadas para los laboratorios virtuales.

Perspectiva pedagógica: Permite saber la percepción del estudiante frente a esta nueva herramienta y cómo influye en su aprendizaje.

La Tabla 10 muestra las preguntas de la encuesta presentada a 19 alumnos, los cuales fueron escogidos aleatoriamente.

Tabla 10. Encuesta online

Visualización e Interactividad

- La interfaz de la plataforma Web para laboratorios virtuales es adecuada.
- Es incómodo usar la plataforma virtual Web.
- Disfruté utilizando la interfaz Web durante la práctica experimental.
- La interfaz de la plataforma Web virtual es fácilmente accesible y manejable.
- No se requiere mucho esfuerzo para llevar a cabo el laboratorio virtual a través de la interfaz Web.

Perspectiva pedagógica

- Las instrucciones indicadas en la guía de trabajo de laboratorio son suficientes.
- Los laboratorios virtuales me permiten entender mejor el curso.
- Los laboratorios virtuales motivan a aprender más sobre los temas control automático.

Fuente: Elaboración propia.

Los resultados se muestran a continuación:

Para la primera pregunta en visualización e interactividad los resultados arrojaron que un 63% de encuestados está de acuerdo y muy de acuerdo con que la interfaz es adecuada para los laboratorios virtuales mientras que un 5% opinó que no era adecuada (Figura 86).

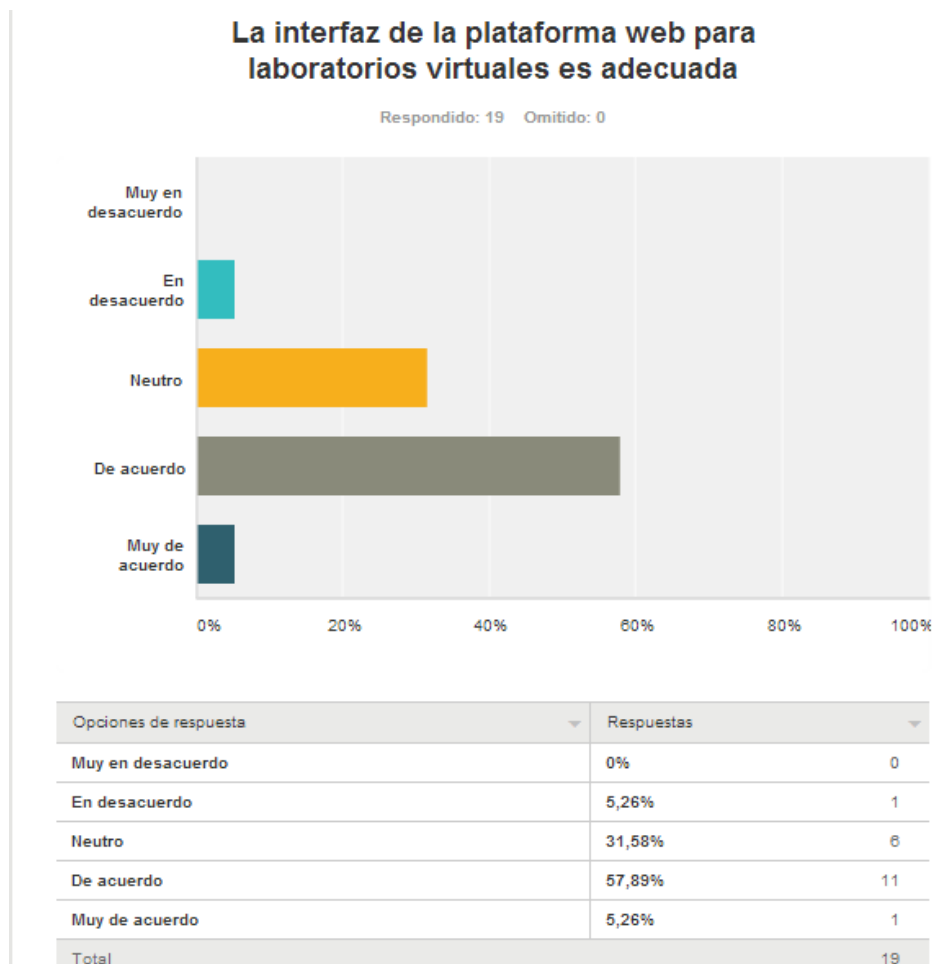


Figura 86. Resultados de la pregunta sobre la interfaz de la plataforma Web
Fuente: Elaboración propia

Sobre la pregunta de si era incómodo usar la plataforma Web los resultados arrojaron que estaban en desacuerdo aproximadamente el 57% de los encuestados y de acuerdo cerca del 10% (Figura 87).

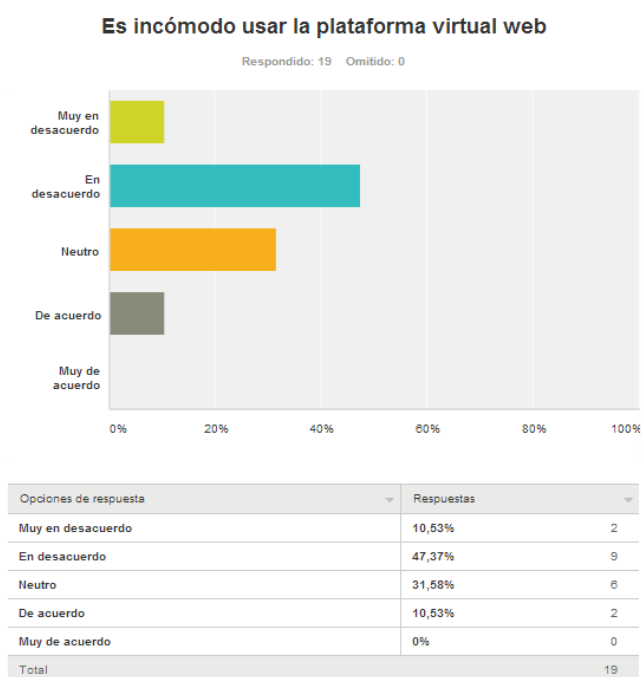


Figura 87. Resultados de la pregunta sobre el uso de la plataforma Web
Fuente: Elaboración propia

Sobre la pregunta de si se disfrutó utilizando la interfaz Web durante la práctica experimental los resultados arrojaron que estaban de acuerdo o muy de acuerdo el 53% de los encuestados y en desacuerdo cerca del 10% (Figura 88).

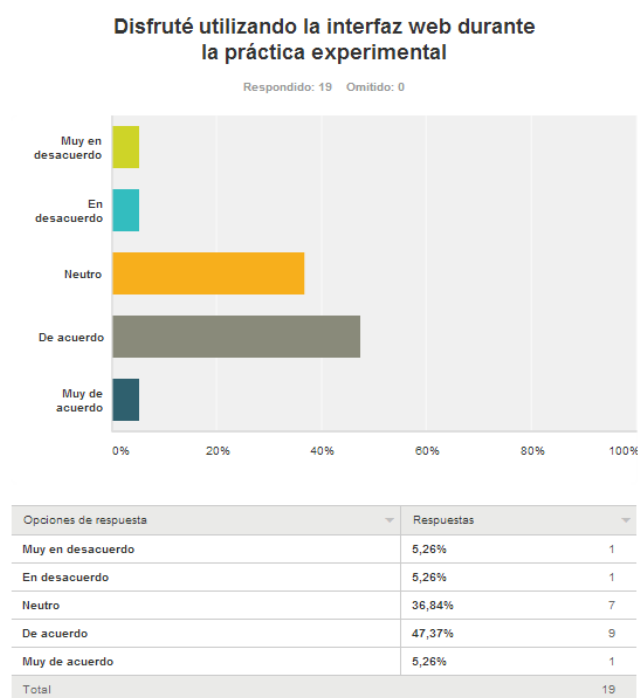


Figura 88. Resultados de la pregunta sobre si se disfruto usando la interfaz Web
Fuente: Elaboración propia

Sobre la pregunta de si la interfaz de la plataforma Web virtual es fácilmente accesible y manejable los resultados arrojaron que estaban de acuerdo o muy de acuerdo el 80% de los encuestados y en desacuerdo cerca del 5% (Figura 89).

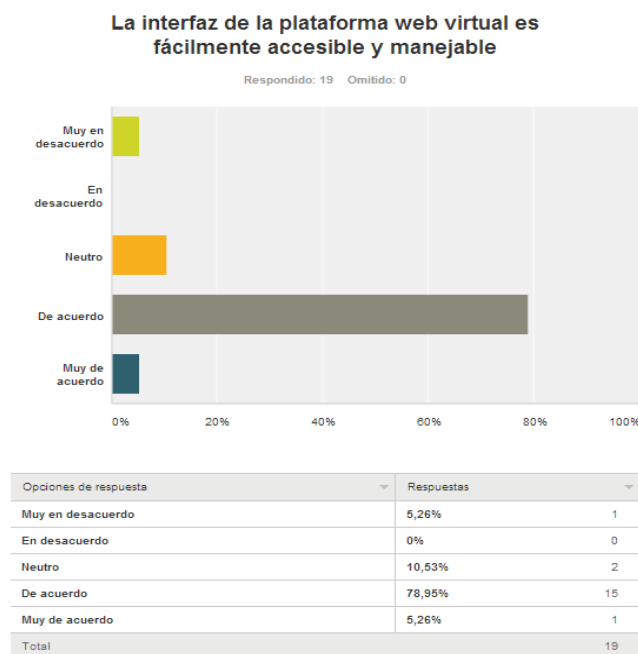


Figura 89. Resultados de la pregunta sobre si la plataforma es accesible y manejable
Fuente: Elaboración propia

Sobre la pregunta de si se requería mucho esfuerzo para llevar a cabo el laboratorio virtual los resultados arrojaron que estaban en desacuerdo cerca del 57% de alumnos encuestados y de acuerdo cerca del 15% (Figura 90).

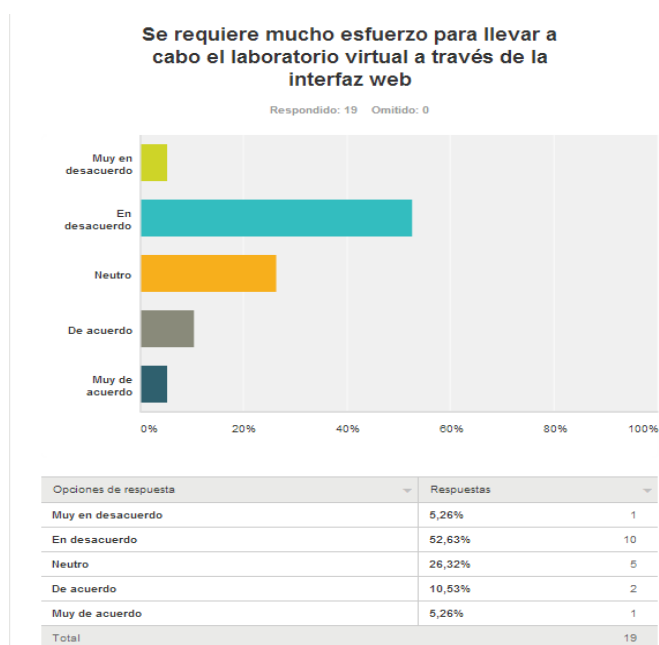


Figura 90. Resultados de la pregunta sobre si se requiere mucho esfuerzo para llevar a cabo el laboratorio
Fuente: Elaboración propia

Sobre la pregunta de si los laboratorios virtuales permiten entender mejor el curso a los alumnos los resultados arrojaron que estaban de acuerdo cerca del 78% de alumnos y en desacuerdo 10% lo cual demuestra que la plataforma Web cumple uno de sus objetivos (Figura 91).

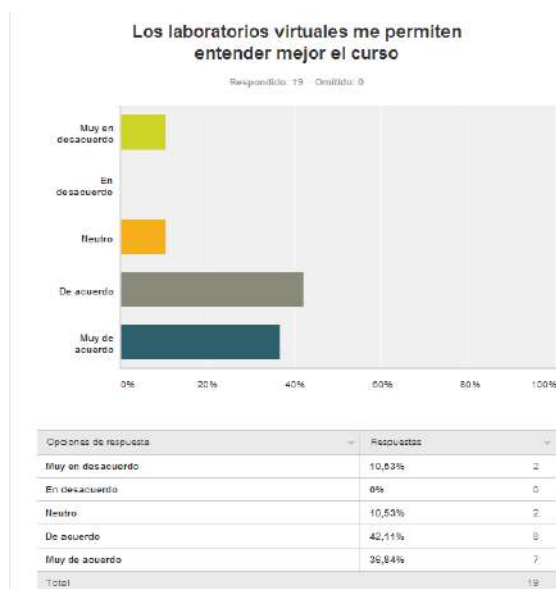


Figura 91. Resultados de la pregunta sobre si los laboratorios virtuales permiten entender mejor el curso

Fuente: Elaboración propia

Sobre la pregunta de si los laboratorios virtuales motivan a aprender más sobre los temas de control automático los resultados arrojaron que estaban de acuerdo cerca del 62% de alumnos y en desacuerdo cerca del 10% (Figura 92).

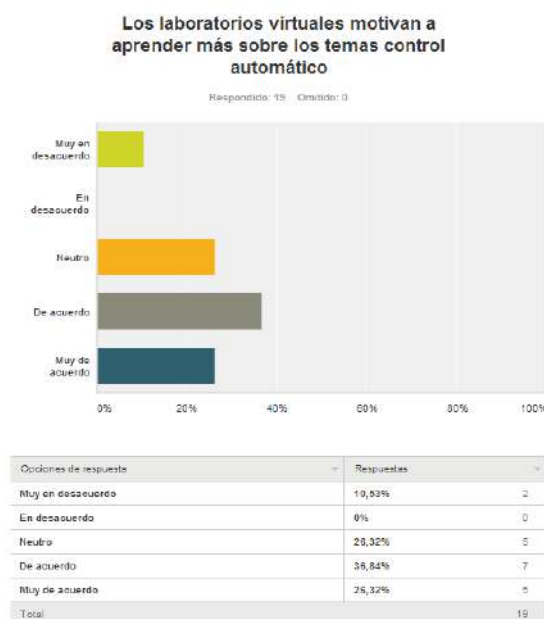


Figura 92. Resultados de la pregunta sobre si los laboratorios virtuales motivan a aprender más sobre el control automático

Fuente: Elaboración propia

La Tabla 11 muestra el resumen de los resultados en porcentajes de los temas, las respuestas a cada pregunta han sido promediadas para cada tema evaluado y clasificado como muy de acuerdo, de acuerdo, neutro, en desacuerdo y muy en desacuerdo.

Los resultados en general indican que alrededor del 65% de los alumnos piensan que la plataforma e interfaz son las adecuadas para un mejor desarrollo de los laboratorios virtuales (visualización e interactividad) mientras que cerca del 10% opinan lo contrario. Con respecto a la perspectiva pedagógica, cerca del 55% encuentran que, con la implementación de los laboratorios virtuales, los conocimientos impartidos en clase se aprovechan mucho más que con los laboratorios tradicionales, no obstante, el 23% está en desacuerdo o muy en desacuerdo.

Tabla 11. Resultados de la encuesta

Tema	Visualización e interactividad	Perspectiva Pedagógica
Muy de acuerdo (%)	6.31	24.56
De acuerdo (%)	56.84	29.83
Neutro(%)	27.37	22.81
En desacuerdo (%)	6.32	12.28
Muy en desacuerdo (%)	3.16	10.53

Fuente: Elaboración propia

Conclusiones

En esta investigación se presenta el desarrollo de una plataforma computacional que permite realizar diferentes laboratorios virtuales a través de comunicación Web utilizando nuevas herramientas como el *toolbox* Matlab Builder JA y el framework Spring de Java.

Se ha investigado sobre las tendencias actuales en cuanto a laboratorios virtuales, encontrando que la mayoría de investigaciones relevantes usan algún software adicional de tipo escritorio por cada cliente para poder realizar un laboratorio virtual.

La propuesta de esta investigación se ha basado en realizar laboratorios virtuales a través de Web, las herramientas elegidas para el desarrollo de la plataforma Web fueron Matlab y Java como lenguaje de programación.

Como primer punto se hizo una descripción de la arquitectura del sistema a desarrollar, el modelado UML y el diseño de base de datos a implementar en la programación de la plataforma virtual. En cuanto a la implementación de la plataforma esta tuvo como requerimientos principales adquirir y enviar datos a un proceso simulado permitiendo el acceso a los usuarios con interfaces de alta interactividad, también permitir la realización de diferentes laboratorios.

El desarrollo de estos laboratorios se hizo primero en código Matlab y después fueron convertidos a Java a través del *toolbox* Matlab Builder JA para que puedan ser utilizados en la programación de Java para Web con la finalidad de que los usuarios no necesiten tener ningún software adicional instalado y puedan realizar el laboratorio sólo a través de un navegador Web.

Para la programación Web se investigó y aprendió el patrón de programación Spring de Java, conociendo sus ventajas y funcionalidades, así como todos los patrones necesarios para hacer de la aplicación desarrollada más robusta y confiable en cuanto a términos de tiempo de respuesta y concurrencia.

Finalmente, estos aspectos fueron conjuntados en una interfaz gráfica con capacidad de interacción con el usuario.

Operativamente la plataforma virtual satisfizo las expectativas, se logró una continua comunicación entre los usuarios de la plataforma y los servidores instalados en el

Laboratorio de Sistemas Automáticos de Control, de esta manera su factibilidad de desarrollo quedó demostrada.

La plataforma fue probada por los alumnos del curso de Sistemas de Control Automático y Control Industrial de la Facultad de Ingeniería de la Universidad de Piura, Perú, ofreciéndoles a los estudiantes una alta flexibilidad de horarios para hacer sus laboratorios virtuales. Esta solución permitió a los usuarios practicar desde casa o donde quiera que se encuentren.

Los resultados obtenidos en las encuestas sobre la realización de estas prácticas experimentales son consistentes con los objetivos de esta tesis, se logró que los alumnos tengan la percepción de que con los laboratorios virtuales entiendan mejor los cursos, mejorando la calidad de la enseñanza, al mismo tiempo se pudo hacer que el alumno tenga una participación activa en el laboratorio sin tener la limitante de los horarios o de algún lugar.

Durante el desarrollo de esta tesis también se tuvo la oportunidad de comprobar la relevancia de estas nuevas tecnologías en la industria desarrollando una aplicación empresarial para la gestión de una línea de producción para el pesado de uvas aprovechando el conocimiento adquirido acerca del *framework* Spring de Java, esta aplicación le ha dado muy buenos resultados a la empresa al aumentar su productividad a más del doble. También es importante resaltar que durante los estudios de esta maestría se han realizado dos publicaciones en eventos internacionales, un artículo científico presentado en el Congreso Latinoamericano de Control Automático CLCA 2014, en Cancún, México y otro presentado en el Congreso Salesiano de Ciencia, Tecnología e Innovación para la Sociedad, CITIS 2015, en Guayaquil, Ecuador.

Se puede concluir que se han tenido repercusiones en dos ámbitos: primero en el sector educativo, al generar una nueva tecnología para el aprendizaje, y en el campo industrial, porque usando estas tecnologías se logró mejorar las condiciones de supervisión, operación y control de los procesos industriales en este caso específico el pesado de uvas para la exportación, con eso se ha comprobado la relevancia del desarrollo de estas plataformas virtuales para la automatización de procesos, lo cual es un punto muy importante para la ingeniería en la actualidad.

En lo que corresponde al trabajo futuro se puede probar la plataforma para otros cursos de la facultad de Ingeniería para que puedan realizar prácticas y exámenes virtuales. También se puede aprovechar la tecnología desarrollada para implementar ya no sólo laboratorios virtuales sino también laboratorios remotos aprovechando los módulos de plantas industriales que se tienen en el laboratorio de Sistemas Automáticos de Control incluyendo en la investigación tecnologías de comunicación industriales como OPC y ModBus que son importantes en las industrias de hoy.

Bibliografía

- Ahmed, M., Krishna K., Interactive e-learning through Second Life with Blackboard Technology, *Procedia - Social and Behavioral Sciences*, Volume 176, 20 February 2015, Pages 891-897, ISSN 1877-0428.
- Andújar Márquez, J.M., Mateo Sanguino, T.J. Diseño de Laboratorios Virtuales y/o Remotos. Un Caso Práctico, *Revista Iberoamericana de Automática e Informática Industrial RIAI*, Volume 7, Issue 1, January 2010, Pages 64-72, ISSN 1697-7912, [http://dx.doi.org/10.1016/S1697-7912\(10\)70009-1](http://dx.doi.org/10.1016/S1697-7912(10)70009-1).
- Arauco Moreno, E. Los Marcos de Trabajo Empresariales [en línea], Lima, 2012, [fecha de consulta: Julio 2015]. Disponible en <http://erick-arauco.blogspot.pe/>.
- Barrios, A., Panche, S., Duque, M., Grisales, V.H., Prieto, F., Villa, J.L., Chevrel, P., y Canu, M. A multi-user remote academic laboratory system. Original Research Article *Computers and Education*, Vol. 62, 2013, 111-122.
- Booch, G., Rumbaugh, J., Jacobson, I. *The Unified Modeling Language User Manual*. Addison-Wesley, 1999.
- Booch, G. *Object-oriented analysis and design with applications*. Benjamin/ Cummings Pub. Co., Redwood City, Calif., 1994.
- Calvo, I., Zulueta, E., Oterino, F., y López, J.M. A remote laboratory for a basic course on control engineering. *International Journal of Online Engineering*, Vol. 5, 2009, 8–13.
- Candelas, F.A., Torres, F., Gil, P., Ortiz, F., Puente, S. y Pomares, J. (2004) Laboratorio Virtual Remoto para Robótica y Evaluación de su impacto en la Docencia. *Red Iberoamericana de Automática e Informática Industrial*, 1(2), 49 – 57.
- Casini, M., Prattichizzo, D., y Vicino, A. Operating remote laboratories through a bootable device. *International Journal of Online Engineering*, Vol. 54, 2006, 3134-3140.
- Caputi, V., Garrido, A. Student-oriented planning of e-learning contents for Moodle, *Journal of Network and Computer Applications*, Volume 53, July 2015, Pages 115-127, ISSN 1084-8045.

- Conrado, M., Vito, L.D., Ramos, H., and Saliga, J. Hardware and software platform for adawan remote laboratory. *Measurement*, 45, 2012, 795 - 807.
- Dormido, S. Sánchez, J., Vargas, H., Dormido-Canto, S., Dormido, R., Duro, N., Farias, G., Canto, y Esquembre, F. (2007a). Análisis, desarrollo y publicación de laboratorios virtuales y remotos para la enseñanza de la automática, II Congreso Español de Informática: Simposio EIWISA, pp. 1-6.
- Egan, D., Zikopoulos, P., Rogers, C., Burlington, S. Chapter 7 - MySQL on Linux, In *DBAs Guide to Databases Under Linux*, 2000, Pages 283-309, ISBN 9781928994046.
- Esquembre, F. Easy Java Simulations: a software tool to create scientific simulations in Java, *Computer Physics Communications*, Volume 156, Issue 2, 1 January 2004, Pages 199-204, ISSN 0010-4655.
- Farías, G., Esquembre, F., Sanchez, J., Dormido, S., Vargas, H., Dormido-Canto, S., Dormido, R., and N. Duro, M.C. Desarrollo de laboratorios virtuales, interactivos y remotos utilizando easy java simulations y modelos simulink. *Proceedings XII Latin-American Congress on Automatic Control*, 2006.
- Fábregas, E., Farías, G., Dormido-Canto, S., Dormido, S., and Esquembre, F. Developing a remote laboratory for engineering education. *Computer and Education*, 57, 2011, 1686 - 1697.
- Gardel, A., Bravo, I., Galilea, J., and Del Toro, P. Remote automation laboratory using a cluster of virtual machines. *Industrial Electronics, IEEE Transactions on*, 57, 2010, 3276-3283.
- Guzmán J.L., Domínguez, M., Berenguel, M., Fuertes, J.J., Rodríguez, F., Reguera, P. Entornos de experimentación para la Enseñanza de Conceptos Básicos de Modelado y Control, *Revista Iberoamericana de Automática e Informática Industrial RIAI*, Volume 7, Issue 1, January 2010, Pages 10-22, ISSN 1697-7912.
- Jacobson, I., Christerson, M., Jonsson, P. and Overgaard, G. *Object- Oriented Software Engineering - A Use Case Driven Approach*. Addison Wesley, 1992.
- Jin H. S., Myoung H. K., An analysis of the optimal number of servers in distributed client/server environments, *Decision Support Systems*, Volume 36, Issue 3, January 2004, Pages 297-312, ISSN 0167-9236.
- Luna-Moreno, D., Espinosa Sánchez, Y.M., Ponce de León, Y.R., Arias, E. Noé, Campos, G. Virtual instrumentation in LabVIEW for multiple optical characterizations on the same opto-mechanical system, *Optik - International Journal for Light and Electron Optics*, Volume 126, Issue 19, October 2015, Pages 1923-1929, ISSN 0030-4026.
- Mahmoud, M., Sabih, M., Elshafei, M. Using OPC technology to support the study of advanced process control, *ISA Transactions*, Volume 55, March 2015, Pages 155-167, ISSN 0019-0578.

- Molina Jordá, J.M. Virtual Tools: Virtual Laboratories for Experimental science – An Experience with VCL Tool, *Procedia - Social and Behavioral Sciences*, Volume 106, 10 December 2013, Pages 3355-3365, ISSN 1877-0428.
- Orduña, P., Rodríguez-Gil, L., Angulo, I., Dziabenko, O., López-de Ipina, D., and García-Zubia, J. Exploring students collaboration in remote laboratory infrastructures. In *Remote Engineering and Virtual Instrumentation (REV)*, 2012, 9th International Conference on, 1-5.
- Perutka K., Zaoral, T. Multimedia Teaching Aid for Students of Basics of Control Theory in MATLAB and SIMULINK, *Procedia Engineering*, Volume 100, 2015, Pages 150-158, ISSN 1877-7058.
- Pop, D., Altar, A. Designing an MVC Model for Rapid Web Application Development, *Procedia Engineering*, Volume 69, 2014, Pages 1172-1179, ISSN 1877-7058.
- Praehofer, H. Sametinger, J., Stritzinger, A. Concepts and architecture of a simulation framework based on the JavaBeans component model, *Future Generation Computer Systems*, Volume 17, Issue 5, March 2001, Pages 539-559, ISSN 0167-739X.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. *Object-Oriented Modeling and Design*. Prentice Hall, 1991.
- Salman, N., Kjell, I., Zaili, Y. Towards Effective Training for Process and Maritime Industries, *Procedia Manufacturing*, Volume 3, 2015, Pages 1519-1526, ISSN 2351-9789.
- Selic. B. *Models, Software Models and UML*, pages 1 – 15. Kluwer Academic Publishers, 2003.
- Sivakumar, S., Robertson, W., Artimy, M. & Aslam, N (2005). *A Web-Based Remote Interactive Laboratory for Internetworking Education*
- Vicente, A., Muñoz, I, Galileo, J. & Del Toro, P. (2010). Remote Automation Laboratory Using a Cluster of Virtual Machines. *Industrial Electronics*, Vol 57, pp 3276-3283
- Walls Craig. *Spring in Action*, Third Edition, June 2011, ISBN 9781935182351, 424 pages.
- Xiaocong F., Chapter 12 - Software Architectures for Real-Time Embedded Systems, In *Real-Time Embedded Systems*, edited by Xiaocong Fan, Newnes, Oxford, 2015, Pages 303-338, ISBN 9780128015070.
- Zschaler, S., Demuth, B., Schmitz, L. Salespoint: A Java framework for teaching object-oriented software development, *Science of Computer Programming*, Volume 79, 1 January 2014, Pages 189-203, ISSN 0167-6423.

Anexos

Anexo A. Artículo científico presentado en el Congreso Latinoamericano de Control Automático CLCA 2014, Cancún, México

Memorias del XVI Congreso Latinoamericano
de Control Automático, CLCA 2014
Octubre 14-17, 2014. Cancún, Quintana Roo, México

Laboratorios Virtuales y Remotos para la Experimentación

W. Ipanaqué* I. Belupú* J. Valdiviezo* G. Vásquez*

* Departamento de Electrónica y Automática, Facultad de Ingeniería,
Universidad de Piura, Av. Ramón Mujica 131, Urb. San Eduardo,
Piura, Perú. (e-mail: william.ipanaque@udep.pe).

Resumen: Se describe el proceso para el desarrollo de laboratorios virtuales y remotos usando como plataforma base diversos software y hardware. Para el caso de los laboratorios virtuales se aplica el *toolbox* Matlab Builder JA y el *framework* Java EE Web. El usuario podrá simular un proceso desde una plataforma web sin la necesidad de contar con Matlab en su ordenador. Para los laboratorios remotos se utiliza el software Labview para crear una interfaz web que permite al usuario conectarse a través de un servidor OPC a un módulo físico para controlarlo.

Keywords: Software, Automatización, procesos, simulación, laboratorios, aplicaciones de computadoras.

1. INTRODUCCIÓN

La realización de prácticas experimentales en la automatización son de suma importancia, ya que beneficia la asimilación de los conceptos teóricos de una mejor manera; estas prácticas permiten resolver futuros problemas de previsión, optimización de tiempos, capacitación, operación, seguridad, flexibilidad en una planta.

Para lograr desarrollar plataformas experimentales se aprovechan los avances en las tecnologías de la información y la comunicación (TIC) (Farias et al., 2006), resaltando el impacto positivo que éstas logran en la enseñanza del control automático, es así que empleando el desarrollo de estas nuevas tecnologías se implementan los laboratorios remotos (Corrado et al., 2012) (Casini et al., 2007) y virtuales (Barrios et al., 2013).

Los laboratorios remotos conectan un proceso real a distancia con un operario mediante un interfaz de alta interactividad (Orduna et al., 2012), mientras que los laboratorios virtuales interfazan simuladores de procesos con un supervisor emulando una planta real. Como ejemplo del desarrollo de estas plataformas podemos citar: a) (Fabregas et al., 2011) que utiliza EJS (*Easy Java Simulation*), Simulink y JIM server permitiendo ejecutar la simulación de forma remota controlando un sistema *Ball and Hoop* (Farias et al., 2006).

Nuestra propuesta en cuanto a laboratorios virtuales es diferente a las alternativas antes mencionadas, en este trabajo se usa el *toolbox* Matlab Builder JA de Matlab para convertir el código de simulación hecho en Matlab a código Java, y así realizar las simulaciones del lado cliente, sin la necesidad de tener instalado el Matlab en su computador a través de una plataforma web desarrollada bajo la programación de Java para web; no se ha encontrado

artículos que usen la combinación de estas herramientas como alternativa a los laboratorios virtuales.

En cuanto a los laboratorios remotos, se ha desarrollado una plataforma web en Labview utilizando la opción que brinda el software de convertirse en un servidor web, y poder acceder a ella desde cualquier punto remoto.

La estructura del presente artículo es la siguiente: la sección 2 resume la descripción del sistema propuesto, en la sección 3 se describe las herramientas utilizadas para el desarrollo de los laboratorios virtuales y su funcionamiento. En el apartado 4 se detalla el desarrollo de la plataforma para laboratorios remotos y su funcionamiento. En el apartado 5 se resume una evaluación pedagógica de la plataforma, señalando finalmente las conclusiones.

2. DESCRIPCIÓN DEL SISTEMA

2.1 Arquitectura del Sistema

La Arquitectura de un sistema es la suma de elementos computacionales que siguen diferentes patrones con el fin de unir distintos servicios informáticos; estos elementos se comunican por medio de mensajes de solicitud y mensajes de entrega para conseguir el intercambio de la información.

Existe más de un estilo arquitectónico: cliente-servidor, organización o modelos por capas, filtros, sistemas jerárquicos por niveles, etc.; en este trabajo se ha usado la arquitectura cliente-servidor, en la figura 1 se puede apreciar la comunicación que se da entre dos elementos: uno o varios clientes (usuarios finales); que pueden ser computadoras de propósito general que determinan los requerimientos de información que se solicitan; y uno o varios servidores que son computadoras con mayores prestaciones (procesador, memoria, disco duro, etc.), pues se encargan de la lógica del sistema, además deben tener la capacidad de

* Con el apoyo de «Fondos para la Innovación, Ciencia y Tecnología (FINCyT)», Perú, a través del proyecto 214-FINCyT-IA-2013.

CLCA 2014
Octubre 14-17, 2014. Cancún, Quintana Roo, México

dar respuesta a varios clientes, procesando las peticiones recibidas.

Algunas características que resumen la noción de arquitectura cliente - servidor son (Cardel et al., 2010):

- El cliente interactúa con el usuario (interfaz) y el servidor lo hace con recursos compartidos (comunicación directa con el proceso).
- Posibilidad de compartir recursos lógicos y físicos: varios clientes hacen uso de un solo servidor.
- El cliente es emisor cuando solicita y el servidor es el receptor, siendo la función del último de carácter pasivo, pues solo espera las solicitudes del cliente. Inmediatamente, cambian de posición, el servidor envía lo requerido (emisor) y el cliente lo recibe (receptor).
- Las prestaciones necesarias de una computadora son diferentes si actuará como cliente o como servidor. Funciones determinadas van dirigidas a cada uno con el fin de lograr un mejor aprovechamiento del ancho de banda de red.
- Se aplica el concepto de escalabilidad horizontal (añadir clientes) y vertical (mejorar características del servidor o añadir más servidores).
- Permite que sistemas distintos puedan compartir información entre sí, logrando así una fácil integración. Diferentes procesos pueden llevarse a cabo en una misma computadora o en varias que se encuentran distribuidas geográficamente a lo largo de una misma red.
- Exigen una verificación tanto en el cliente como en el servidor para evitar riesgos en la seguridad del esquema cliente - servidor.

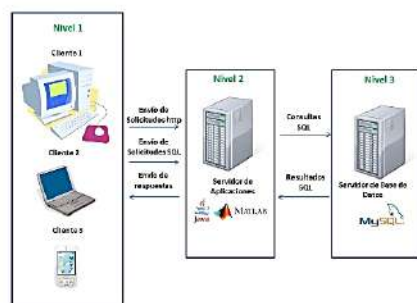


Figura 1. Arquitectura Cliente-Servidor llevada a la plataforma virtual

3. DESARROLLO DE LA PLATAFORMA DE LABORATORIOS VIRTUALES

3.1 Matlab Builder JA

Matlab *Builder JA* es una herramienta que permite crear clases Java a partir del código escrito en Matlab. Estas clases de Java son integradas en las aplicaciones de Java a través del compilador *Matlab Compiler Runtime* (MCR), desplegándose gratuitamente en las computadoras de escritorio o servidores web que no necesariamente tienen

Matlab instalado. Es decir se crean componentes desplegables que hacen cálculos basados en Matlab, visualizaciones e interfaces de usuario accesibles a los usuarios finales de los programas Java. Cuando el programa Java se implementa en la red, varios usuarios pueden acceder a él a través de un navegador web. Las clases Java creadas con *Matlab Builder JA* son portables como se muestra en la figura 2.

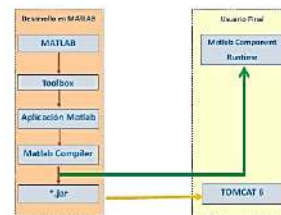


Figura 2. Funcionamiento del toolbox Matlab Builder JA

3.2 Aplicaciones Web con Java EE

Java Enterprise Edition (Java EE) es un framework de desarrollo estándar (ver figura 3) muy popular para construir y desplegar aplicaciones web en Java. Como característica principal tiene dos tecnologías claves para la construcción de los componentes de las aplicaciones web: los servlets y los archivos jsp (Java Server Page).

Los servlets son clases de Java que permiten realizar la lógica de la aplicación para el manejo de una petición web, después de procesar la petición, devuelven un objeto java con la respuesta, la cual contiene información que se muestra en el lado cliente.

Las páginas jsp tienen la ventaja de facilitar la combinación del código html con la lógica escrita en Java. Después de su compilación, se ejecutan como un servlet convirtiéndose en clases java, tomando también una petición web y devolviendo un objeto Java que representa la página de respuesta (Lee and Jung, 2007).

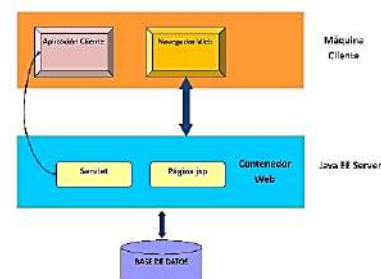


Figura 3. Framework Java EE

3.3 Desarrollo de las Simulaciones en Matlab

Los códigos para implementar los procesos para simulación se deben desarrollar en código .m de Matlab y ser

CLCA 2014
Octubre 14-17, 2014. Cancún, Quintana Roo, México

guardados en archivos con extensión .mat, después se debe crear el archivo independiente de java (archivo .jar) con el *toolbox* Java Builder JA siguiendo los siguientes pasos:

- Escribir el comando “*deploytool*” en la ventana de comandos de matlab para abrir el *toolbox* Matlab Builder JA.
- Se abre la ventana *Deployment Tool*.
- Se procede a crear un nuevo proyecto: File - New Deployment Project.
- Se elige la opción *Matlab Builder JA* y se le asigna un nombre al proyecto.
- El siguiente paso es asignar los archivos .m al proyecto con la opción *Add File*.
- Una vez asignados los archivos .m se compila el proyecto con la opción *Build the Project*.
- Al finalizar se obtiene el código .m de matlab encapsulado en un paquete .jar para que se pueda usar en la programación Web en Java.

3.4 Desarrollo de la plataforma web con Java EE

Materiales y Métodos La plataforma virtual está desarrollada bajo el compilador de Sun Microsystem (JDK1.6.0) para servidores de 64 bits instalado en el sistema operativo Windows Server 2008 para 64 bit. Esto no quiere decir que no se pueda implementar en Servidores que trabajen bajo otros sistemas operativos, de hecho la característica multiplataforma de Java es una de sus mayores ventajas. Para la implementación del entorno de desarrollo se ha utilizado el IDE NetBeans 7.3, el cual es un entorno integrado para programadores que facilita la escritura, compilación y ejecución de programas en Java. El IDE NetBeans está escrito en Java y es un producto libre sin restricciones de uso.

Funcionamiento El funcionamiento de esta plataforma ha sido probado con los alumnos de los cursos de Sistemas Automáticos de Control y Control Industrial de la Facultad de Ingeniería de la Universidad de Piura.

- **Iniciar Sesión:** La plataforma virtual tiene niveles de seguridad, cada alumno tiene un usuario y clave asignado.
- **Test de conocimientos:** Para desarrollar el laboratorio virtual el alumno debe tener un mínimo de conocimientos sobre el tema del laboratorio a realizar, es por eso que después de iniciar la sesión se les pide solucionar un *test* o cuestionario para validar si están capacitados para desarrollar el laboratorio virtual.



Figura 4. Pantalla de inicio de sesión

- Si el alumno responde de forma correcta más del 80% del cuestionario, puede realizar el laboratorio de lo contrario se le brindará una segunda oportunidad con un *test* diferente, si no logra resolver el segundo *test*, no podrá realizar el laboratorio virtual.
- Los alumnos que aprobaron el *test* cargarán la página del laboratorio con las respectivas cajas de texto para ingresar la información del proceso y empezar la simulación.

Como ejemplo de los laboratorios virtuales implementados podemos citar el de control en variables de estados del sistema péndulo invertido. Inicialmente el usuario debe linealizar el modelo matemático del péndulo invertido a un modelo en espacio de estados para obtener las matrices de estado. El usuario debe ingresar a la plataforma web las matrices para graficar los estados y la salida del sistema a lazo abierto, calculando el determinante de las matrices de observabilidad y controlabilidad; de esta manera el usuario puede evaluar si el sistema es controlable y observable parcial o total. Así mismo, para la sintonización del controlador proporcional en espacio de estados, el usuario debe ingresar los polos del controlador y los polos del observador. Tomando en cuenta los polos, la plataforma web grafica los estados y las salidas a lazo cerrado, las figuras 5,6,7,8, muestran las gráficas obtenidas.

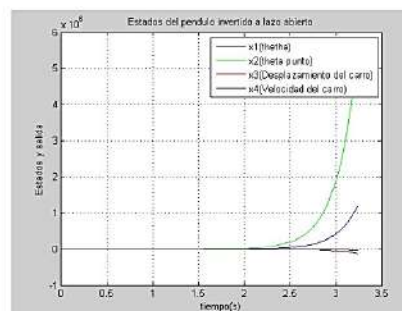


Figura 5. Estados del péndulo invertido a lazo abierto

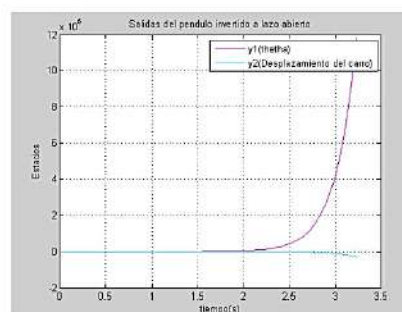


Figura 6. Salida del péndulo invertido a lazo abierto

Las ventajas que ofrece esta plataforma web para el análisis de control en espacio de estados son: permite de manera

CLCA 2014
Octubre 14-17, 2014. Cancún, Quintana Roo, México

inmediata analizar la controlabilidad y observabilidad del sistema, calcula de manera precisa la ganancia proporcional para el control en espacio de estados y permite sacar conclusiones gráficas con respecto a la estabilidad y asignación de polos.

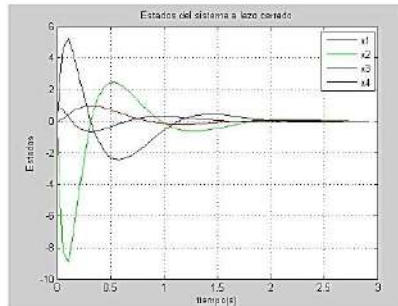


Figura 7. Estados del sistema a lazo cerrado

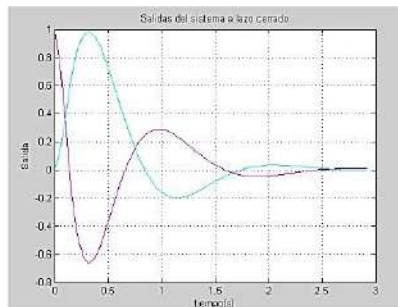


Figura 8. Salidas del sistema a lazo cerrado

4. LABORATORIOS REMOTOS

La visualización y la interactividad son dos aspectos muy importantes para realización de los laboratorios remotos, a medida que se mejore estas dos características los laboratorios remotos serán de mejor calidad para propósitos pedagógicos (Parias et al., 2006).

Teniendo en cuenta las características antes mencionadas se ha desarrollado un laboratorio remoto para el control de un motor de corriente alterna (figura 9) que tiene como objetivo principal que el estudiante pueda identificar un proceso, así como la correcta sintonización de un PID, para esto el PLC ha sido programado tanto a lazo abierto como a lazo cerrado.

Para la elaboración del laboratorio remoto son necesarios los siguientes elementos:

- PLC: Modelo: PLC V130 - 33 - TA24
- Variador de frecuencia GS1.
- Motor Sieber Tipo LS71/1.
- Fuente de alimentación 24VDC National Instruments FP-PS-4.

- Un circuito integrado.
- Un encoder.



Figura 9. Módulo experimental control de un motor de corriente alterna

4.1 Software Utilizado

Para la conexión y el control remoto se utilizan tres software: El Visilogic, para la programación del PLC (del fabricante Unitronics), Labview, para la interfaz gráfica, y el UniOPC para la comunicación entre el PLC y el Labview.

Labview consta de un panel frontal (elaboración de la interfaz) y un diagrama de bloques (programación a través de bloques, código fuente) como se muestra en la figura 10, esto permite crear una interfaz amigable para la obtención de datos del proceso, así como apreciar la variación del mismo ante posibles cambios en su entrada.

El enlace de datos entre el Labview y el PLC se consigue al colocar las direcciones de las memorias internas del PLC en los bloques adecuados donde se leen y escriben las entradas y salidas del proceso.

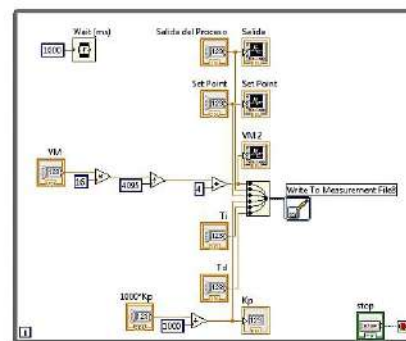


Figura 10. Diagrama de bloques a lazo cerrado en Labview

CLCA 2014
Octubre 14-17, 2014. Cancún, Quintana Roo, México

UniOPC El software Labview se conectará con el PLC (Fabregas et al., 2011) mediante el UniOPC (servidor OPC del fabricante Unitronics). Para el funcionamiento del servidor OPC, se configura el canal de comunicación (con el IP y puerto usados), después se agrega este canal con el nombre del PLC; finalmente se hace clic en el botón *RUN* para verificar si la conexión ha sido establecida.

4.2 Descripción de la Interfaz

La interfaz como se ve en la figura 11 consta una imagen proyectada desde una cámara web, botones de control y diferentes pestañas, donde se encuentran las acciones a seguir durante el laboratorio remoto, por ejemplo una de ellas es la del control a lazo cerrado, el cual consiste en campos donde se enviarán los datos para el control (k_p , T_i , T_d) del proceso y las gráficas obtenidas (*set point*, entrada y salida del proceso).

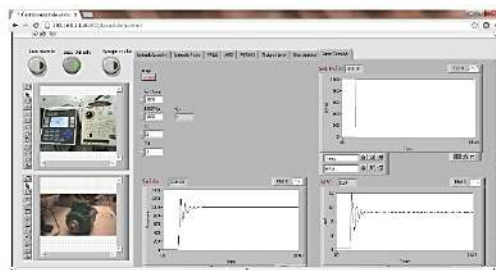


Figura 11. Interfaz del laboratorio remoto

4.3 Experimentación

El lazo abierto y cerrado del proceso están programados en Labview, éstos dividen en 8 pestañas donde el usuario desarrolla distintos eventos:

- Comprobación de la linealidad o no linealidad del proceso. Así como la obtención de la función transferencia a entrada escalón.
- Respuesta del proceso a entradas diferentes (pulso, senoidal, etc).
- Modificar los datos de una señal PRBS (*Pseudo Random Binary Signal*) que será la entrada al proceso y analizar su respuesta. Luego, realizar una identificación paramétrica usando los modelos ARX (*Auto Regressive with Exogenous input*), ARMAX (*Auto Regressive with Moving Average and Exogenous input*), OE (*Output Error*) y BJ (*Box Jenkins*) al variar el orden de los coeficientes de cada modelo.
- Apreciar el efecto al variar la parte proporcional, acción integral y derivativa de un controlador PID. El sistema cuenta con las debidas restricciones en los cuadros de texto que no permitirán al alumno introducir valores indebidos al proceso.

Para la identificación del proceso, Labview genera la señal PRBS, para leer y ordenar datos guardados de la PRBS se ha utilizado el Matlab (se enlazan mediante una función de Labview denominada Matlab Script). Con los

datos obtenidos se procede a la obtención de la función transferencia mediante los modelos paramétricos ARX, ARMAX, OE y BJ (figura 12). Finalmente se calculan los correctos parámetros de PID para el motor.

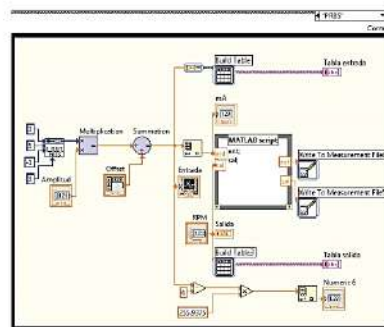


Figura 12. Diagrama de bloques de la pestaña PRBS

Trabajar con Labview facilita la obtención de datos mediante la opción "*Write to Measurement file*" en archivos .lvm los cuales se pueden cambiar a .txt o .xls, estos archivos contienen los datos que se indiquen y se guardarán automáticamente en una ubicación fija.

4.4 Funcionamiento de la Web

Labview tiene la funcionalidad de crear una página web (Peck et al., 2005) a través de la función *Web Publishing Tool*, para ello se debe configurar los datos del servidor web en Labview e indicar el puerto utilizado, activar la opción *Enable Remote Panel Server* y elegir la ubicación de la página web en el lado servidor. En la página web se aprecia la interfaz creada, a ella se puede acceder desde cualquier ordenador sin que en él esté instalado Labview, no obstante necesita ciertos *plugins* para su funcionamiento.

Los alumnos matriculados en el curso, pertenecen a una base de datos donde tienen asignados sus respectivos usuarios y claves para acceder a la plataforma, una de las características de la página web es que pueden acceder varios usuarios a ella, pero sólo uno de ellos puede tener el control del motor. En el lado servidor se puede elegir a quién de los usuarios registrados permitir o negar el manejo del proceso.

5. EVALUACIÓN PEDAGÓGICA

Los laboratorios virtuales y remotos ofrecen flexibilidad de horarios y comodidad a los alumnos, sin embargo no sería una buena alternativa sino se logran los objetivos educativos planteados. Por esta razón, los alumnos respondieron una encuesta online, basada en el trabajo de (Fabregas et al., 2011), acerca de su apreciación sobre los laboratorios virtuales realizados durante el dictado de los cursos de Sistemas Automáticos de Control y Control Industrial de la Universidad de Piura.

Cuadro 1. Resultados expresados en porcentajes de la encuesta online

Tema	Visualización e interactividad	Perspectiva pedagógica
Muy de acuerdo (%)	6.31	24.56
De acuerdo (%)	56.84	29.83
Neutro (%)	27.37	22.81
En desacuerdo (%)	6.32	12.28
Muy en desacuerdo (%)	3.16	10.53

Dos temas principales agruparon a las preguntas del cuestionario:

- Visualización e interactividad: Incluye todo lo relacionado a la plataforma e interfaz creadas para los laboratorios virtuales.
- Perspectiva pedagógica: Permite saber la percepción del estudiante frente a esta nueva herramienta y cómo influye en su aprendizaje.

El cuadro 1 muestra los resultados en porcentajes de los temas, las respuestas a cada pregunta han sido promediadas para cada tema evaluado y clasificado como muy de acuerdo, de acuerdo, neutro, en desacuerdo y muy en desacuerdo. El cuadro 2 muestra las preguntas de la encuesta presentada a 19 alumnos, los cuales fueron escogidos aleatoriamente.

Los resultados indican que alrededor del 65 % de los alumnos piensan que la plataforma e interfaz son las adecuadas para un mejor desarrollo de los laboratorios virtuales (visualización e interactividad) mientras que cerca del 10 % opinan lo contrario. Con respecto a la perspectiva pedagógica, cerca del 55 % encuentran que con la implementación de los laboratorios virtuales, los conocimientos impartidos en clase se aprovechan mucho más que con los laboratorios tradicionales, no obstante el 23 % está en desacuerdo o muy en desacuerdo.

Cuadro 2. Cuestionario de la encuesta online

Visualización e interactividad
- La interfaz de la plataforma web para laboratorios virtuales es adecuada
- Es cómodo usar la plataforma virtual web
- Disfruté utilizando la interfaz web durante la práctica experimental
- La interfaz de la plataforma web virtual es fácilmente accesible y manejable
- No me requiere mucho esfuerzo para llevar a cabo el laboratorio virtual a través de la interfaz web
Perspectiva pedagógica
- Las instrucciones indicadas en la guía de trabajo de laboratorio son suficientes
- Los laboratorios virtuales me permiten entender mejor el curso
- Los laboratorios virtuales motivan a aprender más sobre los temas control automático

6. CONCLUSIONES

En este trabajo se ha presentado el desarrollo de dos plataformas computacionales con características diferentes: La primera plataforma ha sido desarrollada para la realización de laboratorios virtuales, se ha usado herramientas nuevas como el *toolbox* Matlab Builder JA y el *framework* Java EE para implementar esta solución, siendo probada por los alumnos de los cursos de Sistemas Automáticos de Control y Control Industrial de la Facultad de Ingeniería de la Universidad de Piura, ofreciéndoles alta flexibilidad de horarios para realizar sus laboratorios virtuales, esta

solución tiene la ventaja de que los usuarios puedan practicar desde su hogar o en donde estuviesen reduciendo la necesidad de asistir a la Universidad físicamente.

La plataforma para laboratorios remotos ha sido desarrollada utilizando el software Labview, esta plataforma brinda al usuario la posibilidad de acceso al módulo de motor de corriente alterna instalado en el laboratorio de Sistemas Automáticos de Control de la Universidad de Piura. También al igual que la plataforma para laboratorios virtuales ofrece flexibilidad de horarios y espacio físico.

Mediante la encuesta realizada se ha comprobado la relevancia del desarrollo de estas plataformas virtuales y remotas para la automatización de procesos industriales, siendo un punto muy importante para la ingeniería en la actualidad.

REFERENCIAS

- Barrios, A., Panche, S., Duque, M., Grisales, V.H., Prieto, F., Villa, J.L., Chevrel, P., and Canu, M. (2013). A multi-user remote academic laboratory system. *Original Research Article Computers and Education*, 62, 111–122. doi:10.1016/j.compedu.2012.10.011.
- Casini, M., Prattichizzo, D., and Vicino, A. (2007). Operating remote laboratories through a bootable device. *International Journal of Online Engineering*, 54, 3134–3140. doi:10.1109/TIE.2007.907026.
- Corrado, M., Vito, L.D., Ramos, H., and Saliga, J. (2012). Hardware and software platform for advanced remote laboratory. *Measurement*, 45, 795 – 807. doi: 10.1016/j.measurement.2011.12.003.
- Fabregas, E., Farias, G., Dormido-Canto, S., Dormido, S., and Esquembre, F. (2011). Developing a remote laboratory for engineering education. *Computers and Education*, 57, 1686 – 1697. doi: 10.1016/j.compedu.2011.02.015.
- Farias, G., Esquembre, F., Sanchez, J., Dormido, S., Vargas, H., Dormido-Canto, S., Dormido, R., and N. Duro, M.C. (2006). Desarrollo de laboratorios virtuales, interactivos y remotos utilizando easy java simulations y modelos simulink. *Proceedings XII Latin-American Congress on Automatic Control*.
- Gardel, A., Bravo, L., Galilea, J., and del Toro, P. (2010). Remote automation laboratory using a cluster of virtual machines. *Industrial Electronics, IEEE Transactions on*, 57, 3276–3283. doi:10.1109/TIE.2010.2041131.
- Lee, G.H. and Jung, J. (2007). Web framework with java and xml in multi-tiers for productivity. *Future Generation Computer Systems*, 23, 263 – 268. doi: 10.1016/j.future.2006.05.010.
- Orduna, P., Rodríguez-Gil, L., Angulo, I., Dziabenko, O., López-de Ipiña, D., and García-Zubia, J. (2012). Exploring students collaboration in remote laboratory infrastructures. In *Remote Engineering and Virtual Instrumentation (REV), 2012 9th International Conference on*, 1–5. doi:10.1109/REV.2012.6293159.
- Peek, C.S., Crisalle, O.D., Dépraz, S., and Gillet, D. (2005). The virtual control laboratory paradigm: Architectural design requirements and realization through a dc-motor example. *International Journal of Engineering Education*, 21, 1134–1147. doi: 10.1016/j.measurement.2008.09.003.

Anexo B. Artículo científico presentado en el Congreso Salesiano de Ciencia, Tecnología e Innovación para la Sociedad, CITIS 2015, Guayaquil, Ecuador

CONGRESO SALESIANO DE CIENCIA, TECNOLOGÍA E INNOVACIÓN PARA LA SOCIEDAD, CITIS
Guayaquil, 2 y 3 de Diciembre del 2015

Virtual laboratories with Web communication using Matlab and Java Spring Framework (Implementación de laboratorios virtuales a través de comunicación Web usando Matlab y el Framework Spring de Java)

W. Ipanaqué-Alama, I. Belupú-Amaya

Abstract— This article describes the process for the development virtual laboratories for automatic control learning through web pages using as base platform software tools such as Matlab and Java Spring framework. Conducting experimental practices are important in automation area, because the students can understand the theoretical concepts. If students do not have Matlab, it is not a problem because they will be able to do a simulation from the Web browser, using this Web platform in real time and compare results in order to take the best decision. In another hand, this Web platform allow determinate the student's knowledge about a topic with a previous test. The Web platform's architecture used is client-server, which is used to develop enterprise software in real time and makes easy develop the virtual environment. Students can use this Web platform to enhance their knowledge about industrial process and their behavior, in order to solve industrial problems like monitoring, predict, training and industrial safety.

Index Terms— Automation, computer applications, laboratory, processes, real time, simulation, software

I. INTRODUCCIÓN

EL desarrollo de software computacional es importante como herramienta para crear métodos de enseñanza y aprendizaje virtual y remoto en diferentes empresas, universidades, centros tecnológicos [1]. Esta tendencia está relacionada con la creciente demanda del aprendizaje en los últimos años, gracias al auge en el desarrollo de las tecnologías de la información y comunicación (TIC) [2].

El uso de estas herramientas tecnológicas pueden ser implementadas también en la industria para la supervisión de la planta y capacitación del personal [3], con lo cual se puede ahorrar costo, tiempo, mejorar la gestión y supervisión de sus procesos contribuyendo a aumentar la calidad del producto final.

W. Ipanaqué-Alama, Departamento de Electrónica y Automática, Facultad de Ingeniería, Universidad de Piura, Av. Ramón Mugica 131, Urb. San Eduardo, Perú (e-mail: William.ipanaque@udep.pe).

I. Belupú-Amaya, Departamento de Electrónica y Automática, Facultad de Ingeniería, Universidad de Piura, Av. Ramón Mugica 131, Urb. San Eduardo, Perú (e-mail: cesar.belupu@posgrado.udep.edu.pe).

Para lograr desarrollar estas plataformas experimentales se aprovechan los avances en las tecnologías de la información y la comunicación (TIC) [4], resaltando el impacto positivo que los laboratorios remotos [5] [6] y virtuales [7] logran en la enseñanza del control automático.

Los laboratorios remotos conectan un proceso real a distancia con un operario mediante una interfaz de alta interactividad [8], mientras que los laboratorios virtuales interfazan simuladores de procesos con un supervisor emulando una planta real.

El desarrollo de estas plataformas remotas y virtuales son temas de actualidad en la Ingeniería de la automatización [7]; por ejemplo en [9] utilizaron EJS (Easy Java Simulation) [10] y Simulink para controlar un sistema Ball and Hoop, usando el software adicional JIM server [4], para ejecutar la simulación de forma remota.

En la referencia [11] utilizaron la arquitectura cliente – servidor [12] para realizar una plataforma virtual. Este trabajo consistió en usar Labview [13] del lado servidor para adquirir los datos, conectado con el operador remoto que está en el lado cliente a través de tecnología OPC [14], así se lo logró una interacción hombre máquina.

En la referencia [7] realizaron un trabajo parecido al anterior también usando Labview y conectándose al EJS [10] a través del software JIL Server (Java Internet Labview). Con la diferencia de que usan aplicaciones web de código abierto como Moodle [15] y Blackboard [16] del lado cliente para acceder a los experimentos remotos.

Sin embargo En América Latina aún no hay una investigación relevante sobre estos temas, hay muy poco conocimiento y experiencia en marcos de trabajo (framework) para el desarrollo de sistemas computacionales de supervisión y operación remota, virtual y aplicaciones industriales, existen escasas plataformas en tiempo real para aplicaciones en laboratorios virtuales, remotos y en aplicaciones empresariales, es por eso que en este trabajo se presenta el desarrollo de una plataforma Web implementada bajo el framework de programación Spring de Java (framework empresarial) que ayude al aprendizaje virtual y remoto en tiempo real.

Nuestra propuesta en cuanto a laboratorios virtuales es diferente a las alternativas antes mencionadas. En el presente trabajo se usó el toolbox Matlab Builder JA para convertir el código de simulación hecho en Matlab a código Java.

Este código ya convertido pasa a ser una librería de programación, que puede ser usado en cualquier entorno de programación Java ya sea de escritorio o para web. Se decidió realizar una aplicación web para que el usuario final no necesite instalar Matlab o ningún otro software adicional para realizar el laboratorio.

Para el desarrollo de la plataforma web se eligió el framework Spring de Java que permite crear aplicaciones Web empresariales robustas, estables y confiables. De nuestro conocimiento, no se ha usado la combinación de estas herramientas como alternativa a los laboratorios virtuales.

II. ARQUITECTURA DEL SISTEMA

Un sistema informático necesita de un framework de comunicación siguiendo un lenguaje en común, poder enviar y recibir datos entre los elementos; esta estructura es conocida como la arquitectura de un sistema [17].

Existe más de un estilo arquitectónico: cliente-servidor [12], organización o modelos por capas, filtros, sistemas jerárquicos por niveles, etc. En el presente trabajo se ha usado la arquitectura cliente-servidor. La Figura 1 demuestra la comunicación que se da entre dos elementos: uno o varios clientes (usuarios finales); que pueden ser computadoras de propósito general que determinan los requerimientos de información que se solicitan; y uno o varios servidores que son computadoras con mayores prestaciones (procesador, memoria, disco duro, etc.), pues se encargan de la lógica del sistema, además deben tener la capacidad de dar respuesta a varios clientes, procesando las peticiones recibidas.

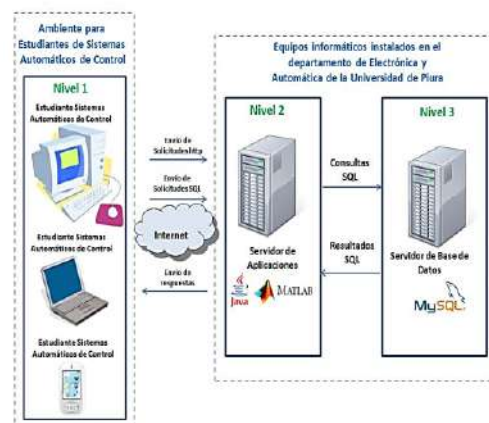


Fig. 1. Arquitectura cliente – servidor llevada a la plataforma Web
Fuente: Elaboración propia

Algunas características que resumen la noción de arquitectura cliente - servidor son [18]:

- Posibilidad de compartir recursos lógicos y físicos: varios clientes hacen uso de un solo servidor.
- El cliente es emisor cuando solicita y el servidor es el receptor, siendo la función del último de carácter pasivo, pues solo espera las solicitudes del cliente. Inmediatamente, cambian de posición, el servidor envía lo requerido (emisor) y el cliente lo recibe (receptor).
- Las prestaciones necesarias de una computadora son diferentes según su rol como cliente o como servidor. Funciones determinadas van dirigidas a cada uno con el fin de lograr un mejor aprovechamiento del ancho de banda de red.
- Se aplica el concepto de escalabilidad horizontal (adicionar clientes) y vertical (mejorar características del servidor o adicionar más servidores).
- Permite que sistemas distintos puedan compartir información entre sí, logrando así una fácil integración. Diferentes procesos pueden llevarse a cabo en una misma computadora o en varias que se encuentran distribuidas geográficamente a lo largo de una misma red.
- Exigen una verificación tanto en el cliente como en el servidor para evitar riesgos en la seguridad del esquema cliente - servidor.

III. DESARROLLO DE LA PLATAFORMA WEB PARA LABORATORIOS VIRTUALES

A. Convertir código a Java con Matlab Builder JA

Matlab Builder JA es una herramienta que permite crear clases Java a partir del código escrito en Matlab [19]. Estas clases de Java son integradas en las aplicaciones de Java a través del compilador Matlab Compiler Runtime (MCR), desplegándose gratuitamente en las computadoras de escritorio o servidores Web que no necesariamente tienen Matlab instalado. Es decir se crean componentes desplegables que hacen cálculos basados en Matlab, visualizaciones e interfaces, accesibles a los usuarios finales de los programas Java. Cuando el programa Java se implementa en la red, varios usuarios pueden acceder a él a través de un navegador Web. Las clases Java creadas con Matlab Builder JA son portables a un servidor de aplicaciones Web como Jboss (ver figura 2).

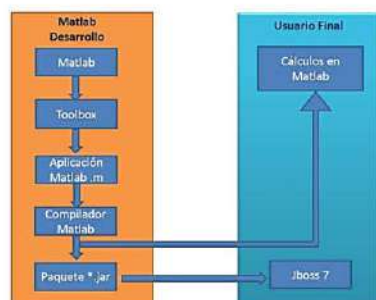


Fig. 2. Esquema de funcionamiento del toolbox Matlab Builder JA.
Fuente: Elaboración propia

B. Aplicaciones Web con el framework Spring de Java

Spring es un marco de trabajo (framework) [20] open source (código abierto) basado en el patrón MVC (Modelo Vista Controlador) [21] (ver figura 3).

Fue creado por Rod Johnson con la finalidad de facilitar la construcción de aplicaciones empresariales utilizando sencillos Javabeans [22], basado en la arquitectura cliente – servidor.

Las características más resaltantes de Spring son:

- Ligero: Spring, es un marco de trabajo ligero en términos de tamaño y memoria. La distribución de Spring puede darse en un simple archivo “jar” de tamaño aproximado de 2.5 mb.
- No intrusivo: Las aplicaciones desarrolladas en Spring no contienen dependencias hacia clases del marco de trabajo, lo que permite mantener la independencia de las mismas.
- Inyección de dependencias: Spring promueve el desacoplamiento entre clases conocido como inyección de dependencias.
- Contenedor: Spring es un contenedor dado que se encarga del ciclo de vida y configuración de los objetos utilizados por las aplicaciones.



Fig. 3. Framework Spring – Modelo Vista Controlador
Fuente: Elaboración propia

Spring también proporciona soporte para el desarrollo de aplicaciones Web en Java, el programador se centra sólo en la gestión y desarrollo de la aplicación, Spring ayuda a unir todos los componentes de la aplicación, gestionar su ciclo de vida y la interacción entre ellos.

Bajo este framework de programación la plataforma Web desarrollada aumentó el rendimiento, la estabilidad, y redujo también el coste computacional del sistema.

C. Implementación de simulaciones en Matlab y creación del paquete Java

Los códigos para implementar los procesos para simulación se desarrollaron en código .m de Matlab y fueron guardados en archivos con extensión .mat, para crear el archivo independiente de java (archivo con extensión .jar) con el toolbox Java Builder JA se ejecutan los siguientes pasos:

Escribir el comando “deploytool” en la ventana de comandos de matlab para abrir el toolbox Matlab Builder JA.

- Se abre la ventana Deployment Tool.
- Se procede a crear un nuevo proyecto: File - New Deployment Project.
- Se elige la opción Matlab Builder JA y se le asigna un nombre al proyecto.
- El siguiente paso es asignar los archivos .m al proyecto con la opción Add File.
- Una vez asignados los archivos .m se compila el proyecto con la opción Build the Project.
- Al finalizar se obtiene el código .m de matlab encapsulado en un paquete .jar para que se pueda usar en la programación Web con Spring.

D. Materiales y métodos para la construcción de la plataforma Web

La plataforma Web está desarrollada bajo el compilador de Sun Microsystems (JDK1.7.0) para servidores de 64 bits instalado en el sistema operativo Centos 6.5 para 64 bit. Esto no quiere decir que no se pueda implementar en servidores que trabajen bajo otros sistemas operativos, de hecho la característica multiplataforma de Java es una de sus mayores ventajas.

Para la implementación del entorno de desarrollo se utilizó el IDE (entorno de desarrollo integrado) NetBeans 7.3, el cual es un entorno para programadores que facilita la escritura, compilación y ejecución de programas en Java. El IDE NetBeans está escrito en Java y es un producto libre sin restricciones de uso.

Como servidor de Aplicaciones Web se usó el Jboss 7, es un servidor gratuito basado en patrones de J2EE (Java Enterprise Edition) y muy potente, además de ser multiplataforma.

Para la gestión de base de datos se usó MySQL [23] que es un gestor de bases de datos importante, muy conocido en entornos Linux tanto como Windows. Sus principales ventajas es que es multihilo y multiusuario permitiendo varias conexiones a la base de datos al mismo tiempo. También es uno de los motores de base de datos más usados en Internet.

E. Diseño Entidad Relación de la base de datos para la plataforma Web

La base de datos está diseñada para guardar el registro de los usuarios y las evaluaciones que se les hace a los usuarios antes de realizar el laboratorio virtual; está conformada por las siguientes tablas relacionadas (ver figura 4):

- Tabla Usuario: Guarda todos los usuarios del sistema ya sean alumnos o administradores o profesores.
- Tabla Usuario - Rol: Enlaza un usuario con un determinado rol.
- Tabla Ciclo: Guarda el ciclo donde se realiza el laboratorio.
- Tabla curso: Guarda el curso donde se realiza el laboratorio.
- Tabla Ciclo - Curso: Enlaza un determinado curso con un ciclo.
- Tabla Tipo - Evaluación: Guarda el tipo de evaluación por ejemplo si es práctica, laboratorio o examen.
- Tabla Evaluación: Guarda la descripción de la evaluación por ejemplo: nombre, día, hora.
- Tabla Tipo - Pregunta: Guarda los tipos de pregunta que pueden hacerse en una evaluación, por ejemplo: teóricas, prácticas, con alternativas.
- Tabla Pregunta: Guarda las preguntas de una determinada evaluación.
- Tabla Banco - Preguntas: Guarda un histórico de las preguntas registradas en el sistema.
- Tabla Alternativas: Guarda las alternativas de una determinada pregunta si esta es con alternativas.
- Tabla Solución: Guarda la alternativa correcta de una pregunta con alternativas.
- Tabla Alumno - Pregunta: Guarda la asignación de preguntas para cada alumno.

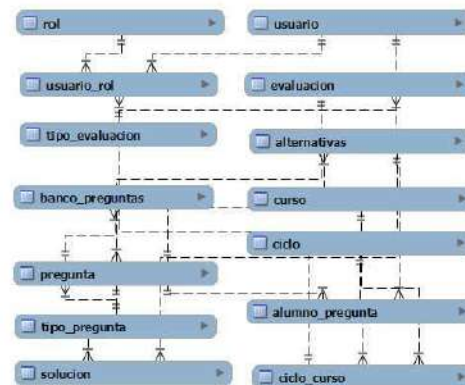


Fig. 4. Diagrama Entidad Relación de la plataforma web para laboratorios virtuales

Fuente: Elaboración propia

F. Funcionamiento de la plataforma Web

El funcionamiento de esta plataforma se probó con los alumnos de los cursos de Sistemas Automáticos de Control y Control Industrial de la Facultad de Ingeniería de la Universidad de Piura, Perú.

Los pasos para el uso de la plataforma virtual son los siguientes:

- Iniciar Sesión: La plataforma virtual tiene niveles de seguridad, cada alumno tiene un usuario y clave asignado (ver figura 5).
- Test de conocimientos: Para desarrollar el laboratorio virtual el alumno debe tener un mínimo de conocimientos sobre el tema del laboratorio a realizar. Es por eso que después de iniciar la sesión se les pide solucionar un test (ver figura 6) o cuestionario para validar si están capacitados para desarrollar el laboratorio virtual.



Fig. 5. Pantalla de inicio de sesión

Fuente: Elaboración propia

- Si el alumno responde de forma correcta más del 80% del cuestionario, puede realizar el laboratorio. De lo

contrario se le brindará una segunda oportunidad con un test diferente; si no logra resolver el segundo test, no podrá realizar el laboratorio virtual.

- Los alumnos que aprobaron el test cargarán la página del laboratorio con las respectivas cajas de texto para ingresar la información del proceso y empezar la simulación.



Fig. 6. Interfaz del test de conocimientos
Fuente: Elaboración propia

Como ejemplo de los laboratorios virtuales implementados podemos citar el de control en variables de estados del sistema péndulo invertido. Inicialmente el usuario debe linealizar el modelo matemático del péndulo invertido a un modelo en espacio de estados para obtener las matrices de estado. El usuario debe ingresar a la plataforma Web las matrices para graficar los estados y la salida del sistema a lazo abierto, calculando el determinante de las matrices de observabilidad y controlabilidad. De esta manera el usuario puede evaluar si el sistema es controlable y observable parcial o total. Así mismo, para la sintonización del controlador proporcional en espacio de estados, el usuario debe ingresar los polos del controlador y los polos del observador. Tomando en cuenta los polos, la plataforma Web grafica los estados y las salidas a lazo cerrado (ver figuras 7, 8, 9, 10).

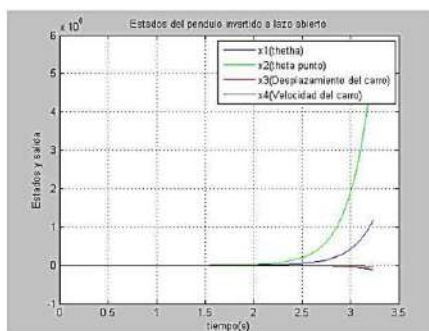


Fig. 7. Estados del péndulo invertido a lazo abierto
Fuente: Elaboración propia

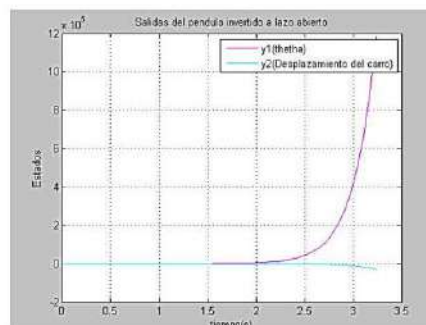


Fig. 8. Salida del péndulo invertido a lazo abierto
Fuente: Elaboración propia

Las ventajas que ofrece esta plataforma Web para el análisis de control en espacio de estados son:

- 1) Permite de manera inmediata analizar la controlabilidad y observabilidad del sistema.
- 2) Calcula de manera precisa la ganancia proporcional para el control en espacio de estados.
- 3) Permite sacar conclusiones gráficas con respecto a la estabilidad y asignación de polos.

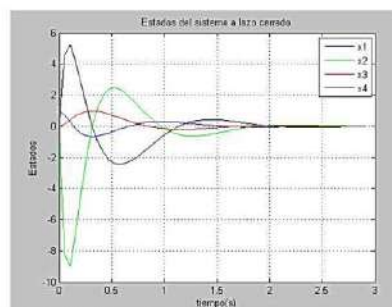


Fig. 9. Estados del sistema a lazo abierto
Fuente: Elaboración propia

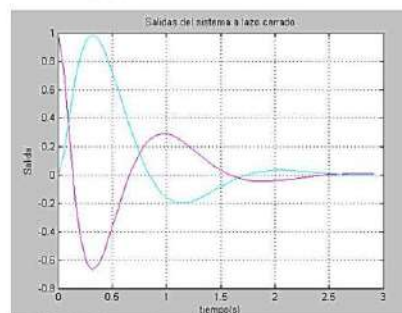


Fig. 10. Salida del sistema a lazo cerrado
Fuente: Elaboración propia

CONGRESO SALESIANO DE CIENCIA, TECNOLOGÍA E INNOVACIÓN PARA LA SOCIEDAD, CITIS
Guayaquil, 2 y 3 de Diciembre del 2015

IV. EVALUACIÓN PEDAGÓGICA

Los laboratorios virtuales y remotos ofrecen flexibilidad de horarios y comodidad a los alumnos, sin embargo no sería una buena alternativa sino se logran los objetivos educativos planteados. Por esta razón, los alumnos respondieron una encuesta online, basada en el trabajo de [9], acerca de su apreciación sobre los laboratorios virtuales realizados durante el dictado de los cursos de Sistemas Automáticos de Control y Control Industrial de la Universidad de Piura, Perú.

Dos temas principales agruparon a las preguntas del cuestionario:

- Visualización e interactividad: Incluye todo lo relacionado a la plataforma e interfaz creadas para los laboratorios virtuales.
- Perspectiva pedagógica: Permite saber la percepción del estudiante frente a esta nueva herramienta y cómo influye en su aprendizaje.

Cuadro 1: Cuestionario de la Encuesta online (19 personas)

Visualización e interactividad
-La interfaz de la plataforma Web para laboratorios virtuales es adecuada.
-Es cómodo usar la plataforma virtual Web.
-Disfrute utilizando la interfaz Web durante la práctica experimental.
-La interfaz de la plataforma Web virtual es fácilmente accesible y manejable.
-No se requiere mucho esfuerzo para llevar a cabo el laboratorio virtual a través de la interfaz Web.
Perspectiva pedagógica
-Las instrucciones indicadas en la guía de trabajo de laboratorio son suficientes.
-Los laboratorios virtuales me permiten entender mejor el curso.
-Los laboratorios virtuales motivan a aprender más sobre los temas en control automático.

El cuadro 1 muestra las preguntas de la encuesta presentada a 19 alumnos, los cuales fueron escogidos aleatoriamente.

El cuadro 2 muestra los resultados en porcentajes de los temas, las respuestas a cada pregunta fueron promediadas para cada tema evaluado y clasificado como muy de acuerdo, de acuerdo, neutro, en desacuerdo y muy en desacuerdo.

Cuadro 2: Resultados expresados en porcentajes de la encuesta online

Tema	Visualización e interactividad	Perspectiva pedagógica
Muy de acuerdo	6.3%	24.6%
De acuerdo	56.8%	29.8%
Neutro	27.4%	22.8%
En desacuerdo	6.3%	12.3%
Muy en desacuerdo	3.2%	10.5%

Los resultados indican que alrededor del 65% de los alumnos piensan que la plataforma e interfaz son las adecuadas para un mejor desarrollo de los laboratorios virtuales (visualización e interactividad) mientras que cerca del 10% opinan lo contrario. Con respecto a la perspectiva pedagógica, cerca del 55% encuentran que con la implementación de los laboratorios virtuales, los conocimientos impartidos en clase se aprovechan mucho más que con los laboratorios tradicionales, no obstante el 23% está en desacuerdo o muy en desacuerdo.

V. CONCLUSIONES

En este trabajo se presentó el desarrollo de una plataforma computacional que permite la realización de diferentes laboratorios virtuales a través de comunicación web; se han utilizado nuevas herramientas como el toolbox Matlab Builder JA y el framework Spring de Java para implementar esta solución.

La plataforma fue probada por los alumnos del curso de Sistemas de Control Automático y Control Industrial de la Facultad de Ingeniería de la Universidad de Piura, Perú, ofreciéndoles a los estudiantes una alta flexibilidad de horarios para hacer sus laboratorios virtuales. Esta solución permite a los usuarios practicar desde casa o donde quiera que se encuentren.

El desarrollo de laboratorios virtuales y remotos utilizando herramientas software y hardware de última generación, ofrece muchas ventajas en dos ámbitos: en el sector educativo, porque permite generar una nueva tecnología para el aprendizaje, y, en el campo industrial, porque usando esta arquitectura se mejoran las condiciones de supervisión, operación y control de los procesos industriales en las plantas, un campo muy importante de la ingeniería de hoy.

AGRADECIMIENTOS

Este trabajo se realizó con el apoyo de la Maestría en Ingeniería Mecánico-Eléctrica con mención en Automática y Optimización financiada por el Consejo Nacional de Ciencia y Tecnología de Perú (CONCYTEC) y con la colaboración del grupo de investigación del Laboratorio de Sistemas Automáticos de Control de la Universidad de Piura, Perú.

CONGRESO SALESIANO DE CIENCIA, TECNOLOGÍA E INNOVACIÓN PARA LA SOCIEDAD, CITIS
Guayaquil, 2 y 3 de Diciembre del 2015

REFERENCES

- [1] J.L. Guzmán, M. Domínguez, M. Berenguel, J.J. Fuertes, F. Rodríguez, P. Reguera, Entornos de experimentación para la Enseñanza de Conceptos Básicos de Modelado y Control, *Revista Iberoamericana de Automática e Informática Industrial RIAI*, Volume 7, Issue 1, January 2010, Pages 10-22, ISSN 1697-7912.
- [2] José Miguel Molina Jordá, Virtual Tools: Virtual Laboratories for Experimental science – An Experience with VCL Tool, *Procedia - Social and Behavioral Sciences*, Volume 106, 10 December 2013, Pages 3355-3365, ISSN 1877-0428.
- [3] Salman Nazir, Kjell Ivar Øvergård, Zaili Yang, Towards Effective Training for Process and Maritime Industries, *Procedia Manufacturing*, Volume 3, 2015, Pages 1519-1526, ISSN 2351-9789.
- [4] Fariás, G., Esquembre, F., Sanchez, J., Dormido, S., Vargas, H., Dormido-Canto, S., Dormido, R., and N. Duro, M.C. Desarrollo de laboratorios virtuales, interactivos y remotos utilizando casy java simulations y modelos simulink. *Proceedings XII Latin-American Congress on Automatic Control*, 2006.
- [5] Casini, M., Prattichizzo, D., y Vicino, A. Operating remote laboratories through a bootable device. *International Journal of Online Engineering*, Vol. 54, 2006, 3134-3140.
- [6] Corrado, M., Vito, L.D., Ramos, H., and Saliga, J. Hardware and software platform for adawan remote laboratory. *Measurement*, 45, 2012, 795 - 807.
- [7] Barrios, A., Panché, S., Duque, M., Grisales, V.H., Prieto, F., Villa, J.L., Chevrel, P., y Canu, M. A multi-user remote academic laboratory system. *Original Research Article Computers and Education*, Vol. 62, 2013, 111-122.
- [8] Orduña, P., Rodríguez-Gil, L., Angulo, I., Dziabenko, O., López-de Ipina, D., and García-Zubia, J. Exploring students collaboration in remote laboratory infrastructures. In *Remote Engineering and Virtual Instrumentation (REV)*, 2012, 9th International Conference on, 1-5.
- [9] Fábregas, E., Fariás, G., Dormido-Canto, S., Dormido, S., and Esquembre, F. Developing a remote laboratory for engineering education. *Computer and Education*, 57, 2011, 1686 - 1697.
- [10] Francisco Esquembre, Easy Java Simulations: a software tool to create scientific simulations in Java, *Computer Physics Communications*, Volume 156, Issue 2, 1 January 2004, Pages 199-204, ISSN 0010-4655.
- [11] Calvo, I., Zulueta, E., Oterino, F., y López, J.M. A remote laboratory for a basic course on control engineering. *International Journal of Online Engineering*, Vol. 5, 2009, 8-13.
- [12] Jin Hyun Son, Myoung Ho Kim, An analysis of the optimal number of servers in distributed client/server environments, *Decision Support Systems*, Volume 36, Issue 3, January 2004, Pages 297-312, ISSN 0167-9236.
- [13] D. Luna-Moreno, Y.M. Espinosa Sánchez, Y.R. Ponce de León, E. Noé Arias, G. Gamica Campos, Virtual instrumentation in LabVIEW for multiple optical characterizations on the same opto-mechanical system, *Optik - International Journal for Light and Electron Optics*, Volume 126, Issue 19, October 2015, Pages 1923-1929, ISSN 0030-4026.
- [14] Magdi S. Maimoud, Muhammad Sabih, Moustafa Elshafie, Using OPC technology to support the study of advanced process control, *ISA Transactions*, Volume 55, March 2015, Pages 155-167, ISSN 0019-0578.
- [15] Valentina Caputi, Antonio Garrido, Student-oriented planning of e-learning contents for Moodle, *Journal of Network and Computer Applications*, Volume 53, July 2015, Pages 115-127, ISSN 1084-8045.
- [16] Ahmed Maajoon Alnezi, Krishna Kirti Shahi, Interactive e-learning through Second Life with Blackboard Technology, *Procedia - Social and Behavioral Sciences*, Volume 176, 20 February 2015, Pages 891-897, ISSN 1877-0428.
- [17] Xiaocong Fan, Chapter 12 - Software Architectures for Real-Time Embedded Systems, In *Real-Time Embedded Systems*, edited by Xiaocong Fan, Newnes, Oxford, 2015, Pages 303-338, ISBN 9780128015070.
- [18] Gardel, A., Bravo, I., Galilea, J., and Del Toro, P. Remote automation laboratory using a cluster of virtual machines. *Industrial Electronics, IEEE Transactions on*, 57, 2010, 3276-3283.
- [19] Penitka Karel, Zsolt Tomas, Multimedia Teaching Aid for Students of Basics of Control Theory in MATLAB and SIMULINK, *Procedia Engineering*, Volume 100, 2015, Pages 150-158, ISSN 1877-7058.
- [20] Steffen Zschaler, Birgit Demuth, Lothar Schmitz, Salespoint: A Java framework for teaching object-oriented software development, *Science of Computer Programming*, Volume 79, 1 January 2014, Pages 189-203, ISSN 0167-6423.
- [21] Dragos-Paul Pop, Adam Altar, Designing an MVC Model for Rapid Web Application Development, *Procedia Engineering*, Volume 69, 2014, Pages 1172-1179, ISSN 1877-7058.
- [22] Herbert Prachhofer, Johannes Sametinger, Alois Stritzinger, Concepts and architecture of a simulation framework based on the JavaBeans component model, *Future Generation Computer Systems*, Volume 17, Issue 5, March 2001, Pages 539-559, ISSN 0167-739X.
- [23] Chapter 7 - MySQL on Linux, In *DBAs Guide to Databases Under Linux*, edited by David EganPaul Zikopoulos and Chris Rogers, Syngress, Burlington, 2000, Pages 283-309, ISBN 9781928994046.