APLICACIONES INFORMÁTICAS PARA APOYAR EL APRENDIZAJE POR PROYECTOS DE SISTEMAS DE CONTROL

Ing. Mauricio Ramírez Ramírez

Trabajo de Investigación presentado como requisito para optar al título de:
Magister en Ingeniería - Énfasis en Automática

Director: José Miguel Ramírez Scarpetta, Ph.D.

Universidad del Valle
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica y Electrónica
Santiago de Cali
Agosto de 2015

A Emilio, motor de mi motivación

A Lina, por su amor, paciencia y apoyo

A mis padres, por contribuir a alcanzar mis metas

AGRADECIMIENTOS

Al profesor José Miguel Ramírez, por su paciencia, apoyo, dedicación y compromiso para guiarme en cada una de las etapas de este proyecto.

A los profesores e integrantes del Grupo de Investigación en Control Industrial GICI por sus aportes.

A mi compañero de estudio Daniel Ruiz por su colaboración, aportes y sugerencias.

A Colciencias y a la Universidad del Valle, por el apoyo en financiamiento a través de las modalidades como asistente de investigación y asistente de docencia.

RESUMEN

La educación en sistemas de control, tiene altas exigencias en habilidades para la definición y solución de problemas, modelado y análisis matemático, uso de conceptos abstractos, diseño por análisis y síntesis y para la operación del sistema, siendo un área de alta dificultad en su aprendizaje, presentando repitencia, desmotivación y deserción en los estudiantes. En la metodología de Aprendizaje Basado en Proyectos ABP, a partir de un problema, los estudiantes autónomamente desarrollan estas habilidades en el marco de un proyecto de control. Para implementar esta metodología, se requiere software para el soporte matemático, utilizar tecnologías de control y comunicación y la operación del sistema de control; sin embargo, el software comercial y los laboratorios académicos de experimentación orientados a estos propósitos, son muy costosos.

En este trabajo de investigación se presenta el desarrollo de tres aplicaciones informáticas: Lúdicas Animadas, Herramienta de Análisis del Controlador Proporcional Integral Derivativo PID HAPID y Herramienta de Análisis de Sistemas de Control HASC, para apoyar la educación de pregrado en sistemas de control, con el enfoque ABP. Estas aplicaciones se desarrollaron con las herramientas libres *Easy Java Simulations* para el diseño de las interfaces gráficas de usuario y el *Scilab* como motor de cálculo matemático.

En las Lúdicas Animadas se describe de manera general el sistema, se dan unas actividades a realizar y se plantean unas preguntas de discusión, de esta forma los estudiantes entienden los conceptos sin una descripción matemática explícita y los motiva a aprender de una manera divertida mientras juegan con las animaciones. La aplicación HAPID entrena a los estudiantes para la fase Operativa y de Mantenimiento de un proyecto de control resuelto con un controlador PID. HAPID simula un ambiente de supervisión y control industrial para un intercambiador de calor, con funcionalidades de control de acceso, gestión de permisos, alarmas, eventos y reportes, ajuste y configuración del PID, válvula y sensor, y generación de diez fallas típicas en el lazo de control, para que los estudiantes las identifiquen y corrijan. La aplicación HASC facilita el procedimiento de análisis al obtener de una manera fácil y rápida el comportamiento temporal, frecuencial o del lugar de las raíces de un sistema continuo o discreto representado por su función de transferencia FDT o por su modelo en el espacio de estados EE y realiza transformaciones entre los dominios continuo y discreto y entre las representaciones (de FDT a EE y viceversa).

Las aplicaciones informáticas se evaluaron mediante encuestas, para conocer la percepción de los usuarios en el funcionamiento de la herramienta, contribución al aprendizaje y desarrollo de habilidades transversales. Los resultados muestran alta aceptación de las Lúdicas Animadas por facilitar el aprendizaje de conceptos, pertinencia de HAPID para el aprendizaje de la operación y el diagnóstico de fallas en sistemas de control y de HASC para el análisis de los sistemas de control. Las aplicaciones informáticas ofrecen un entorno de aprendizaje, donde los estudiantes realizan diferentes actividades para comprender conceptos abstractos, analizan comportamientos y diseñan sistemas de control. Presentan interfaces animadas con capacidades de interactividad, fáciles de manejar y entender; pueden utilizarse en cualquier momento y lugar, ya que son de uso libre, de fácil acceso y ejecutables en los sistemas operativos Windows y Linux. Su uso apoya el aprendizaje por proyectos, estimula el aprendizaje activo y promueve el desarrollo de habilidades transversales como el auto-aprendizaje, la toma de decisiones y la resolución de problemas.

TABLA DE CONTENIDO

RESUMEN ..... **IV**
1 INTRODUCCIÓN **1**
1.1 Dificultades de Aprendizaje **1**
1.2 Importancia y Pertinencia **2**
1.3 Objetivos **3**
1.3.1 Objetivo general 3
1.3.2 Objetivos específicos 3
1.3.3 Objetivos estratégicos 3
1.4 Estructura del Documento **4**
2 ANTECEDENTES Y MARCO TEÓRICO **5**
2.1 Sistemas Lineales de Control **5**
2.2 Metodología de Aprendizaje **6**
2.2.1 Aprendizaje basado en proyectos 6
2.3 Educación en Sistemas de Control en la Universidad del Valle **7**
2.3.1 Principales conceptos 7
2.3.2 Aspectos generales de los cursos 7
2.4 Aplicaciones de Apoyo en la Educación en Control **10**
2.4.1 El juego como estrategia didáctica para el aprendizaje 13
2.5 Herramientas de Desarrollo y Arquitectura Software **14**
2.5.1 Easy Java Simulations 14
2.5.2 Scilab 17
2.5.3 Arquitectura Cliente – Servidor 18
2.6 Resumen **19**
3 DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE HERRAMIENTAS ANIMADAS **20**
3.1 Sitio Web de las Aplicaciones Informáticas **20**
3.1.1 Arquitectura software de la plataforma PERI 20
3.1.2 Requerimientos 21
3.1.3 Diseño y desarrollo 22
3.2 Lúdicas Animadas **23**
3.2.1 Requerimientos 24
3.2.2 Diseño y desarrollo 26
3.2.3 Uso y pruebas de funcionamiento 61
3.3 Herramienta de Análisis del controlador PID **64**
3.3.1 Requerimientos 64

3.3.2 Diseño y desarrollo 66
3.3.3 Pruebas de funcionamiento 87
3.4 Resumen 91
4 DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE LA HERRAMIENTA DE ANÁLISIS MATEMÁTICO 92
4.1 Arquitectura Software de HASC 93
4.1.1 Interfaz JAVASCI 93
4.2 Requerimientos 94
4.2.1 Aplicación HASC-Cliente 94
4.2.2 Aplicación HASC-Servidor 96
4.3 Diseño y Desarrollo 97
4.3.1 Aplicación HASC-Cliente 98
4.3.2 Aplicación HASC-Servidor 106
4.4 Pruebas de Funcionamiento 109
4.4.1 Uso y Prueba de HASC-Servidor 109
4.4.2 Uso y Prueba de HASC-Cliente 110
4.5 Resumen 111
5 EVALUACIÓN 112
5.1 Instrumento de Evaluación 112
5.2 Resultados de la Evaluación 113
5.2.1 Lúdicas Animadas 113
5.2.2 Herramienta de Análisis del Controlador PID 115
5.2.3 Herramienta de Análisis de Sistemas de Control 116
5.3 Resumen 118
6 CONCLUSIONES Y TRABAJOS FUTUROS 119
6.1 Conclusiones 119
6.2 Trabajos Futuros 121
6.3 Publicaciones 122
ANEXOS 123
Anexo A: Guía para el profesor sobre las Lúdicas Animadas 123
Anexo B: Guía para la reparación de fallas en el lazo de control en HAPID 123
Anexo C: Guía de usuario de la aplicación HAPID 123
Anexo D: Guía de usuario de la aplicación HASC 123
Anexo E: Guía para establecer conexión entre EJS y Scilab 123
Anexo F: Guía para la instalación y ejecución local de HASC 123
Anexo G: Formatos de Evaluación 124
REFERENCIAS 127

LISTA DE FIGURAS

Figura 2-1 Elementos y señales de un Sistema de Control 5
Figura 2-2 Interfaz de desarrollo de *Easy Java Simulations* 15
Figura 2-3 Relación entre los componentes del patrón MVC 16
Figura 2-4 Modelo Cliente - Servidor 18
Figura 3-1 Arquitectura Hardware – Software de la plataforma PERI 21
Figura 3-2 Página principal del sitio web que contiene las Aplicaciones Informáticas 22
Figura 3-3 Esquema de navegación para acceder a las aplicaciones informáticas 23
Figura 3-4 Capturas del sitio web de las aplicaciones informáticas 23
Figura 3-5 Organización y presentación general de la página web de cada Lúdica Animada 25
Figura 3-6. Esquema de la Interfaz gráfica de usuario de cada animación 26
Figura 3-7 Página web de la Lúdica Animada Plano y Rueda 27
Figura 3-8 Estrategia didáctica para el aprendizaje de conceptos en las Lúdicas Animadas 28
Figura 3-9 Secuencia de implementación de la lúdica 1 en EJS 29
Figura 3-10 Estructura en forma de árbol para implementar la Vista de la lúdica 1 en EJS 29
Figura 3-11 Algoritmo de simulación de las Lúdicas Animadas 30
Figura 3-12 Diagrama de fuerzas de la rueda en el sistema plano y pivote 31
Figura 3-13 Vista de la lúdica 1: Plano y Rueda 32
Figura 3-14 Sistema del nivel en un tanque 33
Figura 3-15 Vista de la lúdica 2: Válvula y Tanque 34
Figura 3-16 Diagrama de fuerzas de la rueda en el sistema plano inclinado 36
Figura 3-17 Esquema del control de posición para el sistema del plano inclinado 36
Figura 3-18 Vista de la lúdica 3: Plano Inclinado Manual 37
Figura 3-19 Oscilación senoidal y circunferencia trigonométrica 38
Figura 3-20 Vista de la lúdica 4: Cuerda y Masa 39
Figura 3-21 Vista de la lúdica 5: Plano, Rueda y Disturbio 40
Figura 3-22 Vista de la lúdica 6: Válvulas y Tanque 42
Figura 3-23 Diagrama de fuerzas del sistema Masa y resorte 43
Figura 3-24 Vista de la lúdica 7: Masa y Resorte 44
Figura 3-25 Vista de la Lúdica 8: Plano y Ruedas 45
Figura 3-26 Vista de la lúdica 9: Plano Inclinado Automático 47
Figura 3-27 Vista de la lúdica 10: Plano, Rueda y Pivote 49
Figura 3-28 Sistema masa y resorte, excitado con una señal sinusoide 50
Figura 3-29 Vista de la lúdica 11: Masa, Resorte y Circunferencia 50
Figura 3-30 Sistema Masa - circunferencia 52
Figura 3-31 Control de posición con cancelación de disturbio 52
Figura 3-32 Vista de la lúdica 12: Cuerdas, Poleas y Masa 52
Figura 3-33 Sistema del nivel en dos tanques en serie 54
Figura 3-34 Estructura de control en cascada para el sistema del nivel en dos tanques 54
Figura 3-35 Esquema de control PI con Bumpless y Anti-windup para el TK_1 55
Figura 3-36 Vista de la lúdica 13: Tanques en Serie 56
Figura 3-37 Sistema del nivel en dos tanques en paralelo 57
Figura 3-38 Vista de la lúdica 14: Tanques en Paralelo 1 58
Figura 3-39 Vista de la lúdica 15: Tanques en Paralelo 2 60
Figura 3-40 Lúdica animada Válvula y Tanque 62
Figura 3-41 Contenido de los aspectos de la lúdica animada Válvula y Tanque 62
Figura 3-42 Comportamientos obtenidos del nivel para diferentes escalones del flujo de entrada . 63
Figura 3-43 Comportamiento obtenido del nivel alrededor de un punto de operación 63
Figura 3-44 Esquema de las interfaces gráficas de usuario de la aplicación HAPID 66

Figura 3-45 Interfaz del panel Principal del Proceso de la aplicación HAPID 67
Figura 3-46 Esquema del lazo de control de temperatura que implementa HAPID 68
Figura 3-47 Diagrama de flujo del nivel de acceso a paneles y funcionalidades en HAPID 69
Figura 3-48 Tipos de reportes en HAPID 69
Figura 3-49 Formato del nombre asignado al reporte 70
Figura 3-50 Algoritmo para crear y guardar un reporte 70
Figura 3-51 Ventanas emergentes para guardar un reporte 70
Figura 3-52 Esquema del Intercambiador de Calor IC-01 71
Figura 3-53 Calentamiento de un cuerpo homogéneo 71
Figura 3-54 Interfaz del panel de Acceso y sub-panel de Configuración 72
Figura 3-55 Interfaz del panel de Control y sub-panel de Configuración 73
Figura 3-56 Estructura de control, con PID ISA, Bumpless y Anti-windup 74
Figura 3-57 Algoritmo de ejecución de la ley de control en HAPID 76
Figura 3-58 Interfaz del panel de Eventos y sub-panel de Configuración 77
Figura 3-59 Algoritmo para registrar un evento 78
Figura 3-60 Ventana emergente con mensaje sobre el registro de eventos 78
Figura 3-61 Interfaz de configuración de la Válvula de Control 78
Figura 3-62 Partes de una válvula neumática. Imagen tomada de [60] 79
Figura 3-63 Modelo de la válvula de control con alinealidades 80
Figura 3-64 Interfaz de configuración del Transmisor de Temperatura 81
Figura 3-65 Estructura del transmisor de temperatura 81
Figura 3-66 Interfaz de selección de las Fallas 84
Figura 3-67 Algoritmo que implementa HAPID, para el manejo de los sub-paneles ante una falla 86
Figura 3-68 Comportamiento del proceso en modo “Manual” 87
Figura 3-69 Parámetros de ajuste del controlador PID 88
Figura 3-70 Comportamiento del proceso en modo “Automático” 88
Figura 3-71 Efecto de la falla “Perdida de Medida” en el comportamiento del sistema 89
Figura 3-72 Campo para resolver la falla del panel Fallas 89
Figura 3-73 Parámetros de configuración del transmisor TIT-101 90
Figura 3-74 Comportamiento del sistema después de corregir la “Perdida de medida” 90
Figura 4-1 Arquitectura software de HASC 93
Figura 4-2 Esquema de la interfaz gráfica de usuario de la aplicación HASC-Cliente 96
Figura 4-3 Esquema de la interfaz gráfica de usuario de la aplicación HASC-Servidor 97
Figura 4-4 Trama de petición que envía HASC-Cliente 97
Figura 4-5 Trama de respuesta que envía HASC-Servidor 97
Figura 4-6 Interfaz de usuario de la aplicación HASC-Cliente 98
Figura 4-7 Algoritmo de ejecución de la aplicación HASC-Cliente 99
Figura 4-8 Procesamiento de la trama de respuesta en HASC-Cliente 100
Figura 4-9 Construcción de la trama de petición en HASC-Cliente 103
Figura 4-10. Condiciones para realizar un análisis y/o transformación en HASC-Cliente 105
Figura 4-11 Interfaz de usuario de la aplicación HASC-Servidor 106
Figura 4-12 Algoritmo de ejecución de la aplicación HASC-Servidor 106
Figura 4-13 Tratamiento de las tramas de petición y de respuesta en HASC-Servidor 108
Figura 4-14 Resultado de la prueba de HASC-Servidor 109
Figura 4-15 Resultado de la prueba de HASC-Cliente 110
Figura 5-1 Evaluación Lúdicas Animadas, comparación de resultados 114
Figura 5-2 Resultado de la evaluación de HAPID 115
Figura 5-3 Resultado de la evaluación de HASC 117

LISTA DE TABLAS

Tabla 2-1 Características de las aplicaciones informáticas consultadas 12
Tabla 3-1 Clasificación y número de Lúdicas Animadas a desarrollar 24
Tabla 3-2. Lúdica Animada vs. Concepto 27
Tabla 3-3 Unidades y valores de las variables en la Lúdica 1 33
Tabla 3-4 Unidades y valores de las variables en la Lúdica 2 35
Tabla 3-5 Unidades y valores de las variables en la Lúdica 3 37
Tabla 3-6 Unidades y valores de las variables en la Lúdica 4 39
Tabla 3-7 Unidades y valores de las variables en la Lúdica 6 42
Tabla 3-8 Unidades y valores de las variables en la Lúdica 7 44
Tabla 3-9 Unidades y valores de la señal de excitación externa en la Lúdica 11 51
Tabla 3-10 Unidades y valores de las variables en la Lúdica 12 53
Tabla 3-11 Unidades y valores de las variables en la Lúdica 13 56
Tabla 3-12 Unidades y valores de las variables en la Lúdica 14 58
Tabla 3-13 Unidades y valores de las variables del proceso 72
Tabla 3-14 Permisos por defecto de los usuarios en HAPID 73
Tabla 3-15 Rangos y valores por defecto de los parámetros del controlador 76
Tabla 3-16 Descripción del tipo de evento 77
Tabla 3-17 Rangos y valores por defecto de los parámetros de la válvula de control 81
Tabla 3-18 Especificaciones del termopar tipo J 82
Tabla 3-19 Voltaje generado en función de la temperatura para el termopar tipo J 82
Tabla 3-20 Valor porcentual en función del voltaje del sensor 84
Tabla 3-21 Rangos y valores por defecto de los parámetros del transmisor de temperatura 84
Tabla 4-1 Valores permitidos de los parámetros de entrada en HASC-Cliente 95
Tabla 4-2 Argumentos de entrada y salida de las funciones *rtatem(), lgdr(), bod() y nyq()* 104
Tabla 4-3 Descripción de las funciones que utiliza Scilab para un análisis y/o transformación 107
Tabla 5-1 Preguntas de aprendizaje para las Lúdicas Animadas evaluadas 113
Tabla 5-2 Preguntas para el análisis de las encuestas sobre Lúdicas Animadas 114
Tabla 5-3 Preguntas de la encuesta sobre HAPID 115
Tabla 5-4 Preguntas de la encuesta sobre HASC 116

1 INTRODUCCIÓN

Debido al acelerado desarrollo de las tecnologías de la información y comunicación, a la globalización y a la necesidad social de ampliar la cobertura, la educación en ingeniería ha experimentado múltiples cambios. Estos cambios han requerido la reorganización de los planes de estudio y la búsqueda de nuevos modelos de enseñanza para aplicaciones multidisciplinarias y orientadas a problemas.

La educación en sistemas de control, es un área de naturaleza multidisciplinar y experimental, que requiere una forma de aprendizaje que incentive al estudiante a adquirir competencias en búsqueda de información, recolección de datos, solución de problemas y trabajo en equipo; indispensables para su desarrollo profesional [1]. Una metodología que cubre estos aspectos es el Aprendizaje Basado en Proyectos ABP, cuya aplicación requiere el acceso a la información, análisis y herramientas de experimentación de una manera más flexible que otras metodologías de aprendizaje tradicionales, por tanto se requieren herramientas tecnológicas apropiadas para soportarla [2]. A continuación se presentan las dificultades más comunes para el aprendizaje de sistemas de control reportadas por varios autores, la importancia, pertinencia y objetivos del presente trabajo de investigación, y por último la manera como se encuentra estructurado el documento.

1.1 Dificultades de Aprendizaje

En particular, los cursos de sistemas de control han sido enseñados con métodos de enseñanza tradicional basados en clases magistrales complementadas con prácticas de laboratorio orientadas a un diseño limitado de los sistemas de control [3]. En este tipo de enseñanza tradicional, el estudiante se convierte en un receptor pasivo de información [4], ya que solo se limita a escuchar, tomar apuntes y muy ocasionalmente, participa en clase, no potenciando su capacidad de percepción, asimilación y análisis. Generando graves consecuencias sociales y económicas, como altas repitencias [5], sobrecostos para la universidad, para el estudiante, demoras en la graduación, pérdida de autoestima, desmotivación, deserción. A diferencia de la enseñanza tradicional, en un ambiente de aprendizaje activo, el estudiante desarrolla diversas actividades bajo la tutoría del docente, para la solución de un problema o el logro de un objetivo; esto le permite trabajar más uniformemente, con mayor interés y dedicación, y en general involucrarse más con su proceso de aprendizaje [1].

La educación en sistemas de control, tiene altas exigencias en definición y solución del problema, análisis matemático, uso de conceptos abstractos, diseño por análisis y síntesis, siendo un área de alta dificultad en su aprendizaje. Como lo muestra la investigación realizada en [6] para un curso de control clásico, donde se observaron las siguientes dificultades de aprendizaje:

- Los estudiantes a menudo no están familiarizados con, o no se sienten entusiasmados por el fuerte contenido matemático presente en los temas del curso.
- El curso no lo ven útil en sus estudios posteriores.
- Para los estudiantes es una odisea aprobar el curso.
- Los estudiantes no son honestos con su proceso de aprendizaje y se dan cuenta de sus limitaciones en las evaluaciones.

En [7] se presenta un estudio sobre la dificultad de los diferentes temas del curso de Análisis de Sistemas de Control, en el cual los estudiantes calificaron los siguientes temas en orden ascendente de dificultad: transformada de Laplace, gráficos de Bode, criterio de estabilidad de Routh-Hurwitz, análisis de estado estable, análisis de estado transitorio, diagramas de bloques, modelado y finalmente criterio de estabilidad de Nyquist. Además como se plantea en [8], donde el autor afirma que: "*hay evidencia longitudinal sustancial, tanto en términos de calificaciones y realimentación por parte de los estudiantes, que una gran mayoría encuentran estos temas difíciles de entender y por tanto muchos aprueban los exámenes de una manera u otra para no volver a estudiar los sistemas de control*".

La validación experimental representa un componente fundamental de la educación y la investigación en el campo de la ingeniería de control, ya que estimula la reflexión y por tanto una compresión más profunda. Sin embargo, sus implementaciones prácticas presentan numerosas restricciones debido al costo de los equipos, las limitaciones en las capacidades espaciales, el tiempo y el mantenimiento, permaneciendo el laboratorio cerrado e infrautilizado la mayor parte del tiempo [9]. Obviamente, estos factores constituyen por si mismos una barrera a un uso más eficiente de los recursos ya disponibles. Actualmente el uso de las tecnologías de la información y de la comunicación han permitido derribar la barrera existente o reducirla en la medida de lo posible, generando la aparición de un nuevo abanico de conceptos relacionados con la realización de las practicas: tele-laboratorios, laboratorios remotos, laboratorios virtuales interactivos en control, simulación basada en la web, redes de laboratorios, servidores de prácticas, etc, [10]. Sin embargo la mayoría de estas plataformas requieren el uso de software comercial, para el cual no siempre las instituciones disponen de las licencias actualizadas.

1.2 Importancia y Pertinencia

La educación en control es un área multidisciplinar por estar involucrada en la formación de varias disciplinas de la ingeniería, como eléctrica, electrónica, mecánica, mecatrónica, alimentos, química. Sin embargo, para su aprendizaje y aplicación se tienen altas exigencias en análisis matemático y resolución de problemas, que exigen el uso de conceptos abstractos, la simulación previa de las soluciones y la experimentación final.

En el campo profesional la práctica de la ingeniería se realiza mediante proyectos, siendo el ABP, una metodología pertinente para formar nuevos profesionales, ya que promueve el aprendizaje activo de los estudiantes, quienes autónomamente abordan los conceptos, el análisis y la tecnología en la medida que se lo exija un proyecto para la solución del problema. Con esta metodología se genera un mejor ambiente de aprendizaje y por tanto se disminuyen las repitencias en el área, motivando al estudiante para que autónomamente trabaje activamente en la construcción del conocimiento [11].

El uso de esta metodología exige tener herramientas de aprendizaje más flexibles que permitan a estudiantes que se encuentren en sitios aislados acceder a los recursos en cualquier momento, generando de esta manera capacidades de formación asíncrona y ubicua. Por tanto se requiere el uso de la tecnología y de las comunicaciones, siendo los recursos existentes de software comercial y los laboratorios académicos de experimentación muy costosos. Actualmente se están implementando desarrollos tecnológicos emergentes para la enseñanza, aprendizaje o investigación en la educación de la ciencia, tecnología, ingeniería y matemáticas (STEM), como: computación en la nube, aplicaciones móviles, entornos personales de aprendizaje, entre otros [12].

Por lo anterior, la presente investigación busca aportar a la comunidad académica aplicaciones informáticas de apoyo al aprendizaje, implementadas con software libre, que puedan ejecutarse en los sistemas operativos Windows y Linux, accesibles por Internet, con interfaces animadas y con capacidades de interactividad, que apoyen la asimilación de conceptos, el análisis y el diseño en la educación de pregrado en sistemas de control del área de automática de la Universidad del Valle, con el enfoque de aprendizaje por proyectos.

Se debe destacar que este trabajo de investigación se enmarca en el proyecto: "*Entorno de Aprendizaje Basado en Proyectos para Sistemas de Control*" [13], financiado por COLCIENCIAS (C.I. 2650); en el cual se desarrolla un Libro Electrónico concebido para que tenga un impacto a nivel nacional, con contenido en hipertexto, interactivo y principalmente orientado al auto-aprendizaje, que requiere por tanto del desarrollo de aplicaciones informáticas complementarias que apoyen la asimilación de conceptos, el análisis y el diseño de sistemas de control.

1.3 Objetivos

1.3.1 Objetivo general

Concebir y desarrollar aplicaciones informáticas, con software libre, ejecutables en los sistemas operativos Windows y Linux, y accesibles por Internet, para apoyar la educación de pregrado en sistemas de control, del área de automática de la Universidad del Valle, con el enfoque de aprendizaje por proyectos.

1.3.2 Objetivos específicos

- Analizar las principales limitaciones, procesos para el desarrollo del proyecto y conceptos en la educación de pregrado en sistemas de control, del área de automática de la Universidad del Valle.
- Definir los requerimientos y especificaciones técnicas de las aplicaciones informáticas de apoyo al aprendizaje.
- Concebir y desarrollar la arquitectura software de las aplicaciones informáticas de apoyo al aprendizaje.
- Diseñar e implementar las aplicaciones informáticas de apoyo al aprendizaje.
- Validar las aplicaciones informáticas de apoyo al aprendizaje desarrolladas.

1.3.3 Objetivos estratégicos

- Proporcionar aplicaciones informáticas de apoyo al aprendizaje a la comunidad académica, para la educación de sistemas de control.
- Apoyar la asimilación de conceptos, el análisis y el diseño en la educación de sistemas de control.
- Generar capacidades de formación asíncrona y ubicua en la educación en control.
- Posicionar a nivel internacional el grupo GICI en el área de la educación en control al proveer herramientas informáticas de apoyo al aprendizaje de conceptos de sistemas de control.

1.4 Estructura del Documento

Este documento se encuentra conformado por seis capítulos, anexos y referencias bibliográficas, organizados de acuerdo a la siguiente distribución:

En el capítulo 2 se presenta la fundamentación teórica y los antecedentes que soportan el presente trabajo de investigación. Se muestran los aspectos más relevantes y como se implementa la metodología de Aprendizaje Basado en Proyectos en los cursos de sistemas lineales de control en la Universidad del Valle; finalmente se presentan las principales características de los software libres *Easy Java Simulations EJS* y *Scilab* como herramientas de desarrollo de aplicaciones informáticas.

En el capítulo 3 se presenta el diseño y desarrollo e implementación de las herramientas animadas, las cuales consisten de 15 Lúdicas Animadas y de la Herramienta de Análisis del Controlador PID HAPID desarrolladas en EJS. Las Lúdicas Animadas están dirigidas a facilitar el aprendizaje de los conceptos más importantes de la teoría de los sistemas de control, y la aplicación HAPID, se utiliza como soporte para el análisis del controlador PID y aprendizaje de las diversas funciones de operación y de detección de fallas en lazos de control.

En el capítulo 4 se presenta el diseño y desarrollo e implementación de la herramienta de análisis matemático, la cual por el grado de complejidad en su concepción se documentó en este capítulo de manera independiente. Consiste de la Herramienta de Análisis de Sistemas de Control HASC, desarrollada en EJS y Scilab; con EJS se implementó la interfaz gráfica de usuario y se usó Scilab como motor de cálculo matemático.

En el capítulo 5 se presenta la evaluación de las aplicaciones informáticas desarrolladas por medio de encuestas con una escala de valoración de 1 a 5, 1 = no cumplimiento y 5 = excelente nivel de cumplimiento, y se muestra el análisis de los resultados basados en aspectos como: el funcionamiento de la aplicación, su impacto en el aprendizaje y el desarrollo de habilidades transversales.

En el capítulo 6 se presentan las conclusiones y se exponen los trabajos futuros.

2 ANTECEDENTES Y MARCO TEÓRICO

En este capítulo se plantea la fundamentación teórica y los antecedentes que soportan el presente trabajo de investigación. Inicialmente se define que es un sistema de control y se presentan los aspectos más relevantes de la metodología de Aprendizaje Basado en Proyectos ABP. Seguidamente se muestra el ABP como estrategia pedagógica para la enseñanza de sistemas lineales de control en la Universidad del Valle, además se presentan los conceptos y temáticas que conforman los planes de estudio de los cursos. Posteriormente se realiza una revisión sobre antecedentes de aplicaciones informáticas usadas como apoyo en la educación de sistemas de control, y finalmente se muestran las herramientas de software libre *Easy Java Simulations* y *Scilab* como alternativas interesantes para el desarrollo de aplicaciones informáticas.

2.1 Sistemas Lineales de Control

Un sistema de control es un conjunto de elementos dinámicos interconectados que realizan una acción con retroalimentación, para alcanzar un objetivo dado. Los estímulos y respuestas del sistema se denominan señales o variables. La señal de Referencia, es un estímulo aplicado al sistema de control para producir una respuesta deseada. La señal de Salida, es la respuesta obtenida, la cual puede ser diferente a la deseada. La señal de Perturbación es una entrada que afecta adversamente a la salida. Los elementos que conforman el sistema de control son: la planta, el actuador, el controlador y el bloque de medición. La planta es el objeto del control, el actuador amplifica la señal de salida del controlador (señal de control) a los niveles de la variable de entrada a la planta (variable manipulada), la cual es manipulada por el controlador con el fin de que la variable de salida se comporte de acuerdo a una referencia deseada y el bloque de medición permite hacer la medición de la salida de la planta (señal de realimentación) a través de sensores o transductores, como lo muestra la Figura 2-1.

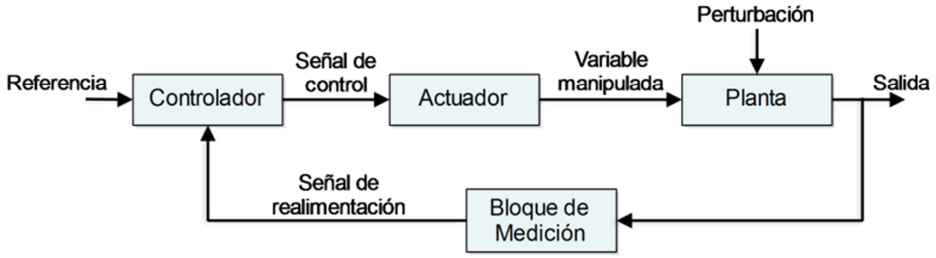


Figura 2-1 Elementos y señales de un Sistema de Control

Para el caso de la ingeniería, la teoría de control se aplica con el propósito de controlar variables, casi siempre de tipo industrial, con el fin de automatizar una tarea para aumentar la producción o mejorar la calidad de un servicio o producto. Para implementar un sistema de control es necesario obtener el modelo de la planta, analizar su dinámica y diseñar la estrategia de control.

El *modelado* es el proceso mediante el cual se obtiene una representación matemática del sistema por medio de leyes físicas y ecuaciones diferenciales; el *análisis* permite observar y evaluar el comportamiento dinámico del sistema de control; y el *diseño* de la estrategia de control, consiste en elegir el controlador adecuado que permita obtener el comportamiento deseado del sistema [11], [14].

2.2 Metodología de Aprendizaje

Los retos actuales de la ingeniería en control exigen una educación que permita a los ingenieros cruzar los límites de su campo profesional. Es decir, deben ser capaces no sólo de analizar los aspectos teóricos y técnicos de la solución de control, sino también de observar su relación con el contexto y otros campos profesionales. Los nuevos desafíos requieren, entre otras cosas, desarrollar habilidades como el aprendizaje permanente para mantener la competitividad en un campo profesional en constante cambio, comunicar ideas a los demás, tomar decisiones, aprender por sí mismos, proteger el medio ambiente, evaluar el impacto social de las soluciones de ingeniería y trabajar en equipo, [14].

2.2.1 Aprendizaje basado en proyectos

Un proyecto es definido como un esfuerzo complejo que requiere un análisis del problema, debe ser planificado, administrado y completado en un determinado tiempo [15]. Su desarrollo exige el aprendizaje de conceptos, procedimientos y actitudes mediante los cuales será posible conseguir los objetivos del mismo; es un ejercicio propio del desempeño profesional de los ingenieros: Concebir, Planear y Ejecutar [11].

En el ABP, el estudiante busca recursos y desarrolla actividades necesarias para resolver un problema a través de un proyecto. En esta metodología, los estudiantes abordan los conceptos cuando los requieren, contrario al enfoque tradicional donde se presentan los conceptos y procedimientos pero en la mayoría de los casos no se observa su aplicación de forma inmediata. Según el Instituto Buck para la Educación (BIE), algunos de los elementos esenciales del ABP son [16]:

- Contenido significativo: En su esencia, el proyecto se centra en enseñar a los estudiantes conocimientos y habilidades importantes, derivadas de las normas y conceptos clave de las materias académicas.
- Competencias del siglo 21: Los estudiantes construyen competencias valiosas para el mundo de hoy, tales como la resolución de problemas, pensamiento crítico, trabajo en equipo, comunicación, creatividad e innovación.
- Necesidad de saber: Los estudiantes ven la necesidad de adquirir conocimientos, comprender los conceptos y aplicar habilidades para resolver el problema.
- Revisión Crítica: El proyecto incluye procesos para que los estudiantes aporten y reciban realimentación sobre la calidad de su trabajo, realizando revisiones y más investigación.

Las principales características del ABP, que lo destacan como una de las mejores estrategias pedagógicas para educar por competencias a estudiantes de ingeniera, son [11]:

- Permite la adquisición de competencias de diversa índole.
- Introduce el contexto de aprendizaje en un entorno real.
- Exige al estudiante un papel más activo.
- Cambia el rol del profesor al de facilitador del proceso de aprendizaje y al de consultor de alta especialización y supervisión.
- Permite optimizar el grado de atención individualizada que se ofrece a los estudiantes.

2.3 Educación en Sistemas de Control en la Universidad del Valle

En la Universidad del Valle la metodología ABP se aplica en los dos cursos del área de sistemas lineales de control del programa de Ingeniería Electrónica. El primer curso corresponde a Fundamentos de Control de Sistemas Lineales FCSL y el segundo a Análisis y Compensación de Sistemas Lineales ACSL.

La implementación del enfoque ABP toma como base la investigación realizada en la tesis de doctorado "*Enfoque de Aprendizaje Basado en Proyectos para Sistemas Lineales de Control*" [5]. Este trabajo muestra una completa experiencia del ABP en la enseñanza de control teniendo en cuenta aspectos como: la definición del problema que utiliza problemas de contexto, el cronograma del proyecto que considera una rotación entre los equipos de estudiantes, la organización de actividades académicas tomando la ejecución del proyecto como carta de navegación, las clases magistrales que incluyen tutorías y actividades lúdicas manos a la obra (*Hands-On)*, recursos de apoyo que incluyen la simulación y la emulación, la evaluación de los estudiantes la cual considera la coevaluación y la autoevaluación, y evaluación del curso a través de observar el impacto del enfoque ABP. En este enfoque los estudiantes desarrollan un proyecto que busca resolver un problema de control definido en un contexto real; de esta manera, aprenden los temas mientras resuelven el problema. El enfoque se centra en promover el desarrollo de competencias transversales como: *trabajo en equipo, solución de problemas, aprendizaje autónomo y habilidades comunicativas*.

2.3.1 Principales conceptos

El trabajo presentado en [17] muestra la elaboración de un plan de estudio práctico el cual considera la educación en control como una disciplina. Según los autores la enseñanza de los sistemas de control debe enfocarse en el conocimiento de cuatro conceptos básicos que deben entenderse por los ingenieros y que deben ser parte fundamental de los planes de estudio de los cursos de sistemas de control:

1. Modelado
2. Estabilidad
3. Realimentación
4. Compensación

Teniendo en cuenta la experiencia presentada en [17], en la educación en sistemas de control en la Universidad del Valle se cubren estos conceptos al relacionarlos en temáticas, como: realimentación, modelado matemático, análisis temporal y frecuencial, estabilidad, compensación, reconstrucción de señales, estructuras de control, controlabilidad y observabilidad.

2.3.2 Aspectos generales de los cursos

2.3.2.1 Objetivos

Los objetivos propuestos para los cursos de control son [5]:

- Generales
  - Analizar sistemas lineales de control analógicos y discretos en el dominio del

tiempo y la frecuencia, con representación de entrada-salida o de estado.
  - Diseñar controladores PID y RST utilizando la teoría de control clásica en la que se incluyen la ubicación de polos, el análisis del lugar geométrico de las raíces y la respuesta en frecuencia.
  - Diseñar controladores y/o observadores de estado para los sistemas descritos por modelos en variables de estado.

- Estratégicos
  - Desarrollar los conocimientos, habilidades y valores del estudiante para afrontar los procesos de identificación y solución de problemas de sistemas de control.
  - Desarrollar capacidades para el trabajo en equipo y el autoaprendizaje.
  - Desarrollar habilidades comunicativas y para la gestión del tiempo, la información y los recursos físicos.

2.3.2.2 Metodología

Para alcanzar los objetivos planteados se utiliza el *aprendizaje activo*, esto significa que el estudiante es el gestor de su propio conocimiento y el docente tiene el rol de facilitador del aprendizaje y experto asesor en el tema.

Como estrategia pedagógica se considera el ABP cuya implementación parte del problema, el cual es definido teniendo en cuenta tres elementos: un contexto, una variable y un objetivo de control. Para resolver el problema se desarrolla un proyecto, el cual es ejecutado en etapas durante el semestre. Las temáticas a tratar en los cursos se basan en el desarrollo de las siguientes etapas del proyecto [5], [18]:

1. Definir el problema de control
2. Representar el sistema mediante modelos
3. Analizar el comportamiento del sistema
4. Diseñar, implementar y ajustar la estrategia de control

En el curso FCSL se trabajan las tres primeras etapas:

- Definición del Problema: Se introduce la importancia del área, la terminología a usar y se describen los principales elementos y señales de las buclas típicas de control análoga y digital.

- Modelado: Se conoce la representación y se simplifica un sistema de control por medio de un diagrama de bloques, se determina el modelo matemático en función de transferencia y variables de estado para sistemas de control análogos, discretos y digitales, se analizan las diferentes interrelaciones entre las representaciones entrada-salida y de estado; y por último se analiza el efecto de la realimentación en el funcionamiento de un sistema de control.

- Análisis: Se calcula la respuesta en el tiempo de sistemas análogos, discretos y digitales, se conocen y determina el efecto de las diferentes acciones de control (Proporcional, Integral y Derivativa PID y sus combinaciones) en el comportamiento de un sistema, y se implementa la acción de control PID de manera análoga y discreta.

En el curso ACSL se trabajan las dos últimas etapas:

- Análisis: Se analiza la estabilidad interna de los sistemas de control, se analizan sistemas de control continuos y discretos mediante las técnicas del lugar de las raíces y la respuesta en frecuencia, se analizan las propiedades de estabilidad, controlabilidad y observabilidad de un sistema.

- Diseño: Se definen los requerimientos para el diseño de un sistema de control, en términos de reglas, restricciones, arquitecturas posibles y las especificaciones deseadas de desempeño tanto en el tiempo como en la frecuencia, se determinan los parámetros de una ley de control utilizando los métodos del lugar geométrico de las raíces y de la respuesta en frecuencia para controladores PID y el diseño analítico para controladores PID y RST, se diseñan leyes de control por realimentación de estados y observadores de estados.

- Implementación y ajuste: Se implementan y evalúan diversas estrategias para el control de un proceso, se identifican y evalúan funciones complementarias, como inicialización, arranque/parada, parada de emergencia, automático/manual, operación en falla y recuperación, cambios sin saltos en la señal de control, conectividad, rebose del controlador, filtrado de antiplegamiento, antiembalamiento, soporte para ajuste en línea, supervisión y diagnóstico del desempeño del controlador.

2.3.2.3 Resultados de aprendizaje

Los resultados de aprendizaje de acuerdo al curso de control son [5]:

En el curso FCSL, los estudiantes describen los componentes del sistema tales como sensores, actuadores y la planta, y sus respectivas señales; representan el sistema usando normas ISA y analizan el contexto donde el sistema se encuentra. Después, los estudiantes obtienen el modelo del sistema, en este caso, ellos aplican sus conocimientos sobre ecuaciones diferenciales, ecuaciones de diferencia, algebra de bloques, función de transferencia y espacio de estados, también, utilizan métodos experimentales como la respuesta al escalón. El modelo del sistema es simulado en tiempo real y su respuesta en el tiempo es analizada. Finalmente, los estudiantes sintonizan controladores PID.

En el curso ACSL, los estudiantes analizan el sistema en el dominio de la frecuencia y a través del lugar geométrico de las raíces, posteriormente, ellos diseñan controladores analógicos y digitales por medio de la respuesta en frecuencia, el lugar de las raíces o la ubicación de polos; también, se estudian estructuras de control como *feedforward* y cascada. Finalmente, los estudiantes diseñan observadores y controladores a través del espacio de estados.

2.3.2.4 Desarrollo de competencias transversales

La metodología ABP considerada, se centra en la observación del desarrollo de las capacidades de trabajo en equipo, auto-aprendizaje, recursividad, creatividad, resolución de problemas y habilidades comunicativas. Adicionalmente, otras competencias transversales como la toma de decisiones, la gestión del tiempo y la información son promovidas desde el trabajo en el proyecto, en donde los estudiantes son libres para organizar el trabajo, definir sus roles, asignar tareas y supervisar el progreso de las tareas asignadas.

2.4 Aplicaciones de Apoyo en la Educación en Control

En la actualidad, una nueva generación de aplicaciones informáticas ha creado una alternativa interesante para el aprendizaje interactivo en el campo de la ingeniería.

Como lo justifica el reporte presentado en [12], donde se analiza el impacto de implementar desarrollos tecnológicos emergentes en la enseñanza, aprendizaje o investigación en la educación de ciencia, tecnología, ingeniería y matemáticas (STEM), en el cual también se identifica las tendencias y los desafíos de la educación para los próximos cinco años (2012 - 2017). El documento define 12 tecnologías emergentes claves: *Cloud computing, Collaborative environments*, *Mobile Apps*, *Social networking*, *Augmented reality*, *Learning analytics*, *Massively Open Online Courses (MOOCs)*, *Personal learning environments*, *Collective intelligence, The Internet of Things, Natural User Interfaces (NUIs)*, *Wearable technology;* de las cuales se destaca su relevancia para la enseñanza y aprendizaje en la educación STEM y se muestran varios ejemplos reales de su uso en la práctica educativa.

La revisión presentada en [19], es uno de los documentos más completos que existen sobre la caracterización de herramientas tecnológicas de apoyo a la educación en control. El autor sugiere que en el diseño de "*herramientas de educación en control*", se deben tener en cuenta aspectos como:

- Interacción hombre-máquina amigable
- Interfaces gráficas de usuario naturales e intuitivas
- Alto grado de interactividad
- Accesibles a través de internet

Además, en este documento se tratan temas como la interactividad en la educación en control, las implicaciones de utilizar la WWW (*World Wide Web*) en el desarrollo instruccional; los criterios para el diseño de simulaciones basadas en la web y sus áreas de aplicación y se define el concepto de tele-operación en la enseñanza, detallando sus ventajas, desventajas y los requisitos para la experimentación remota. Finalmente plantea una clasificación para los ambientes de aprendizaje de acuerdo a la forma que pueden ser accesados (locales o remotos) y a su naturaleza, (simulados o reales).

A continuación se presentan algunos trabajos sobre aplicaciones informáticas usadas como apoyo para el aprendizaje de sistemas de control:

Las aplicaciones "Ictools" y "CCSdemo" [20], [21], utilizan el concepto de *imágenes dinámicas*, donde por medio de un conjunto de ventanas, los estudiantes manipulan directamente las representaciones gráficas de los sistemas a través del mouse, obteniendo de manera simultánea información de este efecto y diferentes comportamientos del sistema. Las aplicaciones se desarrollaron con el software *Matlab* [22], y se utilizan como herramientas de análisis y diseño en los cursos de sistemas automáticos de control y sistemas de control por computador, en el Instituto Tecnológico Lund en Suecia. Para ilustrar cómo las herramientas interactivas de aprendizaje pueden mejorar la comprensión del concepto de modelado matemático de un proceso, los autores implementaron el proceso físico del nivel en un tanque como una animación, en la cual el nivel del líquido se manipula directamente arrastrándolo, reflejando inmediatamente los cambios en una gráfica y en las ecuaciones. El uso de las animaciones para recrear sistemas dinámicos, los autores la denominan Sistemas Interactivos Virtuales (VSI).

En [23] se desarrolló una aplicación de libre acceso basada en la Web para el análisis y diseño de sistemas de control, denominada "WCDAS" [24], que utiliza un intérprete C/C++ conocido como Ch y el *toolkit* de sistemas de control Ch (CCST) que contiene funciones para el análisis y diseño de sistemas lineales de control invariantes en el tiempo (LTI). WCDAS cubre muchas técnicas de control clásico y moderno.

En [25] se presentan dos laboratorios virtuales basados en *Matlab* para el diseño de sistemas de control. El primero se denomina "Web Sisotool", el cual es una aplicación basada en el *Matlab Web Server* (MWS) para el diseño controladores PID, usando el diagrama de Bode o el método del lugar de las raíces. El segundo se denomina "M-file application", que permite a los estudiantes crear y diseñar experimentos de sistemas de control, escribiendo el código *Matlab* en un archivo o script .m. Las aplicaciones dependen del protocolo de comunicación TCP/IP para la transmisión de datos entre el sistema cliente y Matlab.

En [26] se muestra el desarrollo de un laboratorio virtual, basado en una arquitectura "Cliente – Servidor" que utiliza el protocolo TCP/IP. Consiste de una animación de dos tanques en cascada, en el cual se controla el nivel del segundo tanque por medio de un controlador PID. En el Cliente se realiza la interfaz de usuario, por medio de una página web dinámica en *Javascript* que utiliza el formato SVG para la representación gráfica. En el Servidor se desarrolla el modelo matemático del proceso, el cual se implementó en el software *SciLab* [27] ya que se basa en una tecnología abierta y ofrece una funcionalidad similar a *Matlab*. Para el uso de *Scilab* en un entorno de internet se utiliza el *toolbox Socket* y scripts PHP.

En [28] se presenta el desarrollo de aplicaciones interactivas en control usando la herramienta de software *Easy Java Simulations (EJS),* [29]. *EJS* es una herramienta de código abierto desarrollada en *Java*, diseñada específicamente para ayudar a no-programadores crear simulaciones científicas con capacidades gráficas de alto nivel y con un mayor grado de interactividad. El documento describe en detalle el uso de *EJS* en conexión con *Matlab/Simulink* para desarrollar dos aplicaciones interactivas en el campo de la ingeniería de control; básicamente se utiliza *EJS* para implementar la interfaz gráfica de usuario (GUI), y *Simulink* para ejecutar el modelo matemático. La primera aplicación consiste de una simulación interactiva del sistema dinámico de un péndulo simple. La segunda aplicación consiste en una herramienta para el análisis de un sistema de control MIMO. Ambas aplicaciones le permiten al estudiante observar y analizar de manera inmediata, los efectos de modificar parámetros del modelo por medio de las GUI implementadas.

En [30] se describe un nuevo acercamiento para crear laboratorios remotos de control interactivos. El laboratorio creado es una aplicación *Java* diseñada para controlar un Servo-motor a través de internet, el cual se encuentra localizado en la Universidad de Ghent en Bélgica. El acercamiento considera principalmente el uso de las herramientas de software *Matlab* para controlar la planta real y *EJS* para implementar la GUI. El documento describe como realizar la conexión entre *Matlab* y *EJS* de manera local y de manera remota por medio de la herramienta *Jim Server*, que usa el protocolo TCP/IP; además se describe el hardware y software requerido como mínimo para implementar el laboratorio remoto: servidor Web, tarjetas de adquisición (DAQ) y cámaras IP. Una aplicación similar es presentada en [31], [32] donde se muestra un completo laboratorio virtual y remoto de un sistema MIMO no-lineal de un proceso de tres tanques en paralelo, la aplicación se puede acceder por medio de un explorador web ya que se encuentra

embebida a la página como un *Applet de Java*; la interfaz gráfica de usuario y el modelo matemático del proceso es desarrollado en *EJS*, la arquitectura del laboratorio remoto se basa una simple estructura Cliente/Servidor que utiliza como canal de comunicación una conexión TCP.

Otros aportes del uso de recursos tecnológicos para la educación en control es la aplicación presentada en [33] denominada "COCT", la cual fue desarrollada en *Excel* para el análisis de sistemas lineales y diseño de controladores. En [34] se presenta la herramienta de software libre conocida como "Rltool" implementada con *Scilab* para el análisis y diseño de sistemas de control. En [35] se describen tres módulos interactivos de aprendizaje diseñados a desarrollar la intuición y un conocimiento práctico del controlador PID, los tres módulos constituyen un paquete llamado módulo de aprendizaje interactivo para PID "ILM-PID", el cual se implementó en el software *Sysquake* [36] (en su versión libre), similar a *Matlab* y usado principalmente para el diseño y simulación de sistemas dinámicos. Finalmente en [37] el autor presenta su experiencia personal del uso de una nueva familia de herramientas interactivas para la educación en control, donde se destaca la herramienta para el Análisis y Diseño de sistemas Lineales "LSAD".

En la Universidad del Valle se destacan los siguientes trabajos: la Plataforma de Educación Remota en Ingeniería PERI [38] desarrollada por el grupo de investigación en control industrial (GICI) para análisis, simulación y emulación de sistemas dinámicos. En [39], se presenta el diseño de una plataforma para la educación en control con el enfoque ABP, donde se destacan las herramientas de análisis "Analys", simulación "Simweb" y experimentación remota con la plataforma "PI2", desarrolladas en *HTML*, *Javascript* y *Matlab*; y el trabajo de [2], en el cual se desarrolla una herramienta para el aprendizaje basado en problemas denominada JMat, basada en *Java* y *EJS*, con soporte multi–usuario y acceso remoto a *Matlab*, que presenta un esquema cliente/servidor con el protocolo TCP/IP. Actualmente el GICI se encuentra desarrollando el proyecto "*Entorno de aprendizaje basado en proyectos para sistemas de control*" [13] financiado por Colciencias, del cual el presente trabajo de investigación hace parte; en este proyecto se desarrollan herramientas de educación virtual, lúdicas para el aprendizaje de conceptos, recursos de experimentación local y remota, y herramientas de software libre locales y remotas para la simulación, emulación y análisis de sistemas dinámicos.

A continuación en la Tabla 2-1 se listan las características más relevantes de las aplicaciones de apoyo citadas.

Tabla 2-1 Características de las aplicaciones informáticas consultadas

Aplicación	Ilustración de Conceptos de Control	Análisis de Sistemas de Control	Simulación de Procesos	Análisis del PID	Operación del PID	Interactiva	Amigable	Software libre para cálculos matemáticos	Acceso a Internet
Ictools [20]	X	X	X			X			X
CCSdemo [21]	X	X	X			X			X
WCDAS [23]		X						X	X
Web Sisotool [25]				X					X
M-file [25]		X							X
Virtual Lab 1 [26]			X	X		X	X	X	X

Virtual Lab 2 [28]		X	X			X	X		
Remote Lab 1 [30]			X	X		X	X		X
Remote Lab 2 [31]			X	X		X	X		X
COCT [33]		X							
Rltool [34]		X						X	
ILM-PID [35]				X		X			
LSAD [37]		X				X		X	
Analys [39]		X							X
Simweb [39]			X	X		X	X		X
JMAT [2]		X				X	X		X

Se observa que la mayoría de las aplicaciones utilizan software comercial como Matlab para el procesamiento y cálculo matemático lo que limita su uso ya que se requiere de licencias actualizadas, cuentan con interfaces graficas de usuario poco amigables lo cual dificulta su manejo, son pocas las que ilustran conceptos de control y ninguna presenta aspectos prácticos de operación del controlador PID. Sin embargo se encuentran siempre disponibles a través de internet, cuentan con interfaces interactivas y animaciones que simulan procesos dinámicos lo que facilita el aprendizaje de sistemas de control.

2.4.1 El juego como estrategia didáctica para el aprendizaje

Una estrategia didáctica para el aprendizaje de conceptos es el juego, el cual según [40] se considera como una de las herramientas más efectivas para promover el aprendizaje activo y transferir el conocimiento, por su capacidad de simular la realidad, al ofrecer un escenario para cometer errores y aprender de ellos en la práctica. En [41] se considera que las características del juego: espontaneidad, competencia, participación, lo hacen dinamizador del aprendizaje, motivador de la reflexión y despiertan el interés hacia la asignatura.

En la educación en control se propone el uso de juegos para mejorar el aprendizaje; como el trabajo presentado en [6] y [8], donde Los autores hacen una revisión del uso del juego en la enseñanza y presentan un ejemplo llamado “Encontrando a T”, que como su nombre lo indica consiste en hallar el valor de una variable denominada T. Este juego es utilizado para aprender métodos de análisis de respuesta en frecuencia y cubrir tópicos como: Nyquist, Bode, margen de ganancia y frecuencia, y diseño de compensadores en adelanto y atraso.

La aplicación diseñada en [42] consiste de un juego educativo basado en *Matlab* denominado “SpacecraftRT”, el cual se implementó en un curso avanzado de control automático en la Universidad de Stuttgart en Alemania. Los autores describen los diferentes niveles del juego, los cuales requieren diferentes métodos de diseño del controlador; discuten aspectos de aplicación y los conceptos didácticos detrás del juego. Por último, comparan este juego con otros juegos educativos desarrollados para los cursos básicos de control.

En el año 2011 el International Journal of Engineering Education publicó un número dedicado al aprendizaje a través del juego en Ingeniería (Learning through Play in Engineering Education) en esta publicación se destaca el artículo Understanding Automatic Control Concepts by Playing Games, [43]. Este artículo presenta juegos

basados en simulación para aprender conceptos básicos de control. El tema es un controlador PID. Los juegos son parte de los resultados del proyecto piloto "AutoTECH: Automation Technicians Vocational Training Repository" desarrollado en el programa europeo Leonardo da Vinci. El objetivo de este proyecto es desarrollar y difundir paquetes de software innovadores para los técnicos y estudiantes de formación profesional en el campo de la automatización y control. Cada uno de los paquetes contiene recursos de aprendizaje tradicionales (teoría, por ejemplo, ejercicios, pruebas) y basados en la web (por ejemplo, simuladores interactivos, laboratorios remotos operables, juegos).

En la enseñanza de los cursos de pregrado de sistemas de control de la Universidad del Valle, el profesor desarrolla actividades lúdicas manos a la obra (Hands-On), las cuales han sido cuidadosamente diseñadas con el fin de introducir al concepto del tema y facilitar su comprensión por parte de los estudiantes [44]. Hands-On se consideran una nueva estrategia didáctica que motiva a los estudiantes a aprender de una manera divertida mientras juegan en el salón de clase.

2.5 Herramientas de Desarrollo y Arquitectura Software

La revisión previa muestra el desarrollo de aplicaciones de apoyo como: herramientas de análisis, animaciones interactivas, aplicaciones basadas en juegos y laboratorios remotos y virtuales de control. La mayoría de estas aplicaciones ofrecen recursos de simulación, animación e interfaces de usuario implementadas en *Java* para definir parámetros, analizar comportamientos y diseñar estrategias de control. Para realizar complejos cálculos matemáticos o simular el modelo de un proceso, se requiere el uso de software especializado como *Matlab* o *Scilab*. En cuanto al acceso, las aplicaciones están disponibles a través de Internet por medio de arquitecturas Cliente – Servidor y protocolos de comunicación como TCP/IP.

Como alternativas interesantes de software de desarrollo, se destaca el uso de herramientas libres como: *Easy Java Simulations EJS* para diseñar aplicaciones interactivas con capacidades gráficas de alto nivel; y de *SciLab* para brindar soporte matemático. Por tanto, a continuación se muestran los aspectos más relevantes de estas dos herramientas y de la arquitectura Cliente - Servidor.

2.5.1 Easy Java Simulations

Easy Java Simulations (simulaciones sencillas en Java), también conocido como *EJS*, es una herramienta gratuita y de código abierto desarrollada en Java, que ayuda a crear simulaciones dinámicas interactivas, con capacidades graficas de alto nivel, habitualmente con fines de enseñanza o aprendizaje [45]. EJS ha sido creado por Francisco Esquembre y es parte del proyecto Open Source Physics [46]. EJS presenta una interfaz de usuario sencilla, la cual contiene todas las herramientas necesarias para crear una simulación interactiva, como lo muestra la Figura 2-2.

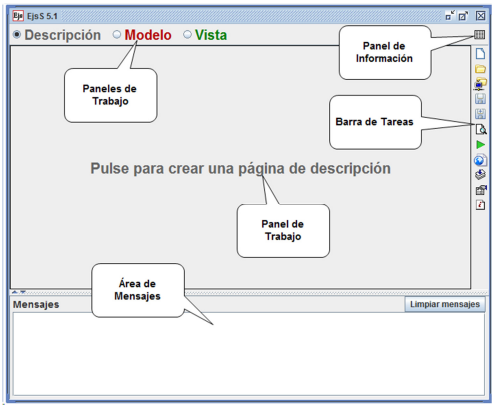


Figura 2-2 Interfaz de desarrollo de *Easy Java Simulations*

La barra de tareas de la derecha proporciona una serie iconos para crear, abrir, buscar en y guardar un archivo, configurar EJS, mostrar información sobre el programa y ayuda. Además, posee iconos para ejecutar una simulación y para empaquetar una o más simulaciones en un archivo comprimido tipo JAR o exportarlas como *Applets* en páginas web. Haciendo clic con el botón derecho en algunos de estos iconos se puede realizar otras acciones alternativas (pero relacionadas). La parte inferior de la interfaz contiene un área de mensajes donde EJS muestra mensajes de información. La parte central de la interfaz contiene los paneles de trabajo donde se construye la simulación. Desde un punto de vista práctico, EJS ayuda a desarrollar aplicaciones interactivas a través de tres paneles principales: Descripción, Modelo y Vista [47] (ver Figura 2-2).

El primer panel, Descripción es donde se realiza la introducción de la simulación, es decir, permite escribir cualquier información didáctica o pedagógica relacionada con la aplicación (por ejemplo, instrucciones sobre cómo utilizar la simulación, ejercicios para completar, cuestionarios, etc). EJS toma toda esta información y la transforma en páginas HTML.

El segundo panel, Modelo, está dedicado al proceso de modelado. Este panel se usa para crear e inicializar las variables que describen el modelo de la simulación, y es donde se escribe el algoritmo que describe cómo varían estas variables con el tiempo. Las variables se declaran e inicializan utilizando una sencilla tabla, el modelo de la simulación se puede escribir especificando el algoritmo en código Java o directamente expresando el modelo en ecuaciones diferenciales ordinarias (ODEs por sus siglas en inglés) utilizando el editor que ofrece EJS para escribir estas ecuaciones, el cual las resuelve utilizando diferentes métodos de solución y generando automáticamente el código necesario.

El tercer panel, Vista, está dedicado a la tarea de construir la interfaz gráfica de usuario de la simulación, que permitirá a los usuarios finales controlar la simulación y mostrar sus salidas. La interfaz se construye seleccionando elementos de unas paletas y añadiéndolos al denominado *Árbol de elementos* de la vista.

En el desarrollo de aplicaciones interactivas, EJS presenta las siguientes ventajas:

- La herramienta ofrece un conjunto integrado de métodos numéricos para resolver ecuaciones diferenciales ordinarias. El método elegido es llamado automáticamente por el editor incorporado de ODEs para resolver el modelo matemático que representa la animación.

- El paradigma Modelo, Vista y Controlador es implementado por EJS. Este proceso es oculto al programador, lo que significa que no se necesita conocimientos de programación avanzados para empezar a trabajar con la herramienta.
- Simplifica la creación de la "Vista" de una simulación, es decir la parte gráfica de la misma, proceso que habitualmente requiere de un gran nivel de conocimientos técnicos de programación gráfica.
- La herramienta opcionalmente retorna un *Applet Java* y archivos HTML, que permite una rápida publicación de la aplicación en un servidor Web.
- Por utilizar Java como lenguaje de desarrollo, EJS crea aplicaciones que son independientes y multiplataforma. Únicamente se requiere que el sistema tenga preinstalado el entorno de ejecución Java (JRE).

2.5.1.1 Arquitectura de las animaciones

EJS utiliza una versión simplificada del patrón de arquitectura de software *Modelo – Vista – Controlador MVC*, cuya filosofía se basa en que las animaciones desarrolladas se conforman por tres elementos: [47]

1. El *modelo* describe el proceso bajo estudio en términos de las variables que almacenan los diferentes estados posibles del proceso, y la relación entre estas variables (leyes físicas que rigen el comportamiento del proceso), expresadas mediante algoritmos informáticos.
2. El *controlador* define las acciones que el usuario puede realizar sobre el modelo de la simulación.
3. La *vista* es la representación del proceso simulado en un formato adecuado para interactuar, usualmente la interfaz gráfica de usuario.

La interacción entre el modelo, la vista y el controlador queda definida en la Figura 2-3.

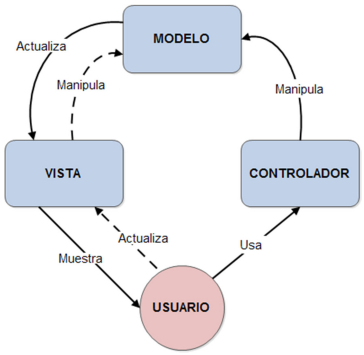


Figura 2-3 Relación entre los componentes del patrón MVC

El flujo de control que generalmente se sigue en MVC es el siguiente:

1. El usuario interactúa con la interfaz gráfica de usuario.
2. El controlador recibe (por parte de los objetos de la interfaz) la notificación de la acción solicitada por el usuario.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo. En EJS es posible proveer

cierta indirección entre la vista y el modelo, permitiendo al usuario actualizar el modelo de manera interactiva observando los cambios de manera inmediata.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

2.5.2 Scilab

Scilab es un software libre y de código abierto para el cálculo numérico que posee un potente entorno de computación para aplicaciones científicas y de ingeniería. Desarrollado por INRIA (Institut National de Recherche en Informatique et en Automatique) y la ENPC (École Nationale des Ponts et Chaussées) desde 1990, por Scilab Consortium dentro de la fundación Digiteo desde 2008, Scilab es ahora desarrollado por *Scilab Enterprises* desde julio 2012 [27].

La sintaxis del lenguaje es muy parecida a la sintaxis utilizada en Matlab. Por esta razón, muchas de las funcionalidades implementadas en Matlab se pueden encontrar en Scilab. Así, de manera similar a Matlab, este software amplía su capacidad mediante el uso de *toolbox* específicos denominados módulos. Como software matemático especializado, Scilab presenta las siguientes características:

1. Desde el punto de vista del software, Scilab es un lenguaje interpretado. En general, esto permite obtener procesos de desarrollo más rápidos, ya que el usuario accede directamente a un lenguaje de alto nivel, con un rico conjunto de funciones que ofrecen las diferentes librerías, además los usuarios pueden desarrollar sus propios módulos para resolver problemas particulares. Scilab es un software interoperable y por tanto es posible utilizarlo desde otras aplicaciones o integrar componentes escritos en otros lenguajes dentro de Scilab:

    - Puede interactuar directamente con otros lenguajes como: Java, Python, Fortran, C, C++.
    - Puede ser utilizado como motor de cálculo independiente por Java, Python, C, y aplicaciones externas en C++.
    - Puede ser llamado desde otro software como: Excel, ISight, LabView, ProActive, Dieta.

2. Desde el punto de vista de licencia, Scilab es un software libre de código abierto, cubierto bajo la licencia Cecill, [48]. El software se distribuye con el código fuente, por lo que el usuario puede acceder a aspectos más internos de Scilab. El consorcio Scilab ofrece versiones ejecutables para los sistemas operativos Windows, Linux y Mac OS.

3. Desde el punto de vista científico, Scilab incluye cientos de funciones matemáticas. Como posee un lenguaje de programación de alto nivel permite el acceso a estructuras de datos avanzadas, y funciones gráficas 2-D y 3-D. A continuación se listan algunas de estas funcionalidades:

    - Matemáticas y simulación: Para aplicaciones habituales de ingeniería y ciencias, se incluyen operaciones matemáticas y análisis de datos.
    - Visualización 2-D y 3-D: Funciones gráficas para visualizar, anotar y exportar datos, además de crear y personalizar varios tipos de gráficos y tablas.
    - Optimización: Algoritmos para resolver problemas de optimización continuos y

discretos restringidos y no restringidos.
- Estadística: Herramientas para llevar a cabo el modelado y análisis de datos
- Diseño y Análisis de Sistema de Control: Algoritmos y herramientas estándar para el estudio de los sistemas de control
- Procesamiento de Señal: Visualizar, analizar y filtrar señales en los dominios del tiempo y frecuencia.
- Desarrollo de Aplicaciones: Aumentar las funcionalidades nativas de Scilab y gestionar el intercambio de datos con aplicaciones externas.
- Xcos – Diseñar y simular sistemas dinámicos: Modelado de sistemas mecánicos, circuitos hidráulicos, y sistemas de control.

2.5.3 Arquitectura Cliente – Servidor

A pesar de su complejidad, cualquier red basada en Internet se puede considerar que consta de dos partes lógicas, llamadas Cliente y Servidor. El modelo Cliente – Servidor es una arquitectura de software que posee una infraestructura modular basada en mensajes para mejorar la usabilidad, flexibilidad, inter-operatividad y escalabilidad [49]. Históricamente es la arquitectura más importante, y continúa siendo la más ampliamente utilizada; en la Figura 2-4 se muestra la sencilla estructura sobre la que interaccionan los procesos clientes con el proceso servidor de manera individual, con el fin de acceder a los recursos compartidos que gestiona [50].

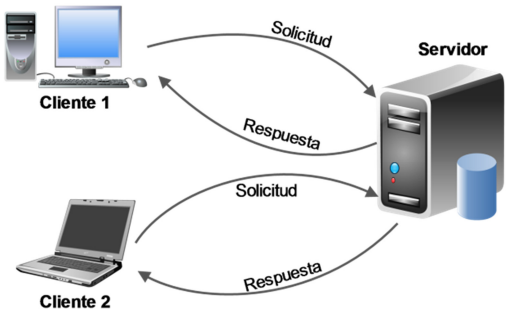


Figura 2-4 Modelo Cliente - Servidor

En términos generales, un cliente se define como un "*solicitador de servicios*" y un servidor se define como un "*proveedor de servicios*". En el entorno Cliente – Servidor el cliente está relacionado con las funciones de presentación de datos junto con actividades lógicas de validación y verificación local, mientras que el servidor desarrolla actividades relacionadas con búsqueda y recuperación de resultados en sus bases de datos. Desde un punto de vista más técnico los clientes (o programas que representan entidades que necesitan servicios) y los servidores (o programas que proporcionan servicios) son objetos separados y se comunican a través de una red de comunicaciones para realizar una o varias tareas de forma conjunta.

De manera más detallada, el cliente se caracteriza por desarrollar las siguientes funciones:

- Mantener y procesar todo el dialogo con el usuario.
- Manejo de pantallas.
- Menús e interpretación de comandos.
- Entrada de datos y validación.
- Procesamiento de ayudas y recuperación de errores.

El servidor por su parte se caracteriza por cumplir con las siguientes funciones:

- Acceso, almacenamiento y organización de datos.
- Actualización de datos almacenados.
- Administración de recursos compartidos.
- Ejecución de toda la lógica para procesar una transacción.
- Procesamiento común de elementos del servidor (datos, capacidad de CPU, almacenamiento, capacidad de impresión, manejo de memoria y comunicación).

Por otro lado, un protocolo de red está involucrado cuando un cliente se comunica con un servidor. El protocolo de Internet más ampliamente aceptado en la mayoría de las redes de comunicación es el TCP/IP por las siguientes características [47]:

- Protocolo orientado a conexión que maneja un canal de comunicación confiable.
- Garantiza que los mensajes lleguen al destino en forma correcta y ordenada.
- Asegura una transferencia de datos libre de falla, sin necesidad de que el programador se involucre en el desarrollo de: verificación de errores, reconocimientos (ACK), retransmisión, secuenciación de paquetes, time-outs, etc.

2.6 Resumen

En este capítulo se definieron los elementos y señales que conforman un sistema de control y el modelado, análisis y diseño como procedimiento para cumplir con unas especificaciones de desempeño deseadas. Se destacó la metodología de Aprendizaje Basado en Proyectos ABP como una de las mejores estrategias pedagógicas para educar por competencias a estudiantes de ingeniera. Se presentaron los aspectos generales de los dos cursos de sistemas lineales de control que se dictan en la Universidad del Valle, como: objetivos, metodología (donde se considera como estrategia pedagógica el ABP), resultados de aprendizaje y desarrollo de competencias transversales; además se listan los principales conceptos sobre los cuales se fundamentan los planes de estudio, y como son relacionados con las temáticas a tratar en los cursos.

Se presentó la revisión de alrededor 26 referencias bibliográficas sobre aplicaciones informáticas usadas como apoyo en la educación de sistemas de control, donde se listaron sus características más relevantes. Se observó que la mayoría de las aplicaciones utilizan software comercial como Matlab para el procesamiento y cálculo matemático lo que limita su uso ya que se requiere de licencias actualizadas, cuentan con interfaces graficas de usuario poco amigables lo cual dificulta su manejo, son pocas las que ilustran conceptos de control y ninguna presenta aspectos prácticos de operación del controlador PID. Sin embargo se encuentran siempre disponibles a través de internet, cuentan con interfaces interactivas y animaciones que simulan procesos dinámicos lo que facilita el aprendizaje de sistemas de control. Además, se evidenció que la tendencia es el desarrollo de: herramientas de análisis, animaciones interactivas, aplicaciones basadas en juegos y laboratorios remotos y virtuales de control; con interfaces de usuario implementadas en *Java*, uso de software como *Matlab* o *Scilab* para realizar cálculos matemáticos y acceso remoto por internet a través de arquitecturas Cliente – Servidor.

Como alternativas interesantes de software de desarrollo, se presentaron las herramientas libres, *Easy Java Simulations* para diseñar aplicaciones interactivas y *SciLab* para brindar soporte matemático.

3 DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE HERRAMIENTAS ANIMADAS

Con el objetivo de apoyar la asimilación de conceptos, el análisis y el diseño en la educación en sistemas de control del área de automática de la Universidad del Valle con el enfoque ABP, se desarrollaron tres tipos de aplicaciones informáticas. Con la finalidad de documentar el diseño y desarrollo de cada aplicación informática estas se agruparon en dos clases de herramientas:

- Herramientas Animadas
  1. Lúdicas animadas
  2. Herramienta de Análisis del Controlador PID HAPID

- Herramienta de Cálculo Matemático
  3. Herramienta de Análisis de Sistemas de Control HASC

Por tanto, a continuación se muestra el diseño y desarrollo de las herramientas animadas, las cuales consisten de animaciones interactivas desarrolladas con la herramienta EJS; mientras que la herramienta de cálculo matemático correspondiente a la aplicación HASC, debido al grado de complejidad en su concepción se documenta en el capítulo 4.

En este capítulo, inicialmente se presentan los requerimientos de funcionamiento y el diseño y desarrollo del sitio web que contiene las aplicaciones informáticas el cual se implementó en la plataforma PERI. Posteriormente se presentan los requerimientos de funcionamiento, de aspecto gráfico, el diseño y desarrollo, y las pruebas de funcionamiento de las Lúdicas Animadas y de la herramienta HAPID. Las Lúdicas Animadas están dirigidas a facilitar el aprendizaje de los conceptos más importantes de la teoría de los sistemas de control, y la aplicación HAPID, se utiliza como soporte para el análisis del controlador PID y aprendizaje de las diversas funciones de operación y de detección de fallas en lazos de control.

3.1 Sitio Web de las Aplicaciones Informáticas

Las aplicaciones informáticas desarrolladas deben estar disponibles en cualquier momento y ser accesibles desde cualquier lugar. Por lo anterior se desarrolló un *sitio web* que las contiene. El *sitio web* se implementó en la Plataforma de Educación Remota en Ingeniería PERI [38] de la Universidad del Valle. A continuación se describe la arquitectura software de esta plataforma.

3.1.1 Arquitectura software de la plataforma PERI

La arquitectura software de la plataforma PERI presenta tres niveles: Aplicación, Servidor y Cliente, como lo muestra la Figura 3-1. El nivel de aplicación cuenta con cuatro servidores los cuales implementan recursos de emulación, simulación y análisis de sistemas dinámicos. A través de un "*switch*" los servidores de aplicación se comunican con el nivel servidor el cual se encarga de administrar las políticas de acceso a usuarios, y el almacenamiento en la base de datos. Por medio del "*switch*", la plataforma se comunica con la red de la Universidad del Valle para finalmente tener acceso a Internet. El nivel cliente brinda acceso a los usuarios finales, quienes se autentican con el fin de utilizar los diversos recursos que dispone la plataforma.

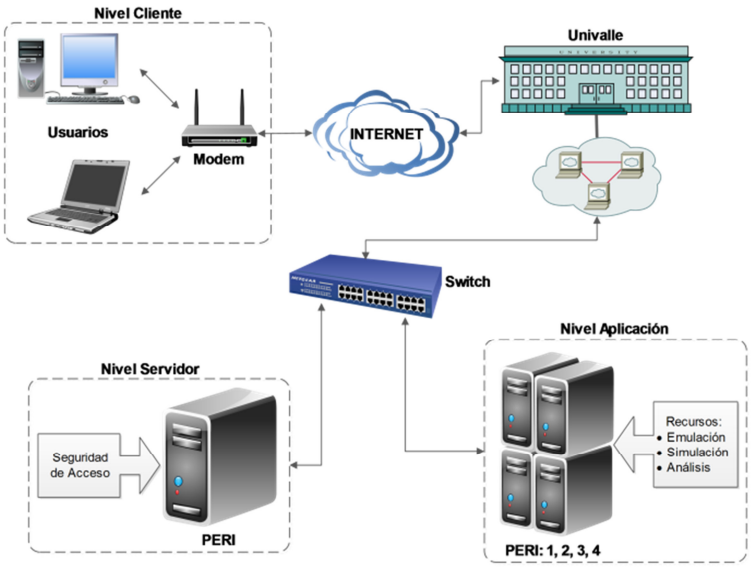


Figura 3-1 Arquitectura Hardware – Software de la plataforma PERI

3.1.2 Requerimientos

A continuación se presentan los requerimientos funcionales y no funcionales del *sitio web* que contiene las aplicaciones informáticas.

3.1.2.1 Requerimientos funcionales

- Debe mostrar un mensaje, con el fin de que el usuario verifique si el sistema cumple con los requisitos mínimos para ejecutar las aplicaciones.
- Debe presentar una página de configuración donde se listen estos requisitos y los pasos para cumplirlos.
- Debe presentar una breve introducción sobre cada aplicación
- Debe permitir un fácil acceso a las aplicaciones.

3.1.2.2 Requerimientos no funcionales

- Debe implementarse en el nivel servidor de la plataforma PERI, el cual cuenta con un servidor *PowerEdge T420 de Dell* que presenta las siguientes especificaciones técnicas:

  - Capacidad de Memoria: 8GB DDR3 RDIMM, 1600 MT/s, x4.
  - Almacenamiento: 1TB, 7.2K RPM SATA 3Gbps, 3.5in Hot-plug Hard Drive, x2.
  - Tarjeta de video: NIPCIE x16 NVIDIA de 128MB.
  - Unidad de DVD-ROM.
  - Procesador: Intel Xeon E5-2420 1.90GHz, 15M Cache, 7.2GT/s QPI.
  - Sistema operativo: Linux Ubuntu 12.04.

- Se debe utilizar el servidor web de código abierto Apache versión 2.
- Debe desarrollarse con los lenguajes HTML, CSS, y JavaScript.
- Las aplicaciones informáticas deben implementarse como *Applets de Java*.
- Se debe instalar y permitir la ejecución del plugin de Java.

3.1.3 Diseño y desarrollo

El *sitio web* consiste de páginas HTML implementadas en *Javascript* y en Hojas de estilo *CSS*, donde las aplicaciones informáticas se encuentran incrustadas como *Applets de Java*. El sitio web se encuentra disponible en la siguiente URL:

http://eieela.univalle.edu.co/ludicas_animadas/menu_ludicas.html

En la Figura 3-2 se muestra la interfaz principal del *sitio web* desarrollado.



Figura 3-2 Página principal del sitio web que contiene las Aplicaciones Informáticas

El *sitio web* presenta las siguientes características:

- Un menú horizontal principal con siete secciones: Lúdicas Animadas, HASC, HAPID, Introducción, Configuración Java, Formatos y Créditos.
- La sección Lúdicas Animadas, cuenta con un sub-menú que se despliega verticalmente con enlaces a 15 actividades Lúdicas Animadas.
- La sección HASC, consiste de una herramienta para el análisis de sistemas de control.
- La sección HAPID, consiste de una herramienta para el análisis del controlador PID.
- La sección Introducción, presenta las aplicaciones informáticas con una breve introducción sobre en qué consisten.
- La sección Configuración Java, establece los requisitos mínimos para ejecutar los *Applets* de las aplicaciones, plantea una serie de pasos que muestran la manera de cumplir estos requisitos.
- La sección Formatos, contiene enlaces con formatos de evaluación para descargar y diligenciar por parte de los usuarios. Los formatos buscan evaluar aspectos de funcionamiento, aprendizaje y desarrollo de habilidades transversales de las Lúdicas Animadas y de las aplicaciones HASC y HAPID.
- La sección Créditos, muestra aspectos de información de los autores e instituciones.

Para acceder a las aplicaciones, el *sitio web* implementa el esquema de navegación que se muestra en el diagrama de flujo de la Figura 3-3.

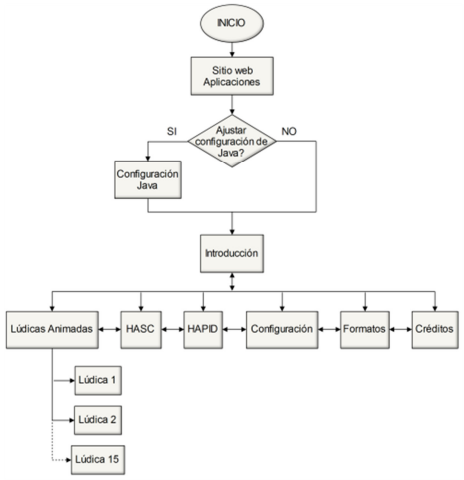


Figura 3-3 Esquema de navegación para acceder a las aplicaciones informáticas

Cuando se accede al *sitio web*, inicialmente se carga un mensaje emergente que pregunta si se desea ajustar la configuración de Java. En caso afirmativo se direcciona al usuario a la sección de Configuración Java. Después de realizar la verificación de requisitos, o si no se desea ajustar esta configuración, se transfiere a la sección de Introducción. Finalmente el usuario por medio del menú horizontal accede a las aplicaciones informáticas. La secuencia de navegación del sitio web implementado se muestra en las capturas de pantalla de la Figura 3-4.



Figura 3-4 Capturas del sitio web de las aplicaciones informáticas

3.2 Lúdicas Animadas

En las sesiones de tutoría de los cursos de pregrado de sistemas de control, al iniciar la clase el profesor propone a grupos de estudiantes el desarrollo de actividades lúdicas físicas o actividades manos a la obra Hands-On, con el fin de introducir al concepto del tema y facilitar su comprensión por parte de los estudiantes [44]. Sin embargo, para su implementación se deben tener en cuenta limitaciones como: espacio, materiales adicionales, número de estudiantes y tiempo; dependiendo del concepto a enseñar, la actividad puede ser compleja y de larga duración; además estas Hands-On por ser grupales, están limitadas como actividad de autoaprendizaje. A fin de evitar estas limitaciones se desarrollaron estas actividades lúdicas como animaciones interactivas, destinadas a los mismos objetivos de las sesiones Hands-On presentadas en [44], "*facilitar la comprensión de los conceptos de control*".

Las animaciones interactivas se desarrollaron con la herramienta EJS y se implementaron como *Applets de Java* en páginas web, con el fin de conocer su operación y facilitar el aprendizaje del concepto. Por lo anterior en esta sección inicialmente se presentan los requerimientos generales de funcionamiento y de aspecto gráfico de las animaciones y de las páginas web que las contienen. Posteriormente se muestra el diseño y desarrollo de la página web de las Lúdicas Animadas, la cual se caracteriza por estar conformada por los aspectos: Descripción, Funcionamiento, Actividades y Discusión para facilitar el aprendizaje del concepto y de 15 actividades Lúdicas Animadas implementadas con la herramienta de desarrollo EJS. Para cada Lúdica Animada se presenta el modelo matemático que simula su comportamiento dinámico, se muestra la vista de la animación y se establecen de manera general los resultados de aprendizaje esperados. Finalmente se muestra el uso y prueba de funcionamiento para una actividad Lúdica Animada, reportando el uso y prueba de las 15 Lúdicas Animadas en el documento: *Guía profesor animaciones*, el cual se encuentra disponible en el Anexo A; esta guía tiene como finalidad presentar los objetivos y resultados de aprendizaje esperados del uso de las Lúdicas Animadas como recurso de apoyo para el aprendizaje de conceptos en la educación en control.

3.2.1 Requerimientos

Teniendo en cuenta que en el desarrollo de las actividades Hands-On se utilizan elementos como: planos y cilindros de icopor, cuerdas y bandas elásticas, vasos plásticos, plastilina, etc., se agruparon los conceptos según los objetos dinámicos que conforman cada actividad y de esta manera se determinó el número de Lúdicas Animadas a desarrollar, como lo muestra la Tabla 3-1.

Tabla 3-1 Clasificación y número de Lúdicas Animadas a desarrollar

OBJETOS	CONCEPTO	LÚDICAS
Plano, rueda y pivote	Bucla típica	4
	Realimentación	
	Estabilidad y lugar geométrico de las raíces	
	Acción de control todo - nada	
Plano inclinado y rueda	Modelado discreto	2
	Acción de control PID	
Válvulas y tanques	Modelado continuo	5
	Respuesta temporal de primer orden	
	Estructura de control en cascada	
	Controlabilidad	
	Observabilidad	
Masa y resorte	Respuesta temporal de segundo orden	2
	Respuesta en frecuencia	
Cuerda y masa	Reconstrucción de señales	1
Cuerda, poleas y masa	Estructuras de control directo de disturbio	1
Total Animaciones Lúdicas a desarrollar		**15**

Por tanto, se requiere el desarrollo de 15 animaciones de actividades lúdicas para apoyar el aprendizaje de conceptos básicos de control en temáticas como: realimentación, modelado matemático, análisis temporal y frecuencial, estabilidad, compensación, reconstrucción de señales, estructuras de control, controlabilidad y observabilidad. Además para su correcto manejo, desarrollo e interpretación, se requiere de una página

web en donde se introduzca y defina el funcionamiento de la animación; y se planteen actividades y preguntas de discusión para el aprendizaje del concepto.

A continuación se presentan los requerimientos funcionales y no funcionales, y el esquema de las interfaces graficas de usuario de las animaciones y de la página web de las actividades Lúdicas Animadas.

3.2.1.1 Requerimientos funcionales

Para las animaciones se tienen:

- Permitir iniciar o detener la ejecución de la animación.
- Permitir restablecer a condiciones iniciales.
- Permitir modificar el valor de las variables del modelo matemático.
- Graficar el comportamiento temporal de las variables del modelo.
- Permitir el ajuste de los ejes de la respuesta gráfica.

Para la página web se tiene:

- La descripción del funcionamiento de la animación, y las actividades y preguntas de discusión planteadas, se deben consultar por medio de una vista por pestañas.

3.2.1.2 Requerimientos no funcionales

- Se debe utilizar el software EJS - versión 5.0, como herramienta de desarrollo de las animaciones.
- Se debe utilizar el entorno de ejecución Java (JRE), versión 1.7 o superior.
- Se debe utilizar el *framework jQuery UI* [51], para implementar la vista por pestañas en la página web de cada lúdica animada.

3.2.1.3 Interfaz gráfica de usuario

La animación de cada actividad lúdica y la descripción o introducción, el funcionamiento, las actividades y las preguntas de discusión planteadas, se deben organizar y distribuir en una página web, como lo muestra la Figura 3-5.

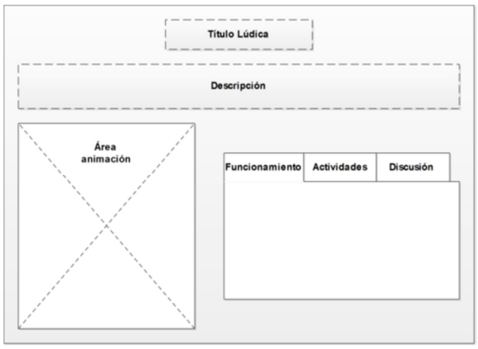


Figura 3-5 Organización y presentación general de la página web de cada Lúdica Animada

En cuanto a la interfaz gráfica de usuario de las animaciones, esta debe contener dos paneles, un panel principal y un panel emergente conformados por las siguientes áreas:

Área 1: Representación animada del sistema, el cual debe conformarse por arreglos de objetos dinámicos simples como:

- Ruedas, pivotes y planos
- Válvulas y tanques de nivel
- Masas y resortes
- Cuerdas, poleas y masas

Área 2: Organización de elementos de Interacción como:

- Botones para ejecutar acciones
- Barras de deslizamiento para modificar variables
- Botones para seleccionar entre diferentes opciones
- Campos para el ingreso y registro de variables

Área 3: Visualización de la respuesta temporal del comportamiento del sistema dinámico.

Estas áreas se deben organizar y presentar de acuerdo al esquema específico mostrado en la Figura 3-6.

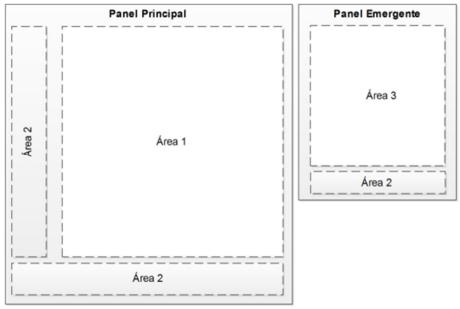


Figura 3-6. Esquema de la Interfaz gráfica de usuario de cada animación

3.2.2 Diseño y desarrollo

Las actividades lúdicas consisten de animaciones interactivas desarrolladas con la herramienta EJS. Se concibieron 15 Lúdicas Animadas, las cuales se caracterizan por:

- Presentar animaciones de sistemas dinámicos, conformados por arreglos de elementos simples como:
  - Ruedas, pivotes y planos
  - Válvulas y tanques de nivel
  - Masas y resortes
  - Cuerdas, poleas y masas

- Contener elementos de interacción para ajustar parámetros y manipular variables de los sistemas dinámicos, como:
  - Botones de acción
  - Barras de deslizamiento
  - Botones y casillas de selección
  - Campos numéricos para el ingreso y registro de variables

- Gráficas que muestran el comportamiento temporal de los sistemas dinámicos.

En la Tabla 3-2 se muestra el nombre asignado a cada lúdica animada desarrollada según los elementos dinámicos que la conforman, relacionando cada lúdica con el concepto que se quiere introducir.

Tabla 3-2. Lúdica Animada vs. Concepto

No.	LÚDICA	CONCEPTO
1	Plano y Rueda	Sistemas de control de lazo abierto vs. de lazo cerrado
2	Válvula y Tanque	Modelado
3	Plano Inclinado Manual	Modelado de Sistemas Discretos
4	Cuerda y Masa	Reconstrucción de Señales
5	Plano, Rueda y Disturbio	Sistemas en red abierta y en red cerrada
6	Válvulas y Tanque	Sistemas de Primer Orden
7	Masa y Resorte	Sistemas de Segundo Orden
8	Plano y Ruedas	Acción de Control Todo – Nada
9	Plano Inclinado Automático	Acción de Control PID
10	Plano, Rueda y Pivote	Estabilidad y Lugar Geométrico de las Raíces
11	Masa, Resorte y Circunferencia	Respuesta en Frecuencia
12	Cuerdas, Poleas y Masa	Estructuras de Control 1 “Cancelación Disturbio”
13	Tanques en Serie	Estructuras de Control 2 “Control en Cascada”
14	Tanques en Paralelo 1	Controlabilidad
15	Tanques en Paralelo 2	Observabilidad

Con el fin de conocer su funcionamiento y facilitar el aprendizaje del concepto, las animaciones interactivas se implementaron como *Applets* en páginas web. Por tanto la página web desarrollada para cada lúdica animada se caracteriza por presentar el esquema general que se muestra la Figura 3-7, en la cual se han diferencia tres áreas.

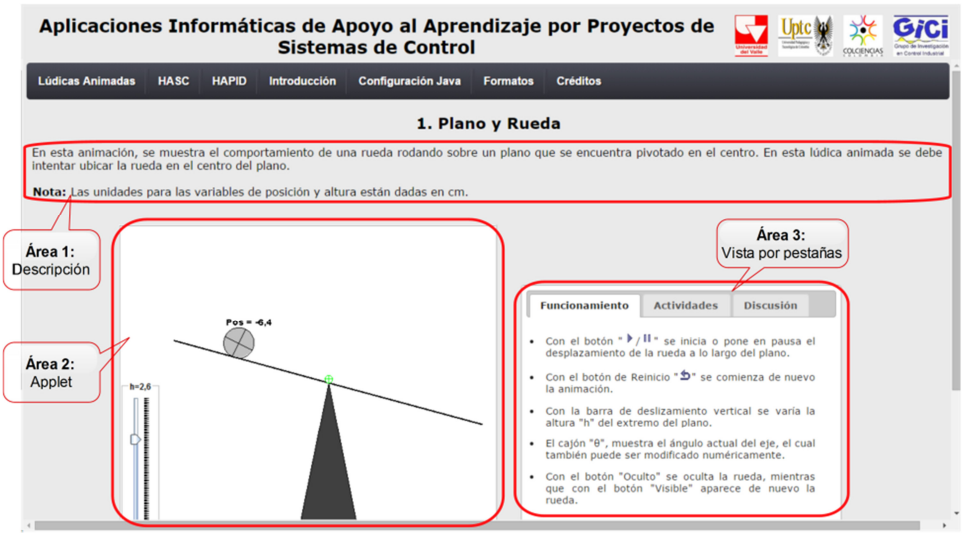


Figura 3-7 Página web de la Lúdica Animada Plano y Rueda

Área 1: En donde se introduce y describe el comportamiento general de la lúdica animada.

Área 2: En donde se implementa la animación de la lúdica, embebida en la página web como un *Applet de Java*.

Área 3: En donde se implementa una vista por pestañas conformada por los aspectos:

- Funcionamiento: En donde se describe la función de cada elemento de la interfaz gráfica de usuario de la animación.
- Actividades: En donde se establece el procedimiento que se debe realizar en la animación con el fin de obtener diferentes comportamientos.
- Discusión: En donde se plantean preguntas de discusión para analizar los comportamientos obtenidos.

El diseño concebido de la página web de las Lúdicas Animadas representa una estrategia didáctica que busca facilitar el aprendizaje del concepto, como lo muestra el diagrama de flujo de la Figura 3-8.

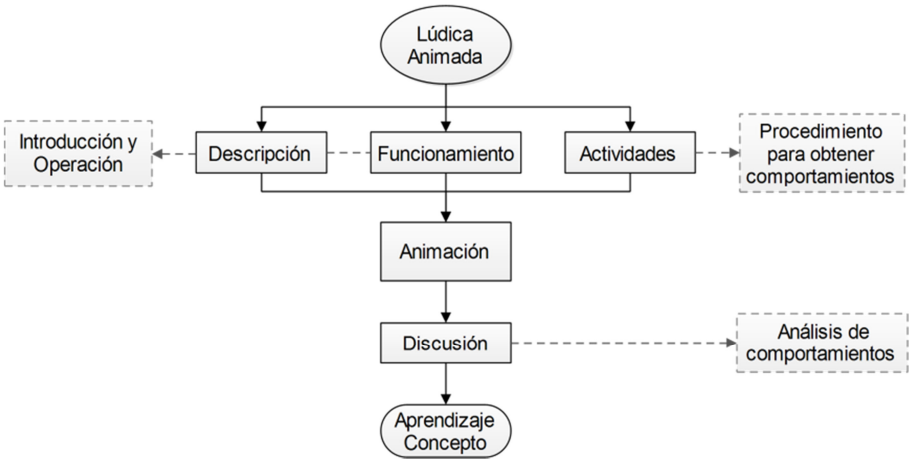


Figura 3-8 Estrategia didáctica para el aprendizaje de conceptos en las Lúdicas Animadas

Con la Descripción y el Funcionamiento se familiariza al estudiante con el entorno de la animación, el desarrollo de las Actividades permiten obtener comportamientos los cuales son analizados por medio de las preguntas de Discusión, facilitando de esta manera el aprendizaje del concepto. Sin embargo el diseño concebido, no plantea de manera explícita los objetivos y resultados de aprendizaje esperados; por lo anterior, para uso de las Lúdicas Animadas por un docente, se generó el documento: *Guía profesor animaciones*, el cual se encuentra disponible en el Anexo A; en esta guía se presentan los aspectos generales, requerimientos para su carga y los objetivos y resultados de aprendizaje esperados del uso de las Lúdicas Animadas como recurso de apoyo para el aprendizaje de conceptos en la educación en control.

En cuanto a las animaciones interactivas, su desarrollo en EJS presenta dos características fundamentales [47]:

1. Se implementan en dos pasos: a) Se define el *modelo* que simula el comportamiento del sistema, y b) Se construye la *vista* de la simulación para mostrar los estados del modelo y las respuestas a los cambios interactivos realizados por el usuario.

   A manera de ejemplo en la Figura 3-9 se muestra la secuencia de implementación en EJS de la animación correspondiente a la Lúdica 1: Plano y Rueda.

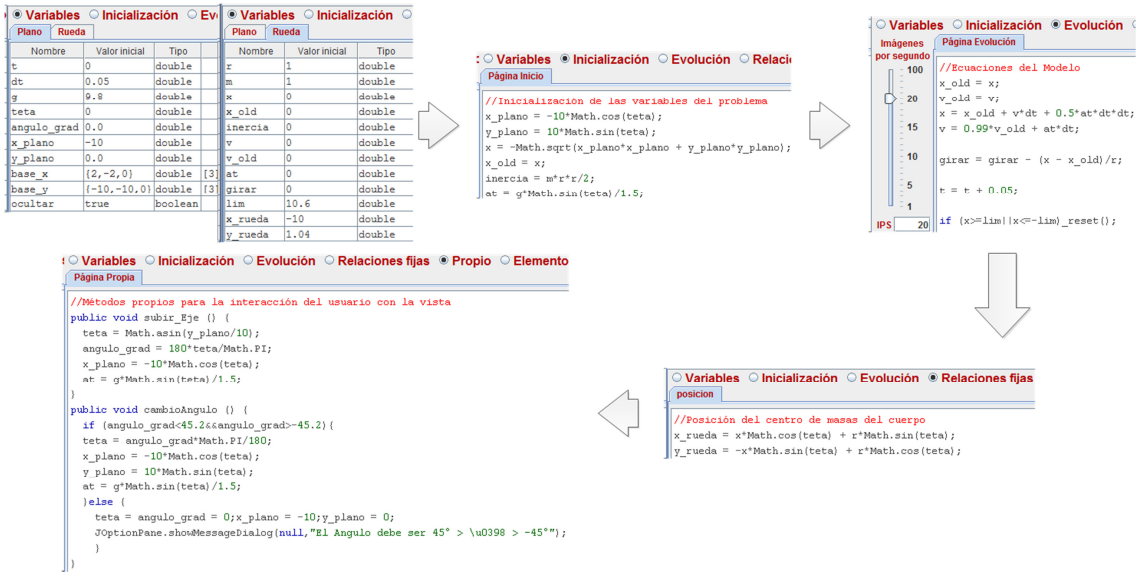


Figura 3-9 Secuencia de implementación de la lúdica 1 en EJS

En el panel de Variables se define el valor y tipo de variables necesarias. En el panel de Inicialización se establecen las condiciones iniciales de las variables. En el panel de Evolución se implementan las ecuaciones matemáticas del modelo. En el panel de Relaciones fijas se definen valores fijos de las variables, las cuales se actualizan continuamente. En el panel Propio se definen los métodos que actualizan las variables del modelo cuando el usuario interactúa con la vista de la animación. Posteriormente se construye la vista de la animación organizando los elementos gráficos en una estructura en forma de árbol, como lo muestra la Figura 3-10.

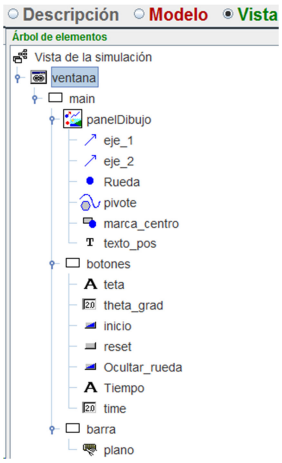


Figura 3-10 Estructura en forma de árbol para implementar la Vista de la lúdica 1 en EJS

2. Su ejecución se basa el algoritmo de simulación que se muestra en la Figura 3-11.

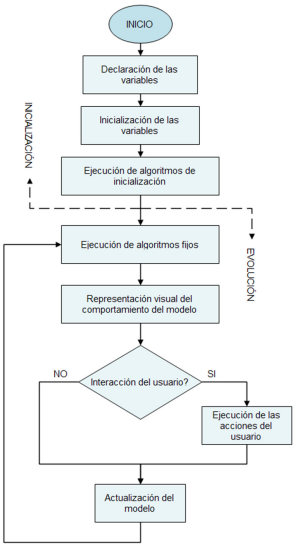


Figura 3-11 Algoritmo de simulación de las Lúdicas Animadas

El algoritmo se divide en dos etapas de ejecución: la inicialización y la evolución. En la primera etapa se declaran las variables y se inicializa el código necesario para establecer las condiciones iniciales del sistema. La segunda etapa cubre los algoritmos que definen la dinámica del sistema durante su evolución, y es donde se establece el estado gráfico del sistema. En esta etapa, la vista de la simulación continuamente transforma los valores actuales de las variables del modelo en una representación gráfica, los cuales también se actualizan por la interacción del usuario.

A continuación se muestra el análisis realizado para definir el *modelo* matemático, se presenta la *vista* de la animación y se establecen de manera general los resultados de aprendizaje esperados del uso de las 15 Lúdicas Animadas desarrolladas.

3.2.2.1 Lúdica 1: Plano y Rueda

Para introducir el concepto de *Sistemas de control de lazo abierto vs. de lazo cerrado*, se requiere de una animación que le permita al estudiante:

- Observar el comportamiento de un sistema estable e inestable
- Identificar los elementos y señales que componen la bucla típica de un sistema de control
- Observar el desempeño de un sistema de control con y sin realimentación

Para ello, se diseñó una animación que muestra el comportamiento de una rueda rodando sobre un plano que se encuentra pivotado en el centro, donde de manera interactiva se manipula la altura del plano, con el fin de ubicar la rueda en el centro. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.1.1 Modelo

La definición del modelo matemático de la animación se basa en analizar el movimiento de un cuerpo de simetría circular (rueda) de radio *r* y masa *m*, que se desplaza sin deslizarse por una superficie plana que se encuentra pivotada en el centro, como lo muestra la Figura 3-12.

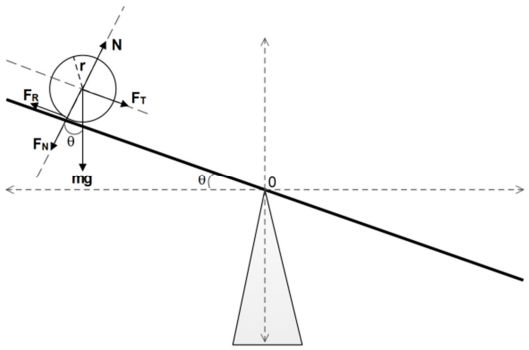


Figura 3-12 Diagrama de fuerzas de la rueda en el sistema plano y pivote

Para simplificar la definición del modelo, se analiza el plano inclinado que se forma en el segundo cuadrante del sistema plano y pivote. El plano inclinado consiste de una superficie plana que forma un ángulo θ con la horizontal, en el cual el sistema de coordenadas está orientado positivamente con el eje *X* apuntando en la dirección de la superficie del plano inclinado, y el eje *Y* apunta según la normal a dicho plano. Las fuerzas que actúan sobre la rueda situada en el plano inclinado son [45] [52]:

- El peso que es consecuencia de la masa *m* que posee el cuerpo apoyado en el plano inclinado con magnitud *mg* y ángulo θ respecto a la normal del plano. Las componentes en las direcciones de los ejes coordenados son:

$$F_T = mgsen(\theta), en\ el\ eje\ X$$

$$F_N = -mgcos(\theta), en\ el\ eje\ Y$$

- La fuerza normal *N* ejercida por la superficie del plano sobre el cuerpo como reacción a F_N y de sentido opuesto a esta.
- La fuerza de rozamiento F_R en el punto de contacto entre la rueda y el plano.

Aplicando la segunda ley de Newton en la dirección de la superficie del plano se obtiene:

$$ma = mgsen\theta - F_R \quad (1)$$

Las ecuaciones necesarias para el análisis son:

- Movimiento de rotación: $F_R r = I\theta$ (2)
- Relación entre el movimiento de traslación y rotación: $a = \theta r$ (3)
- Momento de Inercia para un cilindro: $I = \frac{mr^2}{2}$ (4)

Al combinar (3) y (4) en (2), se obtiene:

$$F_R = \frac{ma}{2} \quad (5)$$

Reemplazando (5) en (1) se obtiene por último la ecuación de la aceleración tangencial:

$$a = \frac{gsen\theta}{1.5} \quad (6)$$

La posición y velocidad del centro de masa de la rueda medida a lo largo de la superficie del plano inclinado, se calculan con las ecuaciones del *movimiento uniformemente acelerado*:

$$x(t) = x(0) + v(0)t + \frac{1}{2}at^2 \tag{7}$$

$$v(t) = v(0) + at \tag{8}$$

Adicionalmente, la posición del centro de masa y el ángulo de giro de la rueda en cada instante vienen dados por las siguientes ecuaciones:

$$\big(x(t), y(t)\big) = (x(t)cos\theta + rsen\theta, -x(t)sen\theta + rcos\theta) \tag{9}$$

$$\alpha(t) - \alpha(0) = -\frac{x(t) - x(0)}{r} \tag{10}$$

En EJS las ecuaciones (7), (8) y (10), deben expresarse en forma incremental, para ello se evalúan los valores de estas variables en un instante *t* y en el instante posterior *t + dt*, siendo *dt* el intervalo de tiempo que transcurre en cada paso de la evolución de la animación [45], obteniendo:

$$x(t + dt) = x(t) + v(t)dt + \frac{1}{2}a\,dt^2 \tag{11}$$

$$v(t + dt) = v(t) + a\,dt \tag{12}$$

$$\alpha(t + dt) - \alpha(t) = -\frac{x(t + dt) - x(t)}{r} \tag{13}$$

3.2.2.1.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema plano y pivote de la Figura 3-12, como lo muestra la Figura 3-13.

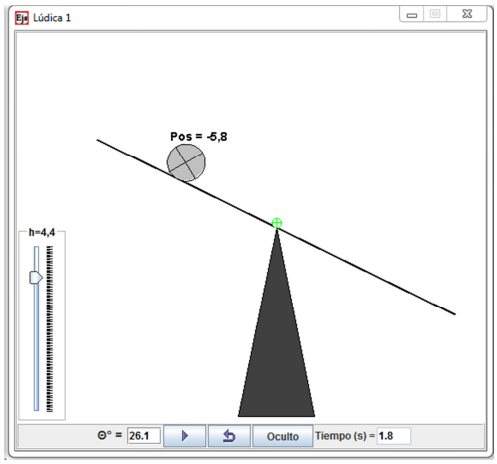


Figura 3-13 Vista de la lúdica 1: Plano y Rueda

Por medio del botón " / " se inicia o pone en pausa el desplazamiento de la rueda a lo largo del plano y con el botón "" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se varía la altura "h" del extremo del plano, mientras que en el campo "θ" se muestra o modifica el ángulo actual del eje. Finalmente con el botón "Oculto" se oculta la rueda.

En la Tabla 3-3 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-3 Unidades y valores de las variables en la Lúdica 1

Variable	Descripción	Unidades	Valor	
			Min.	Max.
x	Desplazamiento de la rueda	m	-10	10
h	Altura del plano	m	-7	7
θ	Angulo de inclinación del plano	Grados	-44.4	44.4
m	Masa de la rueda	kg	1	
r	Radio de la rueda	m	1	
g	Gravedad	m/s^2	9.8	

3.2.2.1.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe manipular la altura de inclinación y observar el comportamiento de la rueda a lo largo del plano. Inicialmente observará que el sistema presenta un comportamiento inestable dado que para una inclinación dada, la rueda se desplaza sin detenerse. Posteriormente buscando posicionar la rueda en el centro del plano, determinará que se trata de un sistema de control de posición; donde la planta a controlar es la rueda y el plano, la salida controlada es la posición de la rueda, la variable manipulada es el ángulo de inclinación del plano, el elemento de realimentación son los ojos del usuario y el controlador él mismo (cerebro); y el actuador es la mano con el Mouse. Concluirá que la realimentación visual permite ubicar la rueda en el centro del plano, estabilizándose el sistema en lazo cerrado.

3.2.2.2 Lúdica 2: Válvula y Tanque

Para introducir el concepto de *Modelado matemático de un sistema*, se requiere de una animación que le permita al estudiante:

- Observar el comportamiento de un sistema lineal y no lineal

Para ello, se diseñó una animación que muestra el comportamiento del nivel en un tanque, donde de manera interactiva se manipula la apertura de la válvula graficando la evolución del nivel a través del tiempo en una ventana emergente. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.2.1 Modelo

La definición del modelo matemático de la animación se basa en analizar el comportamiento del nivel en un tanque, como lo muestra la Figura 3-14.

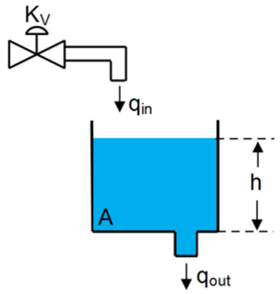


Figura 3-14 Sistema del nivel en un tanque

Dónde, $q_{in}(t)$ corresponde al flujo de entrada, $q_{out}(t)$ al de salida, $h(t)$ es el nivel de agua, K_V

es la ganancia unitaria de la válvula y *A* es el área transversal del tanque la cual se considera constante. El modelo matemático se obtiene de un balance de materia: la integral del líquido que entra menos el que sale es el que se acumula en el tanque [53]:

$$Ah(t) = \int_{-\infty}^{t} \left(K_V q_{in}(t) - q_{out}(t)\right) dt \tag{14}$$

Para relacionar el flujo de salida con el nivel del agua, se usa la ley física que describe este fenómeno: el flujo turbulento $q_{out}(t)$ a través de una restricción es proporcional a la raíz cuadrada del nivel $h(t)$:

$$q_{out}(t) = k\sqrt{h(t)}, \tag{15}$$

donde *k* es una constante de proporción. Si se deriva con respecto al tiempo a ambos lados de la ecuación (14), se obtiene la ecuación que describe la dinámica del sistema, donde se observa que se ha omitido la constante K_V por considerarse de valor unitario:

$$A\frac{dh(t)}{dt} = q_{in}(t) - k\sqrt{h(t)}\ , h(t_0) = h_0, \tag{16}$$

Al analizar esta ecuación en estado estable se obtiene:

$$h(t) = \left(\frac{q_{in}(t)}{k}\right)^2$$

Se observa que para valores pequeños de *k* con un flujo de entrada máximo, se obtienen valores muy grandes de nivel de agua en el tanque. Por tanto con *k* se restringe y regula el comportamiento del nivel, entonces se asumió un nivel límite de agua en el tanque del 90% para un flujo de entrada máximo del 100%, obteniendo un valor de *k = 1.0541*.

3.2.2.2.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema de la Figura 3-14, como lo muestra la Figura 3-15.

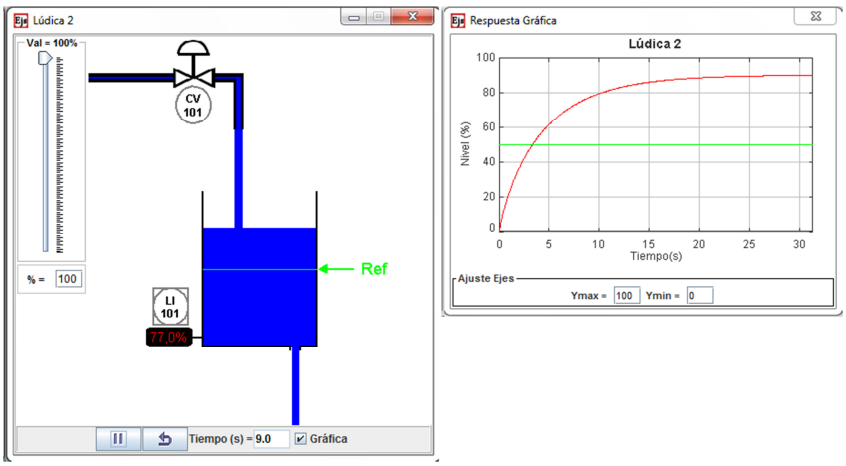


Figura 3-15 Vista de la lúdica 2: Válvula y Tanque

Por medio del botón " / " se inicia o pone en pausa la ejecución de la animación y con el botón "" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se manipula el porcentaje de apertura de la válvula CV101, el cual también puede ser modificado por medio del campo "%". Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico del nivel "h" a través del

tiempo; además esta gráfica cuenta con campos numéricos que permiten ajustar la visualización de la señal.

En la Tabla 3-4 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-4 Unidades y valores de las variables en la Lúdica 2

Variable	Descripción	Unidades	Valor	
			Min.	Max.
h	Nivel en el tanque	%	0	100
A	Área de sección transversal del tanque	m^2	3	
K	Coeficiente de proporción del flujo turbulento	-	1.0541	
K_V	Ganancia de la válvula	-	1	

3.2.2.2.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe manipular la apertura de la válvula y observar la evolución temporal del nivel en la ventana emergente. Inicialmente obtendrá aperturas con variaciones diferentes entre el 0 y 100%, y al analizar los comportamientos obtenidos determinará que el sistema presenta un comportamiento no lineal, dado que la evolución del nivel en velocidad de respuesta y valor de ganancia estática es diferente para cada apertura. Posteriormente realizará pequeñas variaciones de la apertura alrededor de un punto de operación, y al analizar los comportamientos obtenidos determinará que bajo esta condición, el sistema presenta un comportamiento lineal, dado que la evolución del nivel en velocidad de respuesta y valor de ganancia estática es muy similar para cada variación.

3.2.2.3 Lúdica 3: Plano Inclinado Manual

Para introducir el concepto de *Modelado de sistemas discretos*, se requiere de una animación que le permita al estudiante:

- Observar la evolución del error en un sistema, con y sin una acción de control
- Determinar la ecuación de diferencias que describe la acción de control

Para ello, se diseñó una animación que muestra el comportamiento de una rueda rodando sobre un plano inclinado, el cual se encuentra apoyado sobre un plano horizontal en un extremo, donde de manera interactiva se manipula la altura del extremo libre buscando que la rueda alcance una posición deseada en el plano horizontal. Cuenta con una tabla de datos que registra valores de altura y posición. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.3.1 Modelo

La definición del modelo matemático de la animación se basa en analizar el desplazamiento (sin deslizarse) de una rueda por un plano inclinado, como lo muestra la Figura 3-16.

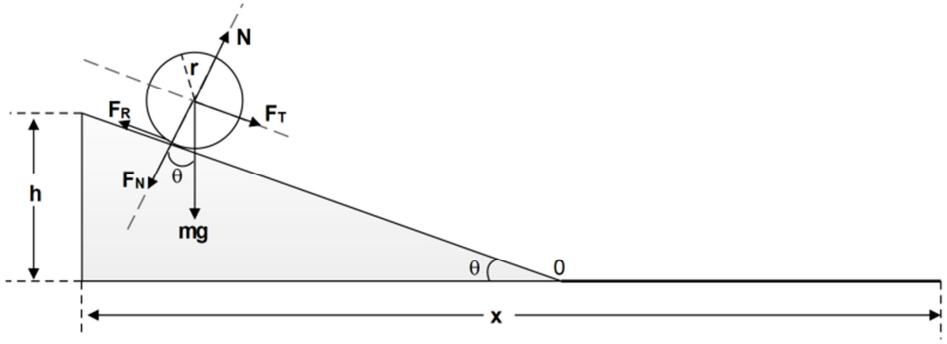


Figura 3-16 Diagrama de fuerzas de la rueda en el sistema plano inclinado

El análisis es similar al realizado en la Lúdica 1: Plano y Rueda, donde se determinaron las ecuaciones para la aceleración (6), posición (7) velocidad (8) y ángulo de giro (9) de la rueda. En este caso se implementa un sistema de control de posición que permite ajustar la altura del extremo del plano en función del valor del error para cada lanzamiento de la rueda, como lo muestra el esquema de la Figura 3-17.

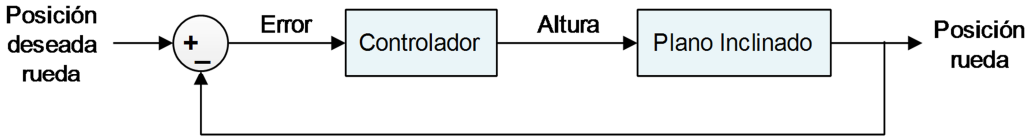


Figura 3-17 Esquema del control de posición para el sistema del plano inclinado

En el controlador se implementa una acción de control integral la cual genera una señal continua de la siguiente forma:

$$u(t) = k_i \int_0^t e(t)dt, \tag{17}$$

donde k_i se denomina ganancia integral. Para implementar esta ecuación en EJS, esta debe expresarse como ecuaciones de diferencias, para ello se utiliza la aproximación de Euler de la integral [54], obteniendo:

$$h_C(k+1) = h_C(k) + k_i e(k), \tag{18}$$

donde $h_C(k+1)$ corresponde a la altura siguiente calculada, siendo $hc(k)$ la altura actual, el error $e(k)$ es la diferencia entre la posición deseada y la alcanzada por la rueda, y k_i es el parámetro de ajuste.

3.2.2.3.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el plano inclinado de la Figura 3-16, como lo muestra la Figura 3-18.

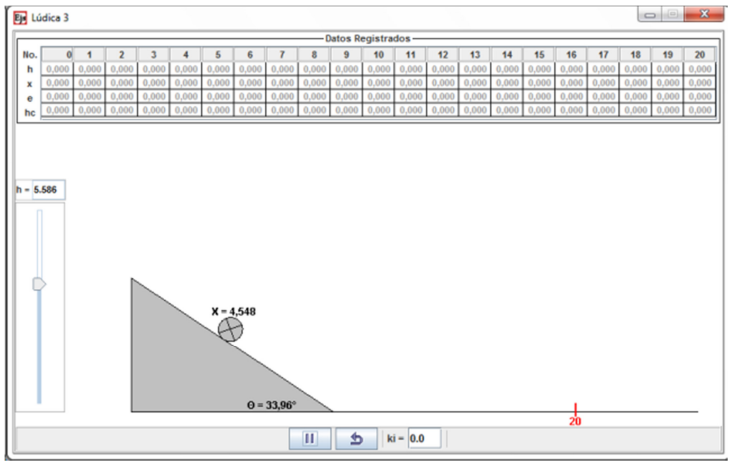


Figura 3-18 Vista de la lúdica 3: Plano Inclinado Manual

Por medio del botón " / " se inicia o pone en pausa el desplazamiento de la rueda y con el botón "" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se varía la altura "h" del extremo del plano inclinado, la cual también puede ser modificada por medio del campo "h" y en el campo "ki" se ajusta el valor de la ganancia integral. Finalmente en la tabla "Datos Registrados", se registra el valor de las variables: Altura actual h, Distancia alcanzada x, Error e (distancia Deseada - Alcanzada) y Altura calculada hc por la aproximación de Euler de la integral.

En la Tabla 3-5 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-5 Unidades y valores de las variables en la Lúdica 3

Variable	Descripción	Unidades	Valor	
			Min.	Max.
x	Desplazamiento de la rueda	m	-10	15
h	Altura del plano	m	0	9
θ	Angulo de inclinación del plano	Grados	0	64.16
m	Masa de la rueda	kg	1	
r	Radio de la rueda	m	0.5	
g	Gravedad	m/s^2	9.8	

3.2.2.3.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe manipular la altura del extremo libre buscando que la rueda rodando por el plano inclinado, alcance una distancia deseada en el plano horizontal. Inicialmente irá ajustando la altura del extremo del plano a partir de la observación del error de posición obtenido para cada lanzamiento. Posteriormente ajustará la altura a partir de la altura calculada por la aproximación de Euler (ver ecuación (18)). Para ambos casos analizará la forma de la evolución del error, observando que la altura calculada por la ecuación de Euler de la integral logra un mejor comportamiento.

3.2.2.4 Lúdica 4: Cuerda y Masa

Para introducir el concepto de *Reconstrucción de señales*, se requiere de una animación que le permita al estudiante:

- Observar el efecto que tiene la selección de la frecuencia de muestreo en la reconstrucción de señales

Para ello, se diseñó una animación que muestra el comportamiento de una masa amarrada a una cuerda que gira sobre su eje a un ritmo constante, donde de manera interactiva se seleccionan los instantes de visualización de la posición de la masa. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.4.1 Modelo

La definición del modelo matemático de la animación, considera el manejo de dos frecuencias de muestreo, la frecuencia natural f_n que define el giro constante de la masa y una frecuencia variable f_s que define los instantes de visualización del sistema. El giro de la masa, se basa en representar una oscilación senoidal en la circunferencia trigonométrica, como lo muestra la Figura 3-19.

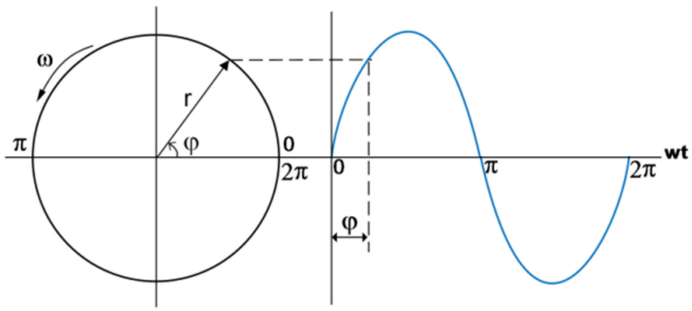


Figura 3-19 Oscilación senoidal y circunferencia trigonométrica

El vector *r* gira con una velocidad angular ω, igual a:

$$\omega = 2\pi f_n \tag{19}$$

En la animación las coordenadas del vector representan la posición de la masa, las cuales vienen dadas por:

$$\big(x(t), y(t)\big) = \big(r\cos(\omega t), r sen(\omega t)\big) \tag{20}$$

La f_s, consiste de una frecuencia variable que es *k-veces* la frecuencia natural del sistema:

$$f_s = k f_n, \tag{21}$$

donde *k* define el número de instantes de visualización del sistema.

3.2.2.4.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en la circunferencia de la Figura 3-19, como lo muestra la Figura 3-20.

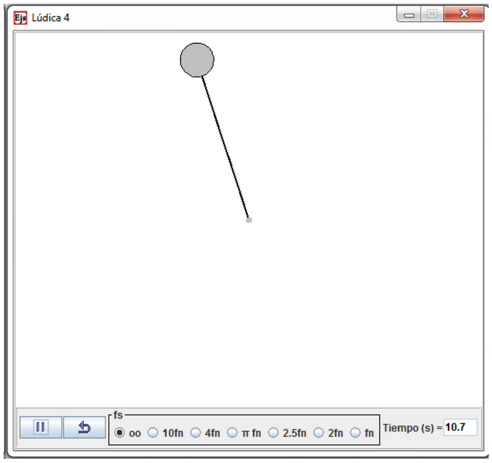


Figura 3-20 Vista de la lúdica 4: Cuerda y Masa

Por medio del botón " / " se inicia o pone en pausa la ejecución de la animación y con el botón "" se reinicia de nuevo la animación. Seleccionando una de las opciones del recuadro "fs", se oculta con un plano gris la visualización de la animación.

En la Tabla 3-6 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-6 Unidades y valores de las variables en la Lúdica 4

Variable	Descripción	Unidades	Valor	
			Min.	Max.
f_s	Frecuencia de visualización de la masa	Hz	0.5	∞
f_n	Frecuencia natural de giro de la masa	Hz	0.5	
r	Longitud de la cuerda	m	10	

3.2.2.4.3 Resultados de aprendizaje esperados

En esta lúdica animada se considera la situación de intentar reconstruir una señal continua original *y(t)* de frecuencia natural *fn*, a partir de su señal muestreada *y*(t)*. Para esto, el estudiante debe seleccionar diferentes frecuencias de visualización f_s y observar la posición de la masa para cada caso. De los comportamientos observados determinará que la frecuencia de visualización debe ser lo suficientemente alta para determinar que la masa gira y no está quieta, y concluirá que para recuperar una señal a partir de sus muestras, se debe muestrear por lo menos al doble de la mayor frecuencia de la señal como lo plantea el *teorema del muestreo*.

3.2.2.5 Lúdica 5: Plano, Rueda y Disturbio

Para analizar el comportamiento de los *Sistemas en red abierta y en red cerrada*, se requiere de una animación que le permita al estudiante:

- Observar el comportamiento de un sistema estable e inestable
- Observar el desempeño ante disturbios de un sistema, con y sin realimentación

Para ello, se diseñó una animación que muestra el comportamiento de una rueda rodando sobre un plano que se encuentra pivotado en el centro, donde de manera interactiva se

manipula la altura del plano, buscando mantener la rueda en el centro ante el efecto de un disturbio que la desplaza periódicamente. La animación cuenta con una ventana emergente donde se registra gráficamente la evolución de la posición de la rueda a través del tiempo. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.5.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis realizado en la Lúdica 1: Plano y Rueda, donde se determinaron las ecuaciones para la aceleración (6), posición (7), velocidad (8) y ángulo de giro (9) de la rueda. En esta animación una fuerza externa se aplica periódicamente a la rueda, la cual corresponde a un incremento de la velocidad que afecta la posición de la rueda, este incremento viene dado por:

$$v(t) = v(0) + 0.1 \tag{22}$$

3.2.2.5.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa el sistema plano y pivote de la Figura 3-12, como lo muestra la Figura 3-21.

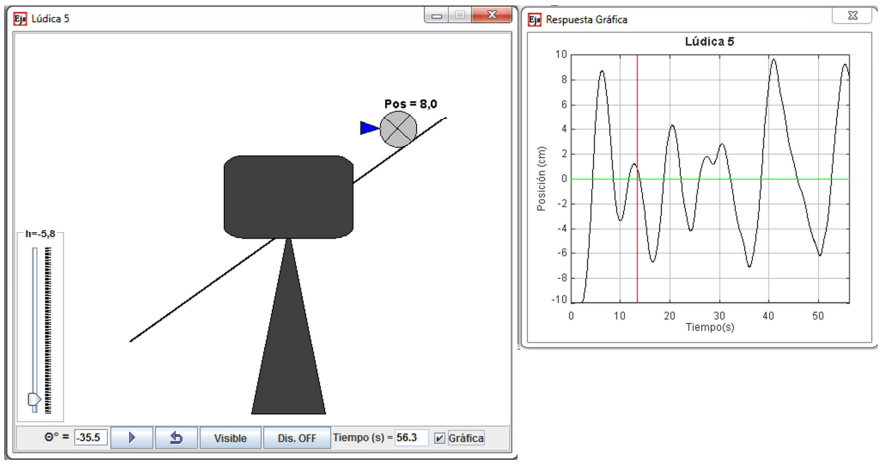


Figura 3-21 Vista de la lúdica 5: Plano, Rueda y Disturbio

Por medio del botón “ / ” se inicia o pone en pausa el desplazamiento de la rueda y con el botón “” se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se varía la altura “h” del extremo del plano, mientras que en el campo “θ” se muestra o modifica el ángulo actual del eje. Con el botón "Oculto" se oculta el centro del plano con un rectángulo gris y con el botón "Dis. ON" se aplica periódicamente una fuerza externa a la rueda, representada por la flecha “”.Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de la posición de la rueda a través del tiempo.

Las unidades y el valor de las variables más relevantes utilizadas en la animación son las mismas especificadas en la Tabla 3-3.

3.2.2.5.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe manipular la altura de inclinación del plano buscando mantener la posición de la rueda en el centro ante el efecto de un disturbio que la desplaza. Inicialmente por efecto de la realimentación visual podrá ubicar y mantener la rueda muy cerca del centro del plano. Posteriormente se perderá la realimentación, y por efecto del disturbio le será más difícil controlar el sistema. Finalmente analizará la evolución temporal de la posición de la rueda en la ventana emergente y concluirá que cuando se pierde la realimentación visual aumenta el error en la posición.

3.2.2.6 Lúdica 6: Válvulas y Tanque

Para introducir el concepto de *Sistemas de primer orden*, se requiere de una animación que le permita al estudiante:

- Observar la respuesta temporal de un sistema de primer orden

Para ello, se diseñó una animación que muestra el comportamiento del nivel en un tanque, donde de manera interactiva se manipula la apertura de la válvula registrando gráficamente la evolución del nivel a través del tiempo en una ventana emergente. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.6.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis realizado en la Lúdica 2: Válvula y Tanque. En este caso el flujo de descarga $q_{out}(t)$ es laminar y se representa de acuerdo a la ley de *Poiseuille*.

$$q_{out}(t) = \frac{h(t)}{R} \tag{23}$$

La relación ahora es lineal, siendo *R* la resistencia al paso de flujo en la descarga. Si se deriva con respecto al tiempo a ambos lados la ecuación definida en (14), se obtiene la ecuación que describe la dinámica del sistema:

$$A\frac{dh(t)}{dt} = K_V q_{in}(t) - \frac{h(t)}{R}\ , h(t_0) = h_0, \tag{24}$$

donde K_V es la ganancia de la válvula. En este caso la dinámica del sistema se representa por medio de una ecuación diferencial de primer orden lineal, a diferencia de la dinámica del sistema de la Lúdica 2: Válvula y Tanque, la cual corresponde a una ecuación diferencial no-lineal (ver ecuación (16)) por encontrarse la función *h(t)* elevada a una potencia distinta de uno o cero.

3.2.2.6.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema de la Figura 3-14, como lo muestra la Figura 3-22.

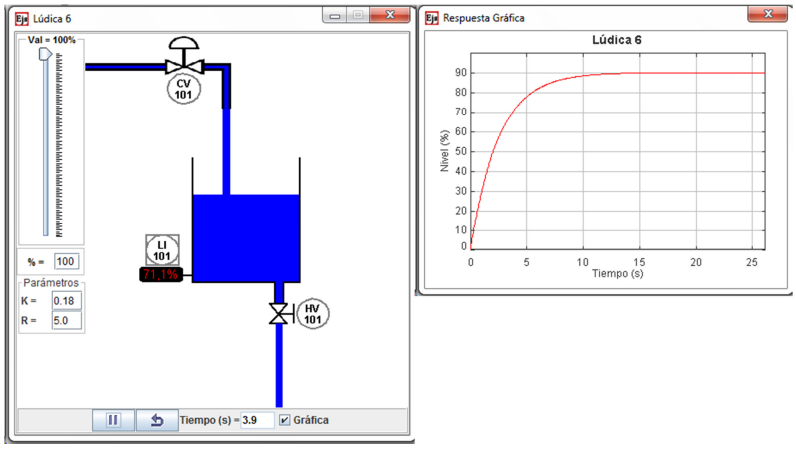


Figura 3-22 Vista de la lúdica 6: Válvulas y Tanque

Por medio del botón " / " se inicia o pone en pausa la ejecución de la animación y con el botón "" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se manipula el porcentaje de apertura de la válvula CV101, el cual también puede ser modificado por medio del campo "%". En el recuadro de "Parámetros" se modifican los valores de la ganancia "K" de la válvula CV101 y de la resistencia al paso del flujo de salida "R" de la válvula HV101. Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico del nivel "h" a través del tiempo.

En la Tabla 3-7 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-7 Unidades y valores de las variables en la Lúdica 6

Variable	Descripción	Unidades	Valor	
			Min.	Max.
h	Nivel del tanque	%	0	100
A	Área de sección transversal del tanque	m^2	0.5	
R	Resistencia de la descarga	s/m^2	5	
K_V	Ganancia de la válvula	-	0.18	

3.2.2.6.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe manipular la apertura de la válvula y observar la evolución temporal del nivel en la ventana emergente. Inicialmente obtendrá escalones de subida y de bajada de máxima amplitud del flujo de entrada (del 0 al 100% y del 100 al 0%) y al analizar los comportamientos obtenidos concluirá que el sistema presenta un comportamiento lineal, dado que la evolución del nivel en velocidad de respuesta y valor de ganancia estática es muy similar para cada caso. Finalmente conocerá la respuesta temporal típica de un sistema de primer orden y a partir de esta respuesta podrá obtener los parámetros de la dinámica del sistema.

3.2.2.7 Lúdica 7: Masa y Resorte

Para introducir el concepto de *Sistemas de segundo orden*, se requiere de una animación que le permita al estudiante:

- Observar la respuesta temporal de un sistema de segundo orden

Para ello, se diseñó una animación que muestra el desplazamiento de una masa unida a un resorte que oscila de manera amortiguada, donde de manera interactiva se adiciona masa al sistema, registrando gráficamente la evolución de la altura de la masa a través del tiempo en una ventana emergente. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.7.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis de un sistema de masa, resorte y amortiguador [45] [55], como lo muestra la Figura 3-23.

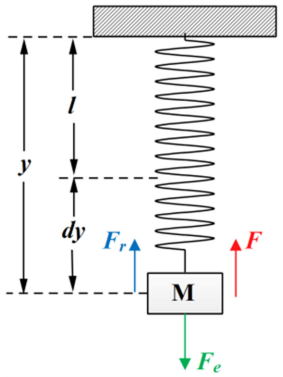


Figura 3-23 Diagrama de fuerzas del sistema Masa y resorte

En este sistema la masa *M* se encuentra unida a un resorte elástico de constante *k* sometida a una fuerza externa F_e que experimenta una fuerza de rozamiento F_r proporcional a la velocidad. Se supone que la respuesta del resorte a un desplazamiento *dy* obedece la Ley de Hook:

$$F = -kdy = -k(y - l), \tag{25}$$

donde, *l* corresponde a la longitud del resorte en equilibrio y *y* es la coordenada vertical del extremo del resorte. La ecuación que rige el movimiento del resorte se obtiene de la aplicación de la segunda ley de Newton:

$$ma = F - F_r + F_e \tag{26}$$

Esta ecuación se puede expresar en forma de ecuación diferencial, teniendo en cuenta que la aceleración y la velocidad corresponden a la segunda y primera derivada de la posición respectivamente. Representado la dinámica del sistema como una ecuación diferencial de segundo, se tiene:

$$\frac{d^2y}{dt^2} = \frac{1}{M}\left(-k(y(t) - l) - b\frac{dy}{dt} + F_e(t)\right), \tag{27}$$

donde *b* es el coeficiente de fricción viscoso. Esta ecuación se puede expresar como un sistema de dos ecuaciones diferenciales de primer orden, de la siguiente manera:

$$\begin{aligned} \frac{dy}{dt} &= v_y(t) \\ \frac{dv_y}{dt} &= \frac{1}{M}\left(-k(y(t) - l) - bv_y(t) + F_e(t)\right) \end{aligned} \tag{28}$$

3.2.2.7.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema de la Figura 3-23, como lo muestra la Figura 3-24.

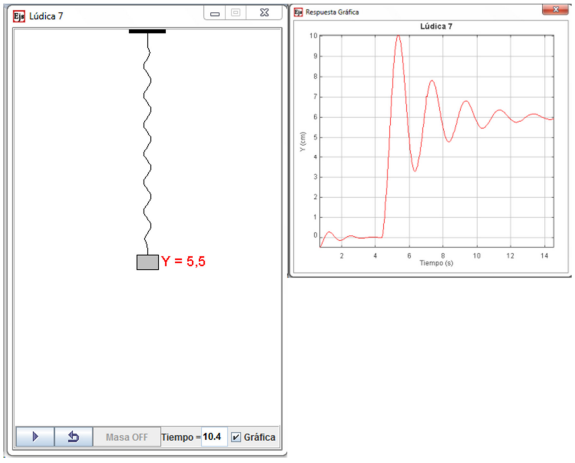


Figura 3-24 Vista de la lúdica 7: Masa y Resorte

Por medio del botón " / " se inicia o pone en pausa la ejecución de la animación y con el botón "" se reinicia de nuevo la animación. Con el botón "Masa On" se agrega una masa de 150 gr al sistema. Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de la altura "Y" a través del tiempo.

En la Tabla 3-8 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-8 Unidades y valores de las variables en la Lúdica 7

Variable	Descripción	Unidades	Valor	
			Min.	Max.
M	Masa del bloque	Kg	0.15	0.25
y	Desplazamiento de la masa	m	-1	10
l	Longitud del resorte en equilibrio	m	1	
k	Constante de elasticidad	Kgf/m	2.5	
b	Coeficiente de fricción viscoso	Ns/m	0.2	
F_e	Fuerza externa	Kgf	0.15	

3.2.2.7.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe agregar masa al sistema y observar la evolución temporal de la altura en la ventana emergente. Al analizar el comportamiento transitorio conocerá la respuesta temporal típica de un sistema de segundo orden sub-amortiguado y a partir de este comportamiento determinará características como sobre-paso, amortiguamiento, frecuencia natural y tiempo de establecimiento.

3.2.2.8 Lúdica 8: Plano y Ruedas

Para introducir el concepto de la *Acción de control todo – nada*, se requiere de una animación que le permita al estudiante:

- Observar los efectos de la acción de control Todo – Nada en el desempeño del sistema

Para ello, se diseñó una animación que muestra el comportamiento de una rueda rodando sobre un plano que se encuentra pivotado en el centro, donde de manera interactiva se manipula la altura del plano únicamente en dos posiciones, buscando posicionar la rueda en el centro. La animación cuenta con una ventana emergente donde se grafica la evolución de la posición de la rueda a través del tiempo. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.8.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis realizado en la Lúdica 1: Plano y Rueda, donde se determinaron las ecuaciones para la aceleración (6), posición (7), velocidad (8) y ángulo de giro (9) de la rueda.

3.2.2.8.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema plano y pivote de la Figura 3-12, como lo muestra la Figura 3-25.

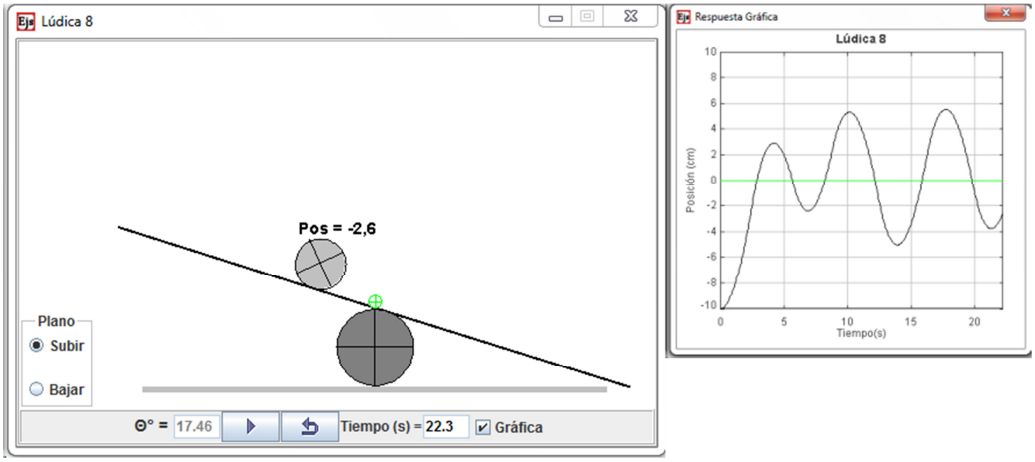


Figura 3-25 Vista de la Lúdica 8: Plano y Ruedas

Por medio del botón "▶/⏸" se inicia o pone en pausa el desplazamiento de la rueda y con el botón "↺" se reinicia de nuevo la animación. En el recuadro "Plano" se selecciona subir o bajar el extremo del plano, mientras que en el campo "θ" se muestra el ángulo actual del eje. Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de la posición de la rueda a través del tiempo.

Las unidades y el valor de las variables más relevantes utilizadas en la animación son similares a las especificadas en la Tabla 3-3, con la diferencia que ahora el ángulo de inclinación del plano toma únicamente dos valores 17.46° y -17.46°.

3.2.2.8.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe manipular la altura de inclinación del plano únicamente en dos posiciones e intentar ubicar la rueda en el centro. Como el ángulo de inclinación del plano solo toma dos valores, le exigirá al usuario conmutar rápidamente

entre ambas para poder mantener la rueda cerca al centro. Al analizar la evolución temporal de la posición de la rueda en la ventana emergente, observará un comportamiento oscilatorio cuasi-estacionario, típico de esta acción de control.

3.2.2.9 Lúdica 9: Plano Inclinado Automático

Para introducir el concepto de la *Acción de control PID*, se requiere de una animación que le permita al estudiante:

- Observar los efectos de las acciones de control: Proporcional (P), Integral (I), Proporcional-Integral (PI), Proporcional-Derivativa (PD) y Proporcional-Integral-Derivativa (PID), en el desempeño del sistema

Para ello, se diseñó una animación que muestra el comportamiento de una rueda rodando sobre un plano inclinado, el cual se encuentra apoyado sobre un plano horizontal en un extremo, donde de manera interactiva se ajusta la altura del extremo libre de acuerdo a la acción de control seleccionada, registrando en una tabla de datos valores de altura y posición. La animación cuenta con una ventana emergente que registra gráficamente la distancia final alcanzada por la rueda para cada lanzamiento. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.9.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis realizado en la Lúdica 1: Plano y Rueda, donde se determinaron las ecuaciones para la aceleración (6), posición (7), velocidad (8) y ángulo de giro (9) de la rueda.

En esta animación se implementa un sistema de control de posición como el descrito en la Figura 3-17 de la Lúdica 3: Plano Inclinado Manual. En este caso se utilizan las acciones de control [54]: Proporcional (P), Integral (I), Proporcional-Integral (PI), Proporcional-Derivativa (PD) y Proporcional-Integral-Derivativa (PID) que permiten ajustar la altura del plano inclinado en función del valor del error para cada lanzamiento de la rueda, de la siguiente manera:

- La acción Proporcional P, genera la señal de control continua:

$$u(t) = bias + k_p e(t), \tag{29}$$

  donde k_p es la ganancia proporcional, y *bias* es la compensación en la señal de control para eliminar el error permanente. Expresando de manera discreta como ecuación de diferencias se obtiene:

$$h_C(k+1) = bias + k_p e(k), \tag{30}$$

  donde $h_c(k+1)$ es la altura calculada y $e(k)$ es el error (diferencia entre la distancia deseada y la alcanzada por la rueda).

- La acción Proporcional-Derivativa PD, genera la señal de control continua:

$$u(t) = bias + k_p e(t) + k_p k_d \frac{d}{dt} e(t), \tag{31}$$

  donde k_p es la ganancia proporcional, y k_d es la ganancia derivativa. Utilizando la aproximación de Euler de la derivada para discretizar, se obtiene:

$$h_C(k+1) = bias + k_p e(k) + k_p k_d [e(k) - e(k-1)] \tag{32}$$

- La acción Integral I, genera la señal de control continua:

$$u(t) = k_i \int_0^t e(t)dt, \tag{33}$$

donde k_i es la ganancia integral. Utilizando la aproximación trapezoidal de la integral, para discretizar se obtiene:

$$h_C(k+1) = h(k) + k_i \left[\frac{e(k) + e(k-1)}{2}\right] \tag{34}$$

- La acción Proporcional-Integral PI, genera la señal de control continua:

$$u(t) = k_p e(t) + k_p k_i \int_0^t e(t)dt, \tag{35}$$

donde k_p es la ganancia proporcional y k_i es la ganancia integral. Utilizando la aproximación de Euler de la integral para discretizar, se obtiene:

$$h_C(k+1) = h(k) + [k_p + k_p k_i] e(k) - k_p e(k-1) \tag{36}$$

- La acción Proporcional-Integral-Derivativa PID, genera la señal de control continua:

$$u(t) = k_p e(t) + k_p k_i \int_0^t e(t)dt + k_p k_d \frac{d}{dt} e(t), \tag{37}$$

donde k_p es la ganancia proporcional, k_i es la ganancia integral y k_d es la ganancia derivativa. Utilizando la aproximación de Euler de la integral y de la derivada para discretizar, se obtiene:

$$h_C(k+1) = h(k) + [k_p + k_p k_i] e(k) + k_p k_d [e(k) - e(k-1)] \tag{38}$$

3.2.2.9.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el plano inclinado de la Figura 3-16, como lo muestra la Figura 3-26.

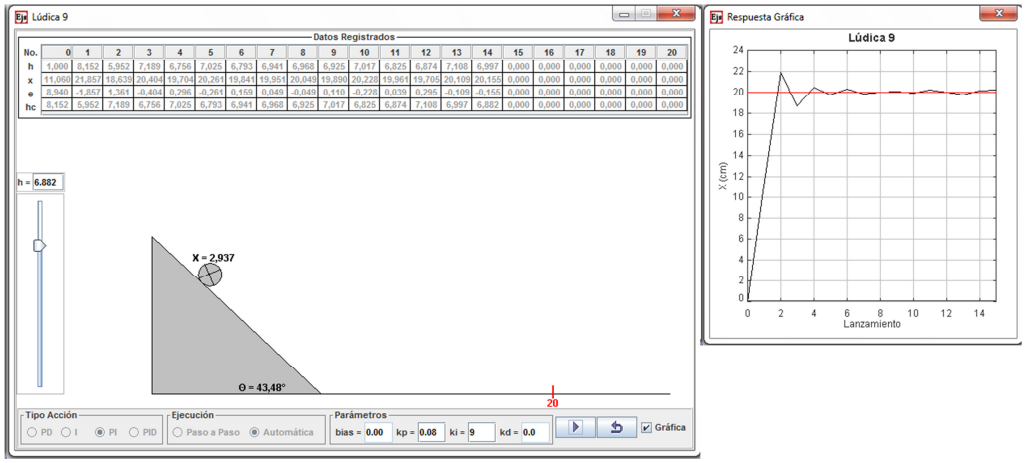


Figura 3-26 Vista de la lúdica 9: Plano Inclinado Automático

Por medio del botón " / " se inicia o pone en pausa el desplazamiento de la rueda y con el botón "" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se varía la altura "h" del extremo del plano inclinado, la cual también puede ser modificada por medio del campo "h". En la tabla "Datos Registrados", se registra el valor de las variables: Altura actual h, Distancia alcanzada x, Error e (distancia Deseada - Alcanzada) y Altura calculada hc. En el recuadro "Tipo Acción" se selecciona la acción de control: PD, I, PI o PID y por medio de campos numéricos se ajusta el valor de los parámetros *kp, ki, kd y bias*. En el recuadro "Ejecución" se selecciona la manera de ajustar la altura del plano inclinado, de manera manual (paso a paso) o Automática. Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de la posición alcanzada por la rueda para cada lanzamiento.

Las unidades y el valor de las variables más relevantes utilizadas en la animación son las mismas especificadas en la Tabla 3-5.

3.2.2.9.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe ajustar la altura del extremo libre (según la acción de control seleccionada: P, PD, I, PI o PID), buscando que la rueda rodando por el plano inclinado, alcance una distancia deseada en el plano horizontal, analizando el desempeño del error de posición alcanzado para cada acción de control. Para la acción P concluirá que el uso de alta ganancia para disminuir el error, se limita por la pérdida de amortiguamiento produciendo inestabilidad. Para la acción PD concluirá que estabiliza y mejora el amortiguamiento y el error del sistema, en comparación con la acción P. Para la acción I concluirá que corrige el error, pero en comparación con la acción P el sistema se vuelve más lento. Para la acción PI concluirá que mejora el amortiguamiento y la velocidad de respuesta en comparación con la acción I. Finalmente para la acción PID concluirá que el sistema logra una alta velocidad de respuesta con un adecuado amortiguamiento y con error nulo.

3.2.2.10 Lúdica 10: Plano, Rueda y Pivote

Para introducir el concepto de *Estabilidad y lugar geométrico de las raíces*, se requiere de una animación que le permita al estudiante:

- Observar el comportamiento de un sistema estable e inestable
- Observar el efecto de la ganancia en la estabilidad del sistema

Para ello, se diseñó una animación que muestra el comportamiento de una rueda rodando sobre un plano, cuyo pivote se ajusta en tres diferentes posiciones: izquierda, centro y derecha. De manera interactiva se selecciona la posición del pivote y se manipula la inclinación del plano, buscando ubicar la rueda en el centro. La evolución de la posición de la rueda a través del tiempo para cada posición del pivote, se grafica en una ventana emergente. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.10.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis realizado en la Lúdica 1: Plano y Rueda, donde se determinaron las ecuaciones para la aceleración (6), posición (7), velocidad (8) y ángulo de giro (9) de la rueda.

3.2.2.10.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa el sistema plano y pivote de la Figura 3-12, como lo muestra la Figura 3-27.

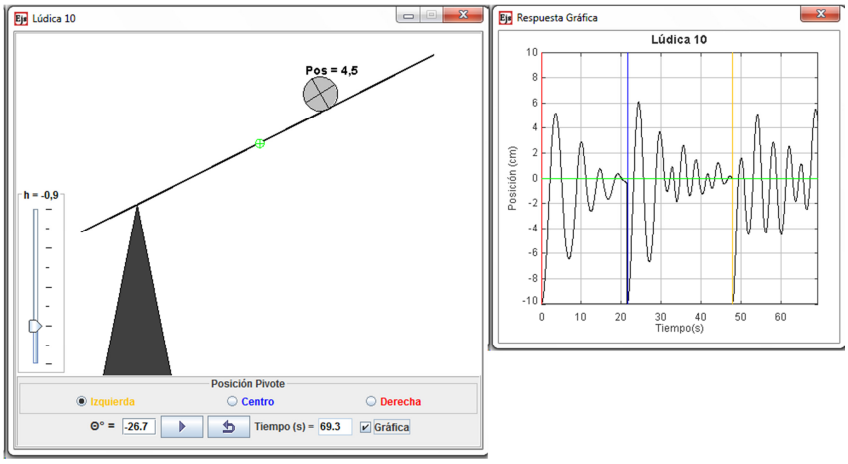


Figura 3-27 Vista de la lúdica 10: Plano, Rueda y Pivote

Por medio del botón "▶/⏸" se inicia o pone en pausa el desplazamiento de la rueda y con el botón "↺" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se varía la altura "h" del extremo del plano, mientras que en el campo "θ" se muestra o modifica el ángulo actual del eje. En el recuadro "Posición Pivote", se selecciona la ubicación del pivote (Izquierda, Centro o Derecha). Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de la posición de la rueda a través del tiempo, donde se traza una línea vertical de color (naranja, azul o rojo) que indica la posición del pivote actual selecciona.

Las unidades y el valor de las variables más relevantes utilizadas en la animación son las mismas especificadas en la Tabla 3-3.

3.2.2.10.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe seleccionar una ubicación del pivote y manipular la altura de inclinación del plano buscando posicionar la rueda en el centro, analizando la evolución temporal de la posición de la rueda para cada posición del pivote en la ventana emergente. Observará que la respuesta del sistema cambia considerablemente según la posición del pivote. Con el pivote ubicado en la posición izquierda, mayor es la amplificación del ángulo del plano para una altura determinada, siendo más difícil controlar el sistema, obteniéndose una respuesta más oscilatoria.

3.2.2.11 Lúdica 11: Masa, Resorte y Circunferencia

Para introducir el concepto de *Respuesta en frecuencia*, se requiere de una animación que le permita al estudiante:

- Obtener la respuesta en frecuencia de un sistema de segundo orden

Para ello, se diseñó una animación que muestra el desplazamiento de una masa unida a un resorte que oscila de manera amortiguada, donde de manera interactiva se excita la

parte superior del sistema con una señal sinusoidal externa de frecuencia variable. La animación cuenta con una ventana emergente donde se registra gráficamente la evolución de la altura de la masa a través del tiempo. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.11.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis realizado en la Lúdica 7: Masa y Resorte, donde se determinó la ecuación diferencial (27) que describe el comportamiento del sistema. En este caso el análisis se complementa con el sistema de la Figura 3-28.

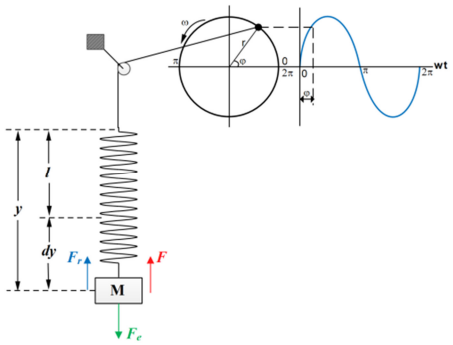


Figura 3-28 Sistema masa y resorte, excitado con una señal sinusoide

De esta figura la longitud *l(t)* del resorte se excita con una señal sinusoide de frecuencia variable "ω" (rad/s), igual a:

$$x(t) = Xsen(\omega t) \tag{39}$$

Por tanto el modelo matemático que representa el comportamiento de esta animación, viene dado por:

$$\frac{d^2y}{dt^2} = \frac{1}{M}\left(-k\big(y(t) - l(t)\big) - b\frac{dy}{dt} + F_e(t)\right) \tag{40}$$

3.2.2.11.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema de la Figura 3-28, como lo muestra la Figura 3-29.

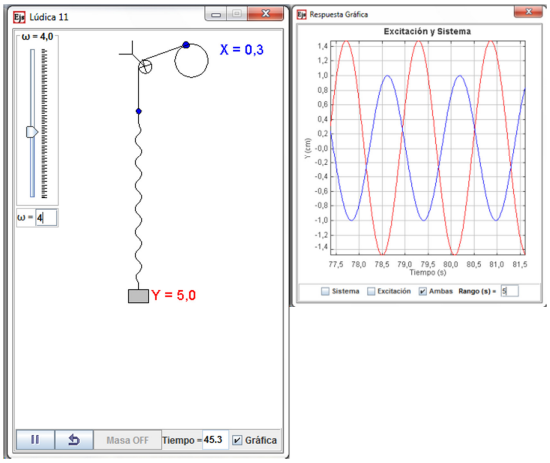


Figura 3-29 Vista de la lúdica 11: Masa, Resorte y Circunferencia

Por medio del botón " / " se inicia o pone en pausa la ejecución de la animación y con el botón "" se reinicia de nuevo la animación. Con el botón "Masa On" se agrega una masa de 150 gr al sistema. Con la barra de deslizamiento vertical se varía la frecuencia "ω" de excitación al sistema, la cual también se puede modificar numéricamente por medio del campo "ω". Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de la altura "Y" de la masa y de la señal de excitación "X" a través del tiempo, además esta gráfica cuenta opciones de selección y campos numéricos que ajustan la visualización de las señales.

Las unidades y el valor de algunas de las variables más relevantes utilizadas en la animación son las mismas especificadas en la Tabla 3-8. Para complementar estas variables en la Tabla 3-9 se especifican las variables de la señal de excitación externa dada en la ecuación (39).

Tabla 3-9 Unidades y valores de la señal de excitación externa en la Lúdica 11

Variable	Descripción	Unidades	Valor	
			Min.	Max.
ω	Frecuencia de excitación	rad/seg	0	9
X	Amplitud señal de excitación	m	-1	1

3.2.2.11.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe ir aumentando la frecuencia de excitación *ω* y analizar el comportamiento oscilatorio de la altura de la masa en la ventana emergente. Realizará un barrido de frecuencia y para cada variación observará y tomará nota del desplazamiento pico de la altura de la masa y el atraso temporal respecto a la señal de excitación *x(t)*. Con estos datos construirá la respuesta en frecuencia del sistema, lo cual le permitirá obtener y analizar características como: frecuencia de resonancia, ancho de banda e índices de estabilidad relativa como los márgenes de ganancia y de fase.

3.2.2.12 Lúdica 12: Cuerdas, Poleas y Masa

Para introducir el concepto de *Estructura de control directo de disturbio*, se requiere de una animación que le permita al estudiante:

- Observar el efecto del disturbio en el desempeño de un sistema con y sin la estrategia de control directo de disturbio

Para ello, se diseñó una animación que muestra el comportamiento de un sistema de tres poleas, donde la cuerda que atraviesa el sistema se encuentra fija en un extremo y el otro extremo se conecta a una circunferencia que describe un movimiento periódico. En el centro del sistema se acopla una masa que sube y baja por causa de este movimiento periódico. De manera interactiva se debe manipular la circunferencia buscando mantener la masa en una posición de referencia deseada. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.12.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis del sistema de la Figura 3-30.

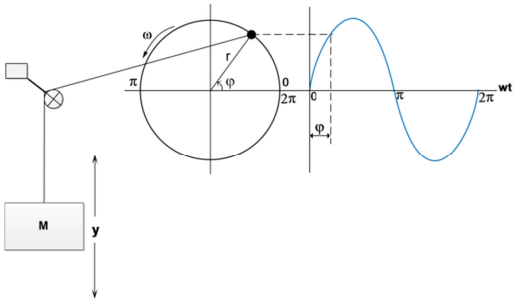


Figura 3-30 Sistema Masa - circunferencia

En este sistema la masa *M* se desplaza periódicamente en el eje vertical, debido a una señal de excitación sinusoidal de frecuencia "ω" (rad/s), igual a:

$$y(t) = sen(\omega t), \qquad con\ \omega = 2\pi f \tag{41}$$

La animación representa un control directo de disturbio que permite cancelar el efecto de la excitación sinusoidal, como lo muestra el esquema de la Figura 3-31.

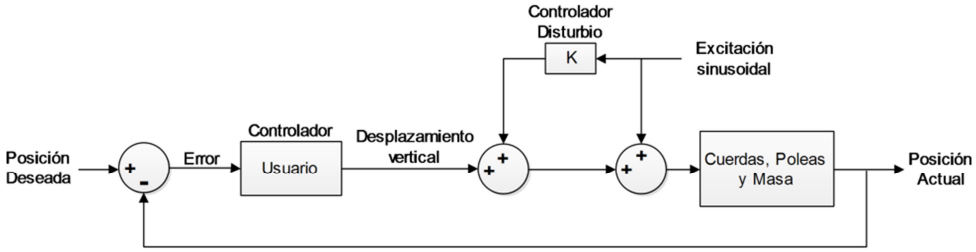


Figura 3-31 Control de posición con cancelación de disturbio

Cuando la ganancia *K* del controlador de disturbio es igual a -1, se obtiene un movimiento periódico opuesto que cancela el efecto del disturbio.

3.2.2.12.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema de la Figura 3-30, como lo muestra la Figura 3-32.

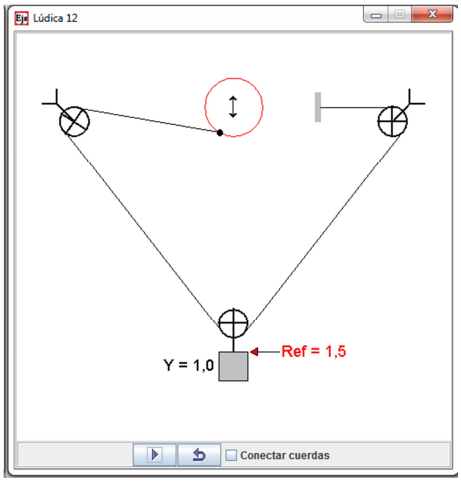


Figura 3-32 Vista de la lúdica 12: Cuerdas, Poleas y Masa

Por medio del botón " / " se inicia o pone en pausa la ejecución de la animación y con el botón "" se reinicia de nuevo la animación. Seleccionando "Conectar cuerdas" se conecta el extremo de la cuerda fija a la cuerda que describe la circunferencia.

En la Tabla 3-10 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-10 Unidades y valores de las variables en la Lúdica 12

Variable	Descripción	Unidades	Valor	
			Min.	Max.
y	Desplazamiento de la masa	m	-1	5
f	Frecuencia de la señal de excitación	Hz	0.167	

3.2.2.12.3 Resultados de aprendizaje esperados

En esta lúdica animada, inicialmente el estudiante debe manipular la circunferencia que conecta la cuerda al sistema de poleas, buscando ubicar la masa en una posición de referencia dada, posteriormente con los extremos de la cuerda conectados nuevamente debe intentar ubicar la masa. Se trata por lo tanto de un sistema de control de posición; donde la planta a controlar consiste en la cuerda, polea y la masa, la salida controlada es la posición de la masa, la variable manipulada es el desplazamiento vertical de la cuerda y el disturbio es el movimiento periódico que desplaza la posición de la masa. El elemento de realimentación son los ojos del usuario y el controlador él mismo (cerebro); el actuador es la mano con el Mouse. El estudiante se dará cuenta que por el movimiento periódico del sistema, es difícil ubicar y mantener la masa en la posición de referencia, pero al conectar los extremos de las cuerdas observará que el movimiento periódico no altera la posición de la masa, cancelando su efecto y logrando ubicar fácilmente la masa en la referencia dada.

3.2.2.13 Lúdica 13: Tanques en Serie

Para introducir el concepto de la *Estructura de control en cascada*, se requiere de una animación que le permita al estudiante:

- Observar el efecto del disturbio en el desempeño de un sistema con y sin la estrategia de control en cascada

Para ello, se diseñó una animación que muestra el comportamiento del nivel en dos tanques ubicados en serie (uno encima del otro), donde de manera interactiva se manipula el porcentaje de apertura de la válvula del primer tanque, buscando mantener el nivel del segundo tanque en una referencia deseada, ante el efecto de un flujo externo que altera el nivel de los tanques. La animación cuenta con una ventana emergente, donde se registra gráficamente la evolución de los niveles a través del tiempo. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.13.1 Modelo

La definición del modelo matemático de la animación se basa en analizar el comportamiento del nivel en dos tanques ubicados en serie, como lo muestra la Figura 3-33.

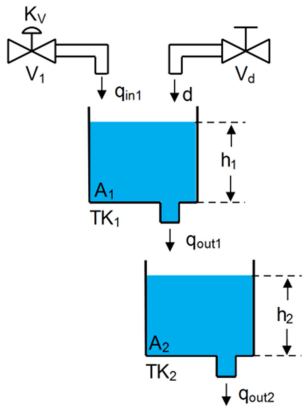


Figura 3-33 Sistema del nivel en dos tanques en serie

En este sistema por medio de la válvula V_1 se manipula el flujo de entrada $q_{in1}(t)$ al tanque 1 TK_1, donde a través de su flujo de descarga $q_{out1}(t)$ se llena el tanque 2 TK_2 el cual presenta un flujo de descarga $q_{out2}(t)$. La válvula V_d genera un flujo externo d que incrementa el nivel $h_1(t)$ del TK_1 aumentando $q_{out1}(t)$ y afectando el nivel $h_2(t)$ del TK_2. Las áreas transversales de los tanques A_1 y A_2 se consideran constantes, K_V es la ganancia de V_1 y los flujos de descarga son laminares (ver ecuación (23)), siendo R_1 y R_2 las resistencias al paso del flujo en las descargas $q_{out1}(t)$ y $q_{out2}(t)$ respectivamente.

El modelo matemático se obtiene de realizar un balance de materia a cada tanque como se planteó en la Lúdica 2: Válvula y Tanque. Derivando se obtienen las ecuaciones que describen la dinámica del sistema:

$$A_1 \frac{dh_1(t)}{dt} = K_V q_{in1}(t) - \frac{h_1(t)}{R_1} + d(t) \tag{42}$$

$$A_2 \frac{dh_2(t)}{dt} = \frac{h_1(t)}{R_1} - \frac{h_2(t)}{R_2} \tag{43}$$

La animación implementa un control en cascada que permite mantener el nivel $h_2(t)$ del TK_2 en una referencia establecida, como lo muestra el esquema de la Figura 3-34.

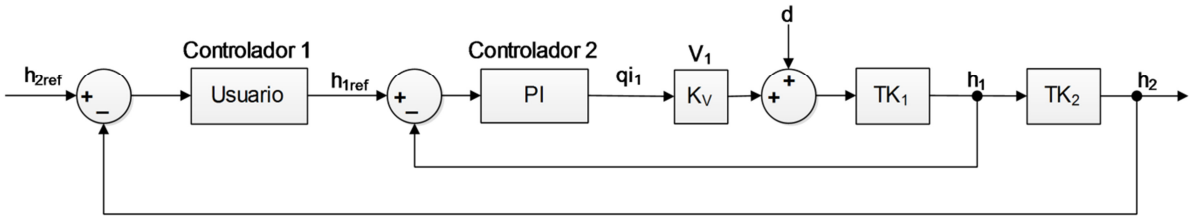


Figura 3-34 Estructura de control en cascada para el sistema del nivel en dos tanques

En la estructura del control en cascada, el lazo interno controla el nivel $h_1(t)$ del TK_1 por medio de un controlador PI, el cual cuenta con un algoritmo de ajuste de la referencia del tanque 1 h_{1ref} para una transferencia sin saltos (*Bumpless*) y sin saturación del actuador debida a la acción integral (método *Anti-windup*) [54]. En la Figura 3-35 se muestra el esquema del controlador PI implementado.

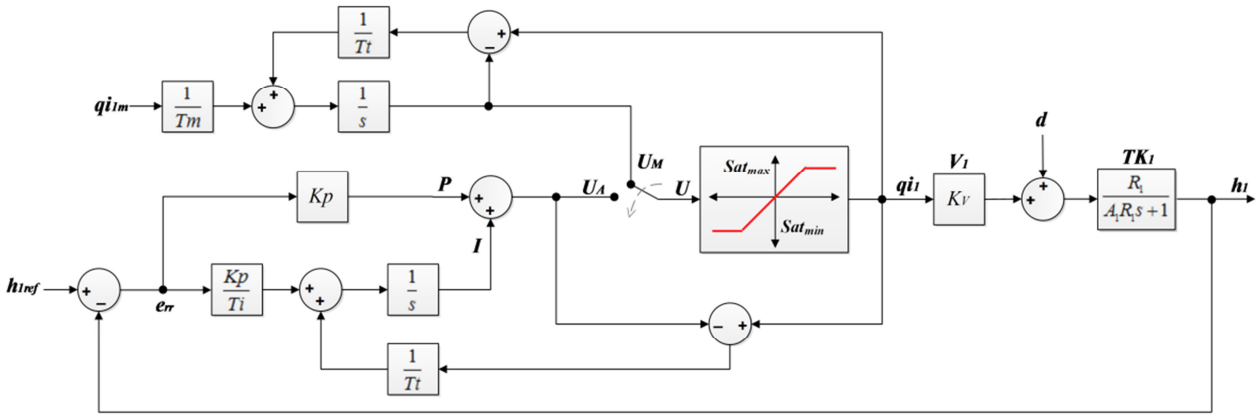


Figura 3-35 Esquema de control PI con Bumpless y Anti-windup para el TK_1

Donde T_t es la constante de tiempo que determina con cuánta rapidez se resetea la integral y T_m es la constante de tiempo que define la rapidez de cambio de la referencia manual $qi_{1M.}$ Del esquema anterior la señal de control $U(s)$ toma dos estados: manual $U_M(s)$ y automático $U_A(s)$ iguales a:

$$U_M(s) = \left[\frac{qi_{1m}(s)}{T_m} + \frac{qi_1(s) - U_M(s)}{T_t}\right]\frac{1}{s} \tag{44}$$

$$U_A(s) = P(s) + I(s) = k_p E(s) + \left(\frac{k_p E(s)}{T_i} + \frac{\left(qi_1(s) - U_A(s)\right)}{T_t}\right)\frac{1}{s} \tag{45}$$

Para implementar esta acción de control en EJS se debe usar su equivalente discreto, el cual se obtiene de discretizar con la aproximación de Euler de la derivada [53]:

$$\frac{du(t)}{dt} = \frac{\left(U(k) - U(k-1)\right)}{T_s} \tag{46}$$

Al reemplazar (46) en (44) y (45), se obtiene:

$$u_m(k) = u_m(k-1) + \frac{qi_{1m}(k)T_s}{T_m} + \frac{\left(qi_1(k) - u_m(k)\right)T_s}{T_t} \tag{47}$$

$$P(k) = k_p e(k)$$

$$I(k) = I(k-1) + \frac{k_p e(k)T_s}{T_i} + \frac{\left(qi_1(k) - u_a(k)\right)T_s}{T_t} \tag{48}$$

$$u_a(k) = P(k) + I(k)$$

3.2.2.13.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema de los dos tanques de la Figura 3-33, como lo muestra la Figura 3-36.

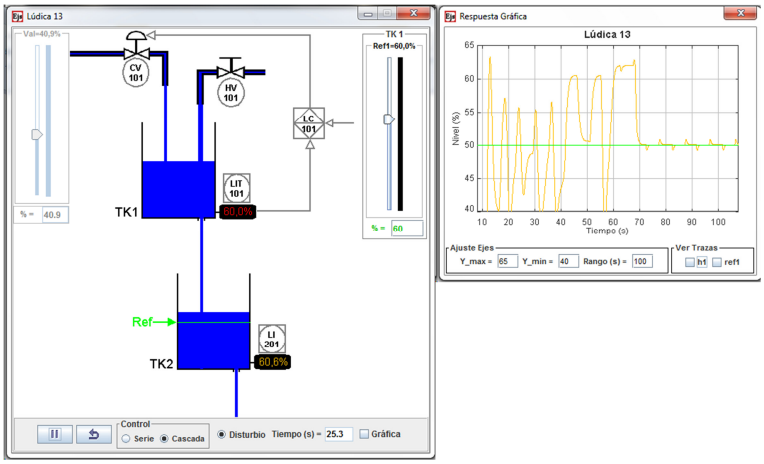


Figura 3-36 Vista de la lúdica 13: Tanques en Serie

Por medio del botón " ▶ / ⏸ " se inicia o pone en pausa la ejecución de la animación y con el botón "↩" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical izquierda se manipula el porcentaje de apertura de la válvula CV101, el cual también puede ser modificado por medio del campo "%", mientras que con la barra de deslizamiento vertical derecha se establece el porcentaje del nivel de referencia para el tanque 1 "Ref1", el cual también puede ser modificado por medio del campo "%".En el recuadro "Control" se selecciona el tipo de estructura de control del sistema, en "Serie" o "Cascada". Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de los niveles a través del tiempo, además esta gráfica cuenta opciones de selección y campos numéricos que ajustan la visualización de las señales.

En la Tabla 3-11 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-11 Unidades y valores de las variables en la Lúdica 13

Variable	Descripción	Unidades	Valor	
			Min.	Max.
h_1, h_2	Nivel de los tanques	%	0	100
A_1, A_2	Áreas de sección transversal de los tanques	m^2	0.1	
K_V	Ganancia de la válvula	-	0.24	
R_1, R_2	Resistencias de la descarga	s/m^2	5	

3.2.2.13.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe manipular el flujo de entrada al TK_1, buscando mantener el nivel del TK_2 en el 50% ante el efecto de un flujo periódico externo (disturbio) que modifica el nivel de los tanques. El estudiante se dará cuenta que por el efecto del disturbio, es difícil mantener el nivel del TK_2 en el 50%, ya que se obtienen grandes variaciones; pero al implementar el control en cascada observará que el efecto del disturbio se disminuye considerablemente, logrando ajustar y mantener el nivel del TK_2 en el 50% con mínimas variaciones. Concluirá que con el control en cascada se explota el beneficio de la realimentación al subdividir una planta compleja y resolver el problema de control a pasos, mediante controladores simples.

3.2.2.14 Lúdica 14: Tanques en Paralelo 1

Para introducir el concepto de *Controlabilidad*, se requiere de una animación que le permita al estudiante:

- Determinar cuándo un sistema es controlable y no controlable

Para ello, se diseñó una animación que muestra el comportamiento del nivel en dos tanques ubicados en paralelo (uno en frente del otro) que se encuentran conectados entre sí, donde de manera interactiva se manipula la apertura de la válvula buscando obtener niveles con alturas diferentes en cada tanque. La animación cuenta con una ventana emergente donde se registra gráficamente la evolución de los niveles a través del tiempo. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.14.1 Modelo

La definición del modelo matemático de la animación se basa en analizar el comportamiento del nivel en dos tanques ubicados en paralelo, como lo muestra la Figura 3-37.

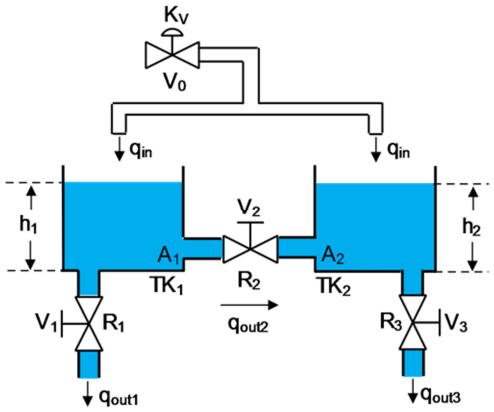


Figura 3-37 Sistema del nivel en dos tanques en paralelo

En este sistema por medio de la válvula V_0 se manipula el flujo de entrada $q_{in}(t)$ el cual se considera igual para cada tanque. Las áreas transversales de los tanques A_1 y A_2 se consideran constantes, K_V es la ganancia de V_0 y los flujos de descarga $q_{out1}(t)$, $q_{out2}(t)$ y $q_{out3}(t)$ son laminares (ver ecuación (22)), siendo R_1, R_2 y R_3 las resistencias al paso del flujo de salida de las válvulas V_1, V_2 y V_3 respectivamente.

El modelo matemático se obtiene de realizar un balance de materia a cada tanque como se planteó en la Lúdica 2: Válvula y Tanque. Derivando se obtienen las ecuaciones que describen la dinámica del sistema:

$$A_1 \frac{dh_1(t)}{dt} = K_V q_{in}(t) - \frac{h_1(t)}{R_1} - \left(\frac{h_1(t) - h_2(t)}{R_2}\right) \tag{49}$$

$$A_2 \frac{dh_2(t)}{dt} = K_V q_{in}(t) - \frac{h_2(t)}{R_3} + \left(\frac{h_1(t) - h_2(t)}{R_2}\right) \tag{50}$$

3.2.2.14.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema de los tanques en paralelo de la Figura 3-37, como lo muestra la Figura 3-38.

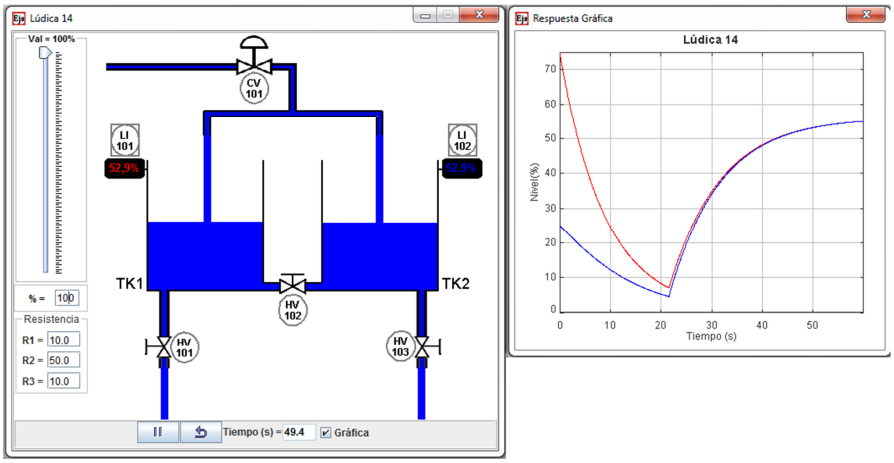


Figura 3-38 Vista de la lúdica 14: Tanques en Paralelo 1

Por medio del botón " / " se inicia o pone en pausa la ejecución de la animación y con el botón "" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se manipula el porcentaje de apertura de la válvula CV101, el cual también puede ser modificado por medio del campo "%". Por medio de campos numéricos se modifican los valores de las resistencias al paso del flujo de salida "R1", "R2" y "R3". Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de los niveles a través del tiempo.

En la Tabla 3-12 se especifican las unidades y el valor de las variables más relevantes utilizadas en la animación.

Tabla 3-12 Unidades y valores de las variables en la Lúdica 14

Variable	Descripción	Unidades	Valor	
			Min.	Max.
h_1, h_2	Nivel de los tanques	%	0	100
A_1, A_2	Áreas de sección transversal de los tanques	m^2	1	
K_V	Ganancia de la válvula	-	0.056	
R_1, R_3	Resistencias de la descarga	s/m^2	10	
R_2	Resistencia de descarga entre los tanques	s/m^2	50	

3.2.2.14.3 Resultados de aprendizaje esperados

En esta lúdica animada el estudiante debe manipular el flujo de entrada a los tanques buscando obtener niveles con alturas diferentes. Inicialmente analizará el comportamiento del sistema con valores iguales de las resistencias R_1 y R_3, observando que no es posible conseguir niveles con alturas diferentes en los tanques, dado que el sistema no es controlable; la pérdida de controlabilidad se debe a que con iguales resistencias de descarga los modelos son iguales y el flujo entre los dos tanques iguala los dos niveles, no permitiendo valores distintos. Posteriormente observará que si los valores de R_1 y R_3 son diferentes, podrá obtener niveles con alturas distintas, y concluirá que bajo esta condición es posible controlar los niveles h_1 y h_2.

3.2.2.15 Lúdica 15: Tanques en Paralelo 2

Para introducir el concepto de *Observabilidad*, se requiere de una animación que le permita al estudiante:

- Determinar cuándo un sistema es observable y no observable

Para ello, se diseñó una animación que muestra el comportamiento del nivel en dos tanques ubicados en paralelo (uno en frente del otro) que se encuentran conectados entre sí. Implementa un observador que estima el valor del nivel del primer tanque a partir de la medida del flujo entre los tanques, en donde de manera interactiva se varían parámetros del sistema observando el comportamiento de este nivel estimado. La animación cuenta con una ventana emergente donde se registra gráficamente la evolución de los niveles a través del tiempo. A continuación se define el modelo y la vista para la implementación de esta lúdica animada en EJS.

3.2.2.15.1 Modelo

La definición del modelo matemático de la animación se basa en el análisis realizado en la Lúdica 14: Tanques en Paralelo 1, donde se determinaron las ecuaciones (49) y (50) que describen la dinámica del sistema. En este caso, a partir de la medida del flujo a través de la válvula V_2 (ver Figura 3-37), se ha diseñado un observador para estimar el valor del nivel h_1, denominado h_{1e}, de la siguiente manera.

Inicialmente a partir de las ecuaciones dinámicas definidas en (49) y (50) se representa el sistema en el espacio de estados:

$$\begin{bmatrix} \dot{h_1} \\ \dot{h_2} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1} - \frac{1}{R_2} & \frac{1}{R_2} \\ \frac{1}{R_2} & -\frac{1}{R_3} - \frac{1}{R_2} \end{bmatrix} * \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} * K_V q_{in} \tag{51}$$

Se asume que solo es posible medir el flujo a través de la válvula V_2: $\frac{h_1(t)-h_2(t)}{R_2}$, siendo la salida del sistema igual a:

$$y = \begin{bmatrix} \frac{1}{R_2} & -\frac{1}{R_2} \end{bmatrix} * \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \tag{52}$$

EL vector de ganancias del observador viene dado por:

$$J = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix} \tag{53}$$

Posteriormente se obtiene el polinomio característico del observador:

$$\begin{gathered} |sI - (A - JC)| = \\ s^2 + (J_1 a_{12} - J_2 a_{12} - a_{11} - a_{22})s + (a_{11}a_{22} - a_{12}a_{21} + J_2 {a_{12}}^2 - J_1 a_{12} a_{21} + J_2 a_{11} a_{12} - J_1 a_{12} a_{22}) \end{gathered} \tag{54}$$

Donde a_{ij} corresponde al valor de los coeficientes de la matriz A del sistema. Por tanto se pueden escoger los polos del observador para determinar los valores J_1 y J_2.

Asumiendo valores por defecto de las resistencias $R_1 = 10$, $R_2 = 50$, y $R_3 = 10$, se calculó la matriz A y se determinaron los polos del sistema con el comando *eig(A)* de Matlab.

$$a_{11} = -\frac{1}{R_1} - \frac{1}{R_2} = -0.12 \qquad a_{12} = a_{21} = \frac{1}{R_2} = 0.02 \qquad a_{22} = -\frac{1}{R_3} - \frac{1}{R_2} = -0.12$$

$$eig(A) = (s + 0.14)(s + 0.1)$$

La dinámica del observador se determinó para un tiempo de establecimiento de 4 segundos, mucho menor al tiempo de establecimiento del sistema. Se seleccionaron por tanto dos polos del observador en *S = -1*; igualando los polos deseados del observador en la ecuación (54), se obtiene:

$$s^2 + (J_1 a_{12} - J_2 a_{12} - a_{11} - a_{22})s + (a_{11}a_{22} - a_{12}a_{21} + J_2 {a_{12}}^2 - J_1 a_{12}a_{21} + J_2 a_{11}a_{12} - J_1 a_{12}a_{22}) = s^2 + 2s + 1$$

Resolviendo para J_1 y J_2:

$$J_1 = \frac{2a_{11} + 2a_{12} + a_{11}a_{12} + a_{12}a_{21} + a_{12}a_{22} + {a_{11}}^2 + 1}{a_{11}a_{12} - a_{12}a_{21} - a_{12}a_{22} + {a_{12}}^2} \tag{55}$$

$$J_2 = \frac{2a_{21} + 2a_{22} + a_{11}a_{21} + a_{12}a_{21} + a_{21}a_{22} + {a_{22}}^2 + 1}{a_{11}a_{12} - a_{12}a_{21} - a_{12}a_{22} + {a_{12}}^2} \tag{56}$$

La ecuación del observador viene dada por:

$$\dot{\hat{x}} = A\hat{x} + Bu + J(y - \hat{y}) \tag{57}$$

Obteniendo finalmente:

$$\frac{dh_{1e}(t)}{dt} = K_V q_{in}(t) - \frac{h_{1e}(t)}{R_1} - \left(\frac{h_{1e}(t) - h_{2e}(t)}{R_2}\right) + J_1\left(\left(\frac{h_1(t) - h_2(t)}{R_2}\right) - \left(\frac{h_{1e}(t) - h_{2e}(t)}{R_2}\right)\right) \tag{58}$$

$$\frac{dh_{2e}(t)}{dt} = K_V q_{in}(t) - \frac{h_{2e}(t)}{R_3} + \left(\frac{h_{1e}(t) - h_{2e}(t)}{R_2}\right) + J_2\left(\left(\frac{h_1(t) - h_2(t)}{R_2}\right) - \left(\frac{h_{1e}(t) - h_{2e}(t)}{R_2}\right)\right) \tag{59}$$

3.2.2.15.2 Vista

Una vez que el modelo se ha definido se construye la vista de la animación, la cual se basa en el sistema de los tanques en paralelo de la Figura 3-37, como lo muestra la Figura 3-39.

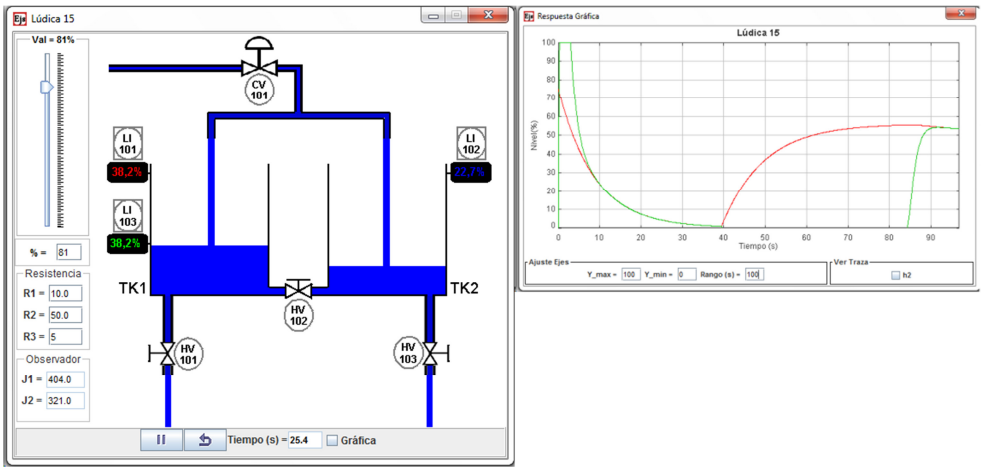


Figura 3-39 Vista de la lúdica 15: Tanques en Paralelo 2

Por medio del botón " / " se inicia o pone en pausa la ejecución de la animación y con el botón "" se reinicia de nuevo la animación. Con la barra de deslizamiento vertical se manipula el porcentaje de apertura de la válvula CV101, el cual también puede ser modificado por medio del campo "%". Por medio de campos numéricos se modifican los valores de las resistencias al paso del flujo de salida "R1", "R2" y "R3". En el recuadro "Observador" se registran los valores calculados de las ganancias del observador "J1" y "J2". Finalmente seleccionando "Gráfica", se abre una ventana emergente que muestra el comportamiento gráfico de los niveles a través del tiempo, además esta gráfica cuenta con campos numéricos que ajustan la visualización de las señales.

Las unidades y el valor de las variables más relevantes utilizadas en la animación son las mismas especificadas en la Tabla 3-12.

3.2.2.15.3 Resultados de aprendizaje esperados

En esta lúdica animada se implementa un observador que estima el nivel h_1 denominado h_{1e}, a partir de la medida del flujo entre los tanques, en donde de manera interactiva el estudiante varía parámetros del sistema y manipula el porcentaje de apertura de la válvula, buscando observar el comportamiento del nivel h_{1e}. Inicialmente analizará el comportamiento del sistema con valores diferentes de las resistencias R_1 y R_3, observando que h_{1e} tiende a ser igual a h_1. Posteriormente observará que si los valores de R_1 y R_3 son iguales, h_{1e} no tiende a seguir el nivel h_1 y concluirá que bajo esta condición no es posible estimar el nivel h_1 a partir de la medida del flujo entre los tanques, siendo el sistema no Observable.

3.2.3 Uso y pruebas de funcionamiento

Las Lúdicas Animadas se usan como etapa previa al estudio de los temas, con la finalidad de introducir y facilitar la comprensión de conceptos de la teoría de los sistemas de control; en ellas los estudiantes entienden su funcionamiento, desarrollan unas actividades y analizan unas preguntas de discusión.

El uso y prueba de las 15 Lúdicas Animadas se reportaron en el documento: *Guía profesor animaciones*, el cual se encuentra disponible en el Anexo A; en esta guía se presenta claramente los objetivos y resultados de aprendizaje esperados del uso de las Lúdicas Animadas como recurso de apoyo para el aprendizaje de conceptos en la educación en control. Con el fin de ilustrar el uso y prueba de una de las lúdicas, a continuación se muestra el uso de la lúdica animada "Válvula y Tanque" para entender el concepto del modelado matemático de un sistema.

3.2.3.1 Uso y prueba de la lúdica animada Válvula y Tanque

Esta lúdica tiene por presentación la página web que se muestra en la Figura 3-40.

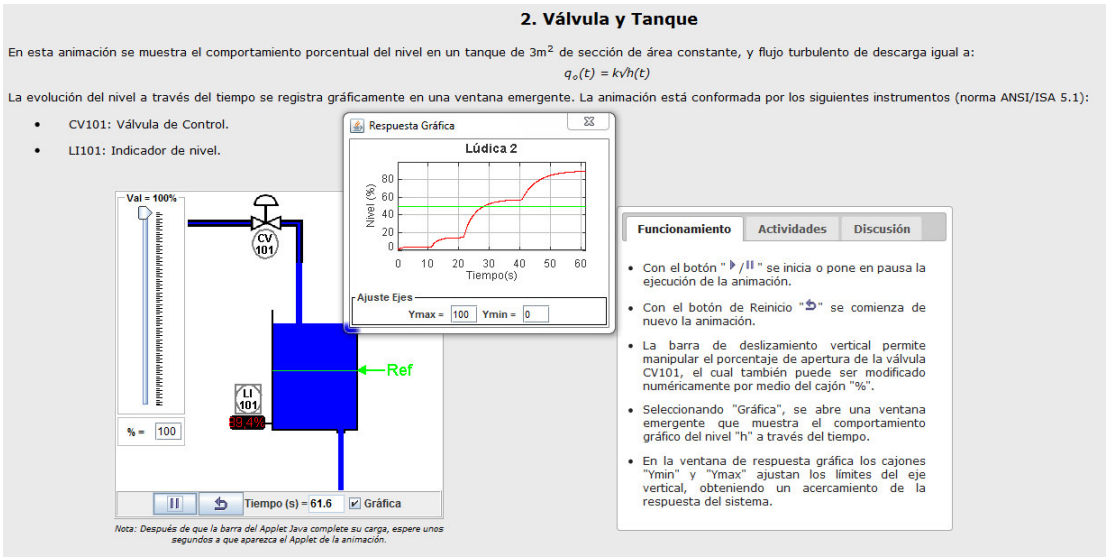


Figura 3-40 Lúdica animada Válvula y Tanque

Donde el contenido de los aspectos Funcionamiento, Actividades y Discusión se muestra en la Figura 3-41.

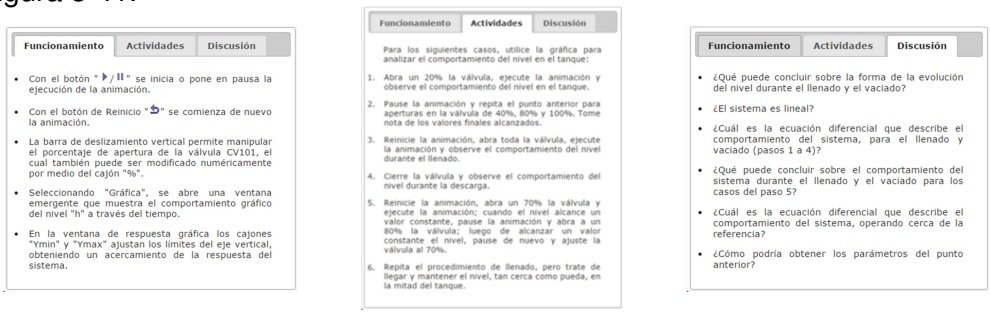


Figura 3-41 Contenido de los aspectos de la lúdica animada Válvula y Tanque

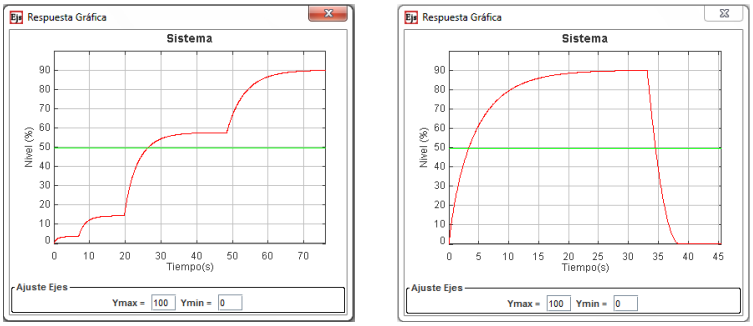
3.2.3.1.1 Objetivos, procedimiento y resultados

Para el aprendizaje del concepto de modelado matemático de un sistema, esta lúdica animada plantea los siguientes objetivos de aprendizaje:

- Identificar el comportamiento de un sistema lineal y no lineal
- Determinar el modelo matemático no lineal que describe el comportamiento del sistema
- Determinar el modelo simplificado lineal del sistema, operando alrededor de un punto de operación

Para cumplir estos objetivos se espera que el estudiante realice el siguiente procedimiento y obtenga los siguientes resultados:

En las actividades 1 y 2, manipulará la apertura de la válvula del 0 al 20%, del 20% al 40% y del 80% al 100%, con lo cual obtendrá escalones del flujo de entrada del 20%; mientras que en las actividades 3 y 4, manipulará la apertura y cierre de la válvula del 0 al 100%, y del 100% al 0% respectivamente, obteniendo un escalón positivo y negativo de máxima amplitud del flujo de entrada. Con estas variaciones de la apertura de la válvula obtendrá los comportamientos de la Figura 3-42.



a) Escalones del 20% b) Escalón de 0-100% y 100-0%

Figura 3-42 Comportamientos obtenidos del nivel para diferentes escalones del flujo de entrada

Al analizar los comportamientos, observará que la evolución del nivel en velocidad de respuesta y valor de ganancia estática, es diferente para cada caso y determinará que el sistema se comporta de manera no lineal. Este comportamiento lo describirá por medio de la siguiente ecuación diferencial:

$$A\frac{dh(t)}{dt} = q_i(t) - k\sqrt{h(t)}\ , h(t_0) = h_0,$$

donde *h(t)* es el nivel, $q_i(t)$ es el caudal de entrada, *A* es el área del tanque y el caudal de salida es proporcional a la raíz cuadrada del nivel.

En las actividades 5 y 6 seleccionará como punto de operación el 70% de apertura de la válvula y evaluará el comportamiento del sistema para pequeñas variaciones del flujo de entrada (10%) alrededor de este punto de operación, obteniendo el comportamiento de la Figura 3-43.

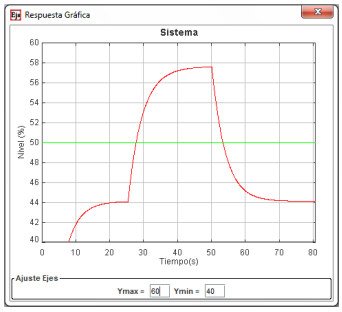


Figura 3-43 Comportamiento obtenido del nivel alrededor de un punto de operación

En este caso, observará que la evolución del nivel en velocidad de respuesta y valor de ganancia estática, es muy similar durante la apertura y cierre, y determinará que bajo estas condiciones, el sistema se comporta de manera lineal. Este comportamiento lo describirá por medio de la siguiente ecuación diferencial:

$$A\frac{dh(t)}{dt} = q_i(t) - \frac{h(t)}{R}\ , h(t_0) = h_0,$$

donde se define a *R* como la resistencia al paso del flujo en la descarga.

Comentario: Las pruebas realizadas basadas en el análisis de la lúdica animada “Válvula y Tanque” fueron satisfactorias; ya que permitieron observar y analizar el comportamiento de un sistema no lineal y lineal, obteniendo los respectivos modelos matemáticos, cumpliendo de esta manera con los objetivos de aprendizaje planteados.

3.3 Herramienta de Análisis del controlador PID

El objetivo principal de los sistemas de control es cumplir con las especificaciones deseadas de desempeño que deben lograrse con un diseño apropiado de la estrategia de control. En los sistemas de control la estrategia de control más ampliamente utilizada es actuar mediante la Proporción, Integración y/o Derivación, PID del error del sistema. El controlador PID se usa en cerca del 95% de los lazos de control industriales, por su sencillez, facilidad de ajuste, flexibilidad y robustez [53].

La Herramienta de Análisis del Controlador PID HAPID diseñada es una aplicación que permite realizar el análisis de un controlador PID práctico, ya que implementa funciones de operación y mantenimiento a partir de la generación de diez fallas en el lazo de control. Consiste de una animación interactiva desarrollada con la herramienta EJS, que recrea un ambiente de *Supervisory Control And Data Acquisition SCADA* para un intercambiador de calor controlado con el PID.

En esta sección inicialmente se presentan los requerimientos generales de funcionamiento y de aspecto gráfico de la aplicación. Posteriormente se muestra el diseño y desarrollo de siete paneles que conforman el ambiente SCADA de HAPID: Proceso, Acceso, Control, Eventos, Fallas, Válvula y Transmisor; donde se consideran ciertas generalidades de funcionamiento. Finalmente se muestran pruebas de funcionamiento, basadas en la implementación práctica de la solución de un problema de control, donde se analiza el efecto de una falla para su diagnóstico y reparación. El efecto de las 10 fallas en el comportamiento del sistema, su diagnóstico y procedimiento de reparación se reportaron en el documento: *Guía reparación fallas*, el cual se encuentra disponible en el Anexo B.

Con el fin de ilustrar el correcto manejo y operación de HAPID, se elaboró el documento: *Guía HAPID usuario*, el cual se encuentra disponible en el Anexo C. En esta guía se presentan los requerimientos para su ejecución, aspectos generales y de funcionamiento; y se explica en detalle la función de cada elemento de los paneles que conforman la aplicación.

3.3.1 Requerimientos

A continuación se presentan los requerimientos funcionales y no funcionales los cuales se han separado por módulos o paneles, y el esquema de las interfaces graficas de usuario.

3.3.1.1 Requerimientos funcionales

El panel del Proceso debe:
- Permitir el inicio o parada del proceso.
- Permitir el acceso a seis paneles: Acceso, Control, Eventos, Fallas Válvula y Transmisor.
- Permitir guardar en un archivo el valor de los parámetros de configuración.

El panel de Acceso debe:
- Permitir el acceso al sistema por medio de tres tipos de usuario: Ingeniero, Supervisor y Operario.
- Permitir la asignación de permisos a los usuarios.

El panel de Control debe:
- Permitir la visualización de la Variable de Proceso PV.
- Permitir la visualización y ajuste de la Variable Manipulada MV y de la referencia o Set-Point SP.
- Permitir seleccionar el modo de operación del controlador como Manual o Automático.
- Permitir el ajuste de los parámetros del controlador.
- Graficar el comportamiento temporal de las variables SP, PV y MV.
- Permitir guardar en un archivo los datos de las variables SP, PV, MV y del tiempo.

El panel de Eventos debe:
- Registrar el suceso de eventos.
- Permitir el reconocimiento de alarmas.
- Permitir el ajuste de los límites alarmables de las variables PV y MV.
- Permitir guardar en un archivo el registro de los eventos.

Los paneles de la Válvula y del Transmisor deben:
- Permitir el ajuste de los parámetros de la válvula de control.
- Permitir el ajuste de los parámetros del transmisor de temperatura.

El panel de Fallas debe:
- Permitir seleccionar entre diez tipos de fallas.

3.3.1.2 Requerimientos no funcionales

- Se utilizaran tres niveles de jerarquía: 1. Ingeniero, 2. Supervisor, 3. Operario.
- Se utilizará la estructura estándar o forma ISA de la acción de control PID.
- Se utilizará una dinámica del proceso con constante de tiempo (τ) de 1 segundo y tiempo muerto (T_m) de 0.5 segundos.
- Se utilizará el filtrado de la acción derivativa.
- Se utilizará el antiembalamiento de la acción integral.
- Se utilizará la transferencia sin saltos entre modos de operación en el controlador.
- Se utilizará una válvula de control con comportamiento de Histéresis, banda muerta y característica de caudal: lineal o isoporcentual.
- Se utilizará un transmisor de temperatura con ajuste de Zero, Span y filtro de antiplegamiento frecuencial.
- Se debe utilizar el software EJS, versión 5.0 como herramienta de desarrollo.
- Se debe utilizar el entorno de ejecución Java (JRE), versión 1.7 o superior.

3.3.1.3 Interfaz gráfica de usuario

Teniendo en cuenta las funcionalidades de la aplicación HAPID y con la finalidad de recrear el ambiente SCADA, se determinó que la interfaz gráfica de usuario debe estar conformada por un panel principal (donde se muestre el mímico del proceso) y seis paneles auxiliares (para acceder a funcionalidades y configuraciones): Acceso, Control, Eventos, Fallas, Válvula y Transmisor, conformados por las siguientes áreas:

<u>Panel del proceso:</u>
- Área 1: Grafica del mímico del proceso.
- Área 2: Menú con botones para seleccionar los paneles: Acceso, Control, Eventos y Fallas.

Panel de Acceso:
- Área 3: Lista desplegable con los usuarios: Ingeniero, Supervisor y Operario.
- Área 4: Campos para el ingreso del nombre de usuario y contraseña.
- Área 5: Botones de acceso.

Panel de Control:
- Área 6: Panel frontal del PID.
- Área 7: Comportamiento grafico de las variables del proceso.
- Área 8: Botones de configuración del PID.

Panel de Eventos:
- Área 9: Tabla con registro de eventos.
- Área 10: Botones de configuración de los eventos.

Panel de la Válvula y Transmisor:
- Área 11: Campos numéricos para el ajuste de parámetros.
- Área 12: Graficas de comportamientos.

Panel de Fallas:
- Área 13: Casillas de verificación para seleccionar entre 10 fallas.
- Área 14: Área de mensajes con información acerca de la falla seleccionada.

Por tanto, la organización y presentación gráfica general que deben tener los paneles de la aplicación HAPID, se muestra en la Figura 3-44.

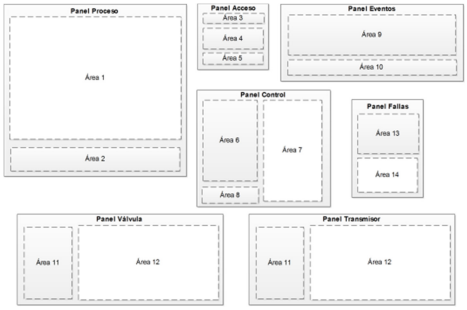


Figura 3-44 Esquema de las interfaces gráficas de usuario de la aplicación HAPID

3.3.2 Diseño y desarrollo

La aplicación HAPID consiste de una animación interactiva desarrollada con la herramienta EJS, que recrea un ambiente SCADA para un intercambiador de calor; la cual se caracteriza por presentar las siguientes funcionalidades: control de acceso con tres niveles de jerarquía (Ingeniero, Supervisor y Operario), gestión de permisos, generación y reconocimiento de alarmas, ajuste de parámetros del PID, modo de operación (Manual - Automático), configuración de la válvula de control y del transmisor de temperatura, registro de eventos, elaboración de tres tipos de reportes (configuración, datos y eventos), y generación de diez fallas típicas que afectan los elementos (controlador, actuador y sensor) del lazo de control.

La aplicación diseñada cuenta con una interfaz gráfica de usuario o panel principal donde se implementa el mímico del proceso y un menú para el acceso a paneles auxiliares, como lo muestra la Figura 3-45, donde se han diferenciado dos áreas.

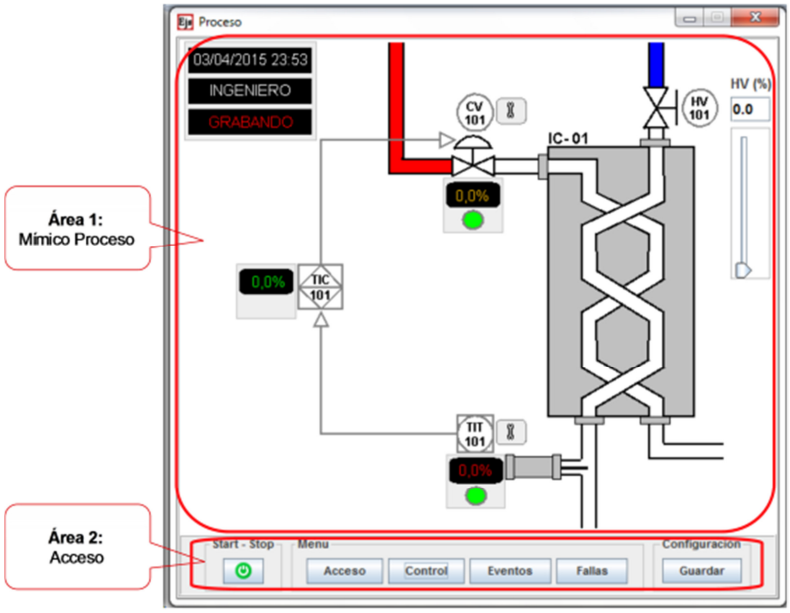


Figura 3-45 Interfaz del panel Principal del Proceso de la aplicación HAPID

Área 1: Donde se representa de manera animada el mímico del proceso con instrumentación y lazo de control según Norma ANSI/ISA 5.1 [56]. La nomenclatura de los elementos indica:

- IC-01: Intercambiador de Calor
- TIC-101: Controlador Indicador de Temperatura
- TIT-101: Transmisor Indicador de Temperatura (con piloto indicador de alarma)
- CV-101: Válvula de Control (con piloto indicador de alarma)
- HV-101: Válvula de ajuste Manual

En la parte superior izquierda de la interfaz se encuentra una pantalla de información con tres campos donde se muestra: la fecha y hora actual, el tipo de usuario y un mensaje que indica si se están grabando los datos.

Área 2: Donde por medio de botones se inicia o detiene el proceso, se accede a los paneles auxiliares: Acceso, Control, Eventos y Fallas, y se guarda en un archivo de texto valores de parámetros de configuración del SCADA.

3.3.2.1 Generalidades

HAPID se caracteriza por presentar aspectos de implementación y funcionamiento como: un lazo de control de temperatura, el cual por motivos de diseño considera el manejo de dos tiempos de muestreo; una jerarquía de tres niveles representada por tres tipos de usuario; y un procedimiento común para generar y guardar tres tipos de reportes. A continuación se explica en detalle estos aspectos.

3.3.2.1.1 Lazo de control, normalización y muestreo

En HAPID se implementa un control de temperatura de acuerdo al esquema mostrado en la Figura 3-46.

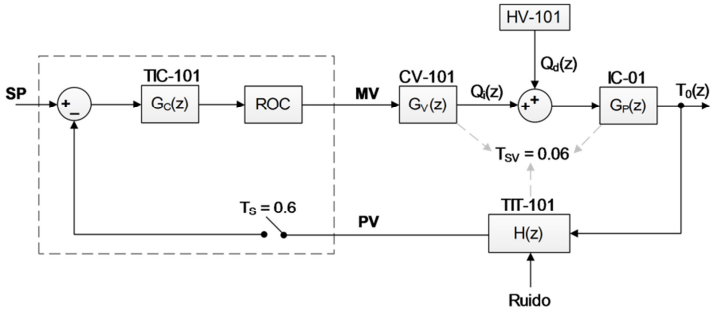


Figura 3-46 Esquema del lazo de control de temperatura que implementa HAPID

Donde MV es la variable manipulada, PV es la variable del proceso y SP es la referencia deseada o Set-Point; $Q_i(z)$ es el flujo de calor de entrada y $T_o(z)$ la temperatura del flujo de calor de salida del IC-01. El disturbio consiste de un flujo externo $Q_d(z)$ que afecta el flujo de entrada al IC-01, el cual es ajustable por medio del porcentaje de apertura de la válvula HV-101, y el ruido es una señal aleatoria que afecta el elemento de medición.

La planta $G_P(z)$ objeto del control es el IC-01, el controlador TIC-101 ($G_C(z)$) se encarga de manipular el porcentaje de apertura de la válvula CV-101 (actuador $G_V(z)$) y a su vez el flujo de entrada a la planta, con el fin de que la temperatura de salida se comporte de acuerdo a una referencia establecida; el elemento de medición TIT-101 (H(z)) permite hacer la medición de la temperatura de salida de la planta a través de sensores.

Normalización:
Con el fin de que las variables SP, PV y MV tengan rangos de operación similar, estas se relacionan con las variables de flujo y temperatura del proceso de la siguiente manera:

- Las variables SP y PV expresan valores porcentuales en el rango del 0 al 100% y representan la temperatura deseada y medida respectivamente, las cuales se encuentran en el rango de operación de 0 a 100℃.
- La variable MV expresa valores porcentuales en el rango del 0 al 100% y representa el flujo de calor de entrada, el cual se encuentra en el rango de operación de 0 a 100 m^3/s.

Muestreo:
Por motivos de diseño, el lazo de control considera el manejo de dos tiempos de muestreo, un tiempo de muestreo T_S para el controlador $G_C(z)$ y otro tiempo de muestreo de visualización T_{SV} para la válvula $G_V(z)$, la planta $G_P(z)$ y el sensor H(z). La selección de T_S se debe realizar para la velocidad del sistema en red cerrada y teniendo en cuenta el siguiente criterio [53]:

$$0.2 \leq w_n T_S \leq 0.6 \tag{60}$$

Asumiendo un comportamiento del sistema en red cerrada con $\rho = 0.8$ y $w_n = 1\ rad/s$, se obtiene:

$$0.2 \leq T_S \leq 0.6 \tag{61}$$

Como se debe cumplir que $T_m < T_s$, (siendo el tiempo muerto del proceso $T_m = 0.5$ segundos, según requerimiento definido en la sección 3.3.1.2) se seleccionó un valor de $T_S = 0.6$ segundos. Con el fin de simular un comportamiento continuo del proceso, de la válvula de control y del transmisor se selecciono T_{SV} diez veces menor que el valor obtenido para T_S, siendo entonces $T_{SV} = 0.06$ segundos.

3.3.2.1.2 Jerarquía de acceso

HAPID presenta una jerarquía de tres niveles: Ingeniero, Supervisor y Operario, para acceder a paneles y funcionalidades, como lo muestra la Figura 3-47.

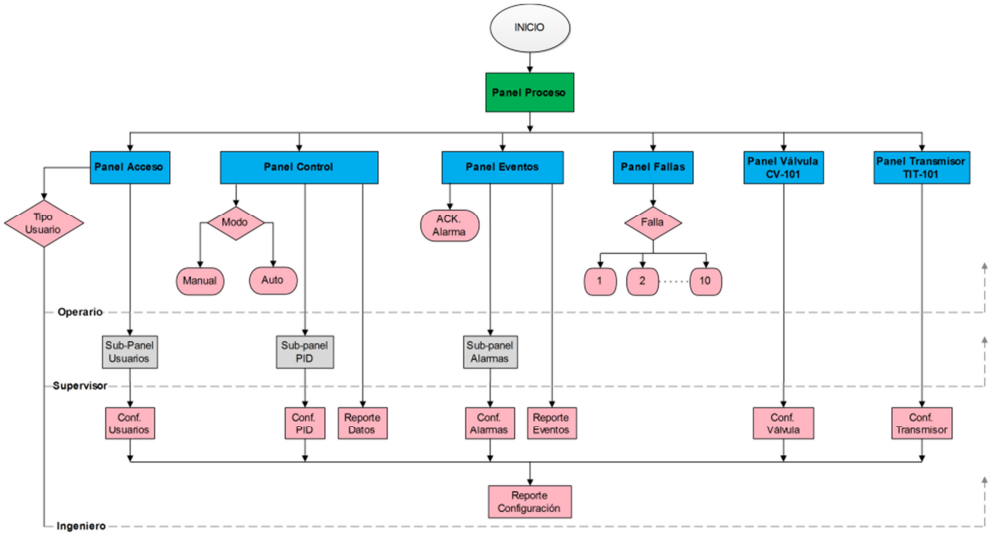


Figura 3-47 Diagrama de flujo del nivel de acceso a paneles y funcionalidades en HAPID

Donde el bloque de color verde corresponde al panel principal del proceso, los bloques de color azul a los paneles auxiliares: Acceso, Control, Eventos, Fallas, Válvula y Transmisor; y los bloques de color gris a los sub-paneles: Usuarios, PID y Alarmas. Los bloques de color rosa representan funcionalidades como: acceso por medio de tres tipos de usuario (Ingeniero, Supervisor y Operario), modo de operación (manual o automático), generación de diez tipos de fallas, reconocimiento (ACK) y configuración (Conf.) de alarmas, gestión de permisos de los usuarios, ajuste de los parámetros del PID, configuración de la válvula de control y del transmisor de temperatura, y elaboración de tres tipos de reportes (datos, eventos y configuración).

Por medio del panel principal se accede a los paneles auxiliares, los cuales presentan sub-paneles y funcionalidades accesibles según el tipo de usuario ingresado; siendo el usuario Ingeniero el nivel de mayor acceso.

3.3.2.1.3 Tipos de reportes

HAPID se caracteriza por generar tres tipos de reportes: Configuración, Datos y Eventos, como lo muestra la Figura 3-48.

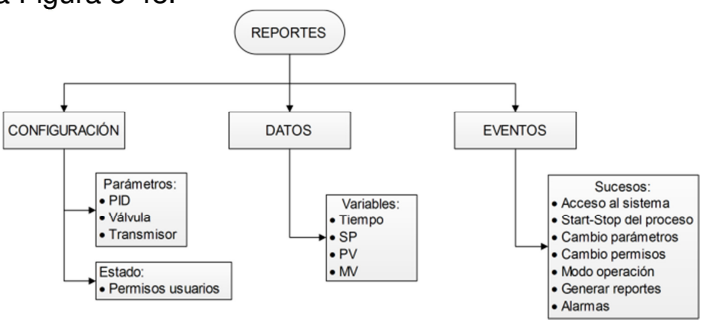


Figura 3-48 Tipos de reportes en HAPID

En Configuración se guardan los valores de los parámetros de los elementos del lazo de control y el estado de los permisos de los usuarios. En Datos se guarda el valor de las variables del proceso y el tiempo de simulación. En Eventos se guarda el suceso de los eventos registrados durante la operación del sistema.

El reporte se guarda en un archivo de texto, al cual se le asigna un nombre de acuerdo al formato general de la Figura 3-49.

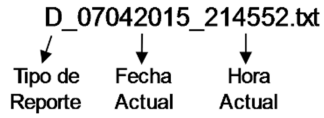


Figura 3-49 Formato del nombre asignado al reporte

El primer campo indica el tipo de reporte: D (Datos), E (Eventos) y C (Configuración), el segundo campo la fecha actual (*día/mes/año)*, y el tercer campo la hora actual (*hora/min/seg)*. Cada reporte se crea y guarda de acuerdo al algoritmo de la Figura 3-50.

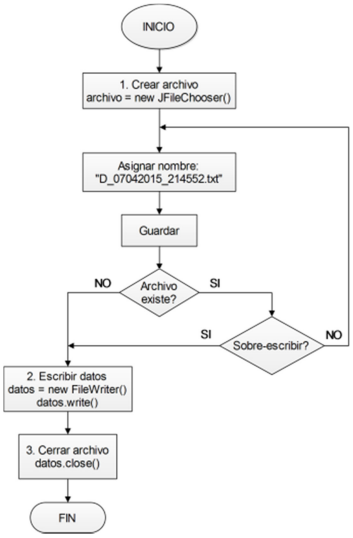


Figura 3-50 Algoritmo para crear y guardar un reporte

Básicamente el reporte se crea y guarda en dos pasos: 1) por medio de la clase *JFileChooser* se crea y abre el archivo, y se selecciona su ruta de ubicación; 2) con la clase *FileWriter* se escribe la información y se cierra el archivo para finalmente guardarlo en la ruta especificada. Las ventanas emergentes que resultan del procedimiento de guardar un reporte se muestran en las capturas de pantalla de la Figura 3-51.

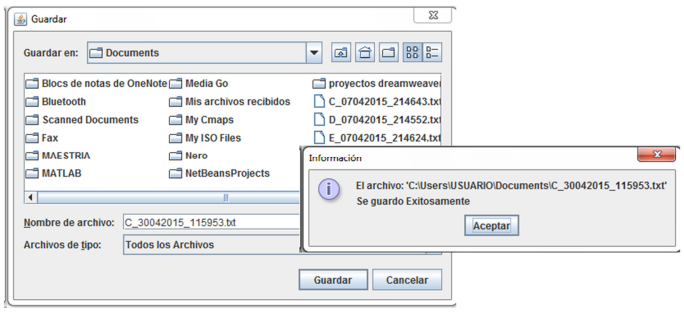


Figura 3-51 Ventanas emergentes para guardar un reporte

3.3.2.2 Panel de Proceso

Como se mencionó anteriormente en este panel se representa de manera animada el mímico del proceso con instrumentación y lazo de control. Para implementar la animación del proceso en EJS se debe definir el *modelo matemático* que simule el comportamiento del Intercambiador de Calor IC-01 como se explica a continuación.

3.3.2.2.1 Modelo del proceso

La definición del modelo del proceso, se basa en analizar el comportamiento de la temperatura de salida en el IC-01, como lo muestra la Figura 3-52.

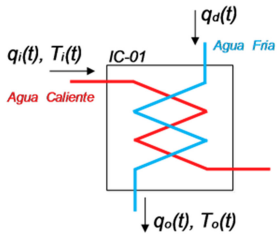


Figura 3-52 Esquema del Intercambiador de Calor IC-01

Donde, $T_i(t)$ es la temperatura del flujo de calor de entrada $q_i(t)$, $T_o(t)$ es la temperatura del flujo de calor de salida $q_o(t)$, y $q_d(t)$ es un flujo externo de agua fría, el cual se considera como un disturbio. Para simplificar el análisis, se considera el IC-01 como un cuerpo homogéneo con almacenamiento térmico [53], como lo muestra la Figura 3-53.

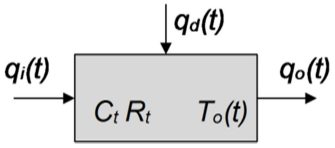


Figura 3-53 Calentamiento de un cuerpo homogéneo

Donde el cuerpo presenta una capacitancia térmica C_t y una resistencia R_t. Por tanto el modelo matemático se obtiene de un balance de energía en estado dinámico [57]:

$$C_t \frac{dT_o(t)}{dt} = q_i(t - T_m) + q_d(t) - q_o(t), \tag{62}$$

donde, T_m se considera el tiempo muerto por la conducción del calor, entre la fuente de calor y la medida de temperatura. El flujo de calor $q_o(t)$ viene dado por:

$$q_o = \frac{T_o(t)}{R_t} \tag{63}$$

Reemplazando y organizando se obtiene:

$$\tau_t \frac{dT_o(t)}{dt} + T_o(t) = R_t[q_i(t - T_m) + q_d(t)], \tag{64}$$

donde $\tau_t = C_t R_t$ es la constante de tiempo térmica. Aplicando la transformada de Laplace se obtiene la Función De Transferencia FDT que relaciona $T_o(t)$ con $q_i(t)$:

$$T_o(s) = \frac{R_t e^{-T_m s}\big(Q_i(s) + Q_d(s)\big)}{\tau_t s + 1} \tag{65}$$

El valor de las variables de esta FDT se especifican en la Tabla 3-13.

Tabla 3-13 Unidades y valores de las variables del proceso

Variable	Descripción	Unidades	Valor	
			Min.	Max.
T_o	Temperatura del proceso	℃	14.28	100
q_i	Flujo de calor de entrada	m^3/s	0	100
q_d	Flujo externo	m^3/s	20	40
C_t	Capacitancia térmica	Julio/Kelvin	1.4	
R_t	Resistencia térmica	Kelvin/Watts	0.714	
τ_t	Constante de tiempo térmica	s	1	
T_m	Tiempo muerto	s	0.5	
T_{SV}	Tiempo de muestreo de visualización	s	0.06	

Se debe tener en cuenta que el flujo externo $q_d(t)$, se ajusta con el porcentaje de apertura de la válvula HV-101 la cual presenta un rango de apertura de 0-100%. En este caso para 0% se tiene un flujo de 40 m^3/s y para el 100% un flujo del 20 m^3/s, buscando que la temperatura del proceso disminuya a medida que se incrementa el porcentaje de apertura de HV-101, es decir a medida que se aumenta el flujo externo de agua fría de entrada al proceso. Por defecto HV-101 presenta un apertura del 0%, siendo el rango de operación de la temperatura del proceso de 28.56 – 100℃.

En EJS la FDT de la ecuación (65) se debe implementar como ecuaciones de diferencia, para ello inicialmente se obtuvo la representación discreta con el comando *c2d()* de Matlab (para un tiempo de muestreo T_{SV} de 0.06 segundos y método "*tustin*"):

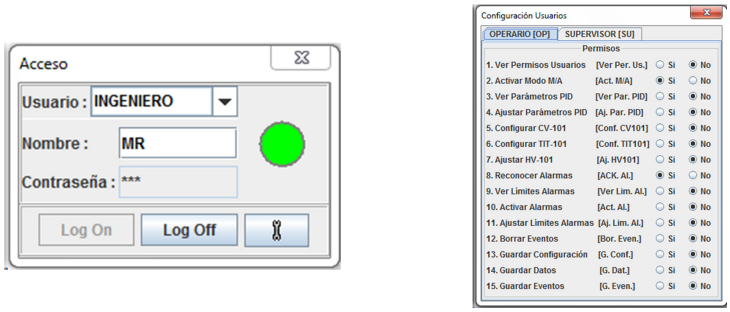
$$\frac{T_o(z)}{Q_i(z)} = Z^{-8}\,\frac{(0.02913z + 0.02913)R_t}{z - 0.9417} \tag{66}$$

Y posteriormente se expresó como una ecuación de diferencia:

$$T_o(k) = T_o(k-1)0.9417 + \big(Q_i(k-8) + Q_i(k-9)\big)0.02913R_t \tag{67}$$

3.3.2.3 Panel de Acceso

Este panel implementa la interfaz para el control de acceso que cuenta con un sub-panel de configuración de permisos, como lo muestra la Figura 3-54.



a) Panel de Acceso b) Panel de Configuración de permisos

Figura 3-54 Interfaz del panel de Acceso y sub-panel de Configuración

El panel de Acceso presenta las siguientes características:

- Lista desplegable para seleccionar entre tres tipos de usuario: Operario, Supervisor e Ingeniero.

- Campos de texto y numéricos para el ingreso de un nombre de identificación y contraseña, por medio de los cuales se realiza la respectiva validación y acceso a través del botón “Log On”.
- Sub-panel de configuración para asignar permisos a los usuarios: Operario y Supervisor; por defecto los permisos otorgados para configurar las funcionalidades y acceder a los sub-paneles, se muestran en la Tabla 3-14, siendo el usuario Ingeniero el nivel que tiene acceso a todos los permisos y por medio del cual se le otorgan los permisos a los otros dos usuarios.

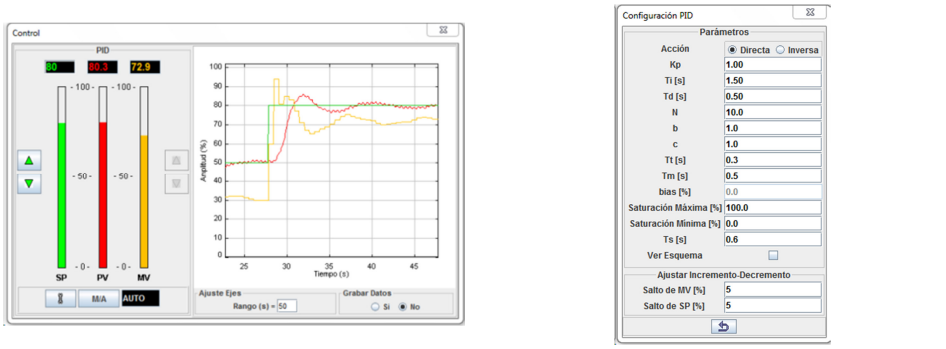
Tabla 3-14 Permisos por defecto de los usuarios en HAPID

Permiso		Usuario		
		Operario	Supervisor	Ingeniero
Funcionalidades	Otorgar Permisos a los usuarios	NA	NA	✓
	Activar Modo de operación Manual o Automático	✓	✓	✓
	Ajustar parámetros del controlador PID	✗	✗	✓
	Ajustar parámetros de la Válvula CV-101	✗	✗	✓
	Ajustar parámetros del Transmisor TIT-101	✗	✗	✓
	Ajustar la Válvula HV-101	✗	✗	✓
	Reconocer Alarmas	✓	✓	✓
	Ajustar parámetros de las Alarmas	✗	✗	✓
	Activar la generación de Alarmas	✗	✓	✓
	Borrar registro de los Eventos	✗	✗	✓
	Guardar parámetros de configuración del SCADA	✗	✗	✓
	Guardar Datos de las variables del proceso	✗	✗	✓
	Guardar registro de Eventos	✗	✗	✓
	Generar Fallas	✓	✓	✓
Sub-panel	Ver configuración de los usuarios	✗	✓	✓
	Ver configuración del PID	✗	✓	✓
	Ver configuración de la CV-101	✓	✓	✓
	Ver configuración del TIT-101	✓	✓	✓
	Ver configuración de las Alarmas	✗	✓	✓
	Ver Fallas	✓	✓	✓

✓ = Permitido, ✗ = No permitido, NA = No Aplica

3.3.2.4 Panel de Control

Este panel implementa la interfaz para la manipulación y configuración del controlador PID, como lo muestra la Figura 3-55, la cual toma como referencia el panel frontal de un controlador industrial como el SIPART [58].



a) Panel de Control b) Panel de Configuración de parámetros

Figura 3-55 Interfaz del panel de Control y sub-panel de Configuración

El panel de Control presenta las siguientes características:

- Registro de la variable PV y ajuste de SP y MV.
- Selección del modo de operación del controlador como Manual o Automático.
- Área grafica donde se registra la evolución temporal de las variables SP, PV y MV.
- Opción para generar el reporte de Datos.
- Sub-panel de configuración para ajustar los parámetros del controlador.

3.3.2.4.1 Modelo del controlador

En HAPID se implementa un esquema de control que cuenta con aspectos prácticos de implementación como:

- Acción de control PID en estructura estándar o forma ISA [53], [59]:

$$U(s) = \pm\left[K_p\left(bR(s) - B(s) + \frac{1}{T_i s}\big(R(s) - B(s)\big) + \frac{T_d s}{1 + \frac{T_d}{N}s}\big(cR(s) - B(s)\big)\right) + A_b(s)\right], \quad (68)$$

  donde, *U(s)* es la señal de control, *R(s)* es la entrada de referencia, *B(s)* es la salida controlada, K_p es la ganancia proporcional, T_i es el tiempo integral, T_d es el tiempo derivativo, *N* es el factor del filtro de la acción Derivativa D, *b* y *c* son constantes de peso en la referencia para la acción Proporcional P y D, *Ab(s)* es la compensación manual (*bias*) en la señal de control para las acciones *P* o *PD*. El signo ± define la acción de control directa (+) o inversa (–).

- Filtrado de la derivada: Para evitar amplificar el ruido de medición, se adiciona un filtro de primer orden con constante de tiempo *Td/N* a la acción derivativa pura, como se observa en la ecuación (68).

- Antiembalamiento (*Anti-windup*) de la acción integral: Para evitar sobrevalores de la acción integral ante la saturación del actuador, se recalcula la parte integral del controlador [54].

- Cambio suave Manual – Automático: Para asegurar una transferencia suave sin saltos (*Bumpless*) en el cambio de modo manual a automático [54].

En la Figura 3-56 se muestra el esquema de control práctico implementado.

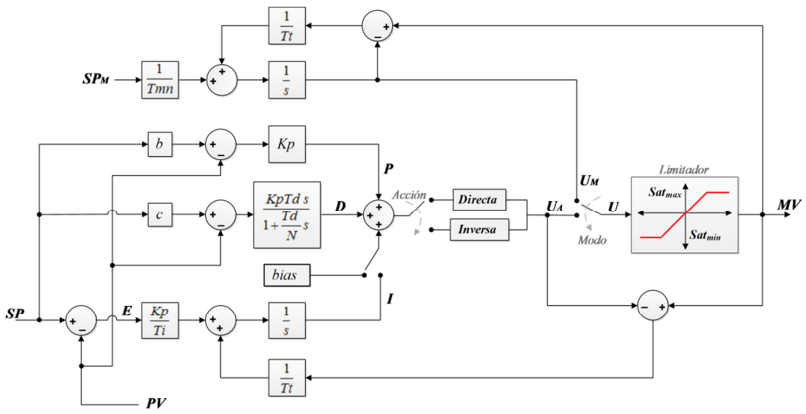


Figura 3-56 Estructura de control, con PID ISA, Bumpless y Anti-windup

Donde T_t es la constante de tiempo que determina con cuánta rapidez se resetea la acción integral, T_{mn} es la constante de tiempo que define la rapidez de cambio de la referencia manual SP_M, U_M es la señal de control del modo manual y U_A la del modo automático. Para implementar este esquema de control en EJS se debe determinar su equivalente discreto, el cual se obtiene de discretizar cada acción de control por separado, de la siguiente manera:

La acción Proporcional es:

$$P(s) = K_p\big(bS_P(s) - P_V(s)\big) + bias \tag{69}$$

Esta acción se discretiza simplemente reemplazando la variable continua por su versión muestreada, por tanto:

$$P(k) = K_p\big(bS_P(k) - P_V(k)\big) + bias \tag{70}$$

La acción Derivativa es:

$$D(s) = \frac{K_p T_d s}{1 + \frac{T_d}{N} s}\left(cS_P(s) - P_V(s)\right) \tag{71}$$

Esta acción se discretiza por medio de la aproximación de Euler rectangular hacia atrás:

$$\frac{T_d}{N}\frac{dD(t)}{dt} + D(t) = K_p T_d\left(c\frac{dS_P(t)}{dt} - \frac{dP_V(t)}{dt}\right)$$

$$\frac{T_d}{NT_s}[D(k) - D(k-1)] + D(k) = \frac{K_p T_d}{T_s}\big[c(S_P(k) - S_P(k-1)) - \big(P_V(k) - P_V(k-1)\big)\big]$$

Despejando $D(k)$ se obtiene:

$$D(k) = D_1 D(k-1) + D_2\big[c\big(S_P(k) - S_P(k-1)\big) - \big(P_V(k) - P_V(k-1)\big)\big],$$
$$con\ \ D_1 = \frac{T_d}{T_d + NT_s}\ \ y\ \ D_2 = \frac{K_p T_d N}{T_d + NT_s} \tag{72}$$

La acción Integral con el Anti-windup es:

$$I(s) = \left[\frac{K_p\big(S_P(s) - P_V(s)\big)}{T_i} + \frac{\big(M_V(s) - U_A(s)\big)}{T_t}\right]\frac{1}{s} \tag{73}$$

De igual manera se discretiza por medio de la aproximación rectangular hacia atrás:

$$I(k) = I(k-1) + I_1\big(S_P(k) - P_V(k)\big) + I_2\big(M_V(k) - U_A(k)\big)$$
$$con\ \ I_1 = \frac{K_p T_s}{T_i}\ \ y\ \ I_2 = \frac{T_s}{T_t} \tag{74}$$

El término U_M se discretiza igual que la acción integral:

$$U_M(s) = \left[\frac{S_{PM}(s)}{T_{mn}} + \frac{M_V(s) - U_M(s)}{T_t}\right]\frac{1}{s} \tag{75}$$

$$U_M(k) = U_M(k-1) + \frac{S_{PM}(k)T_s}{T_{mn}} + \frac{\big(M_V(k) - U_M(k)\big)T_s}{T_t} \tag{76}$$

Finalmente la señal de control U_A es la suma de las acciones:

$$U_A(k) = P(k) + I(k) + D(k) \tag{77}$$

Por tanto, el esquema de control de la Figura 3-56 se implementa en EJS con las ecuaciones (70), (72), (74) y (77) y se ejecuta de acuerdo al algoritmo de la Figura 3-57.

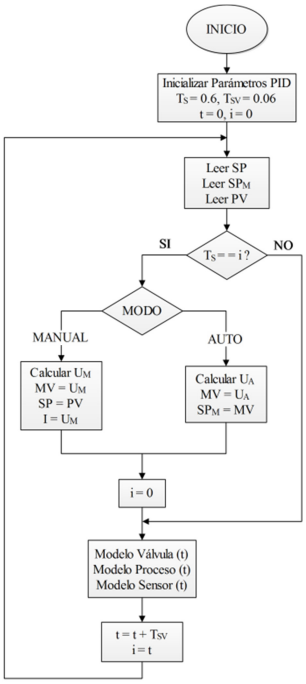


Figura 3-57 Algoritmo de ejecución de la ley de control en HAPID

El algoritmo de ejecución considera los dos tiempos de muestreo definidos T_S y T_{SV}. Primero se inicializan los parámetros del PID, se establecen los valores respectivos de los tiempos de muestreo, se inicializa la variable temporal *t* del sistema y una variable auxiliar *i*; y se leen las señales: SP, SP_M y PV. Posteriormente si no se cumple el tiempo de muestreo T_S del controlador, se actualizan los modelos de la válvula, del proceso y del sensor, incrementando la variable temporal y la variable auxiliar con el tiempo de muestreo T_{SV}, retornando la ejecución del algoritmo al estado de lectura de las variables del sistema. Después de diez iteraciones del algoritmo, T_S es igual T_{SV} y se ejecuta la ley de control, donde dependiendo del modo de operación se calcula la señal de control manual U_M o automática U_A y se actualizan estados del controlador; inicializando la variable auxiliar *i*, actualizando los modelos del sistema, y empezando nuevamente el ciclo.

En el diseño del controlador se asumió un rango y valor por defecto para los parámetros, como lo especifica la Tabla 3-15.

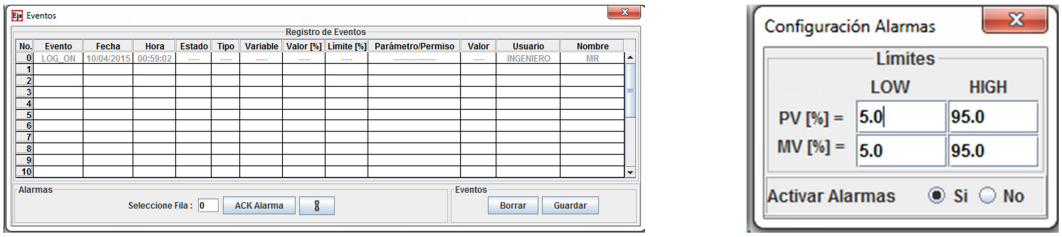
Tabla 3-15 Rangos y valores por defecto de los parámetros del controlador

Parámetro	Descripción	Und.	Valor		
			Min.	Max.	Def.
K_p	Ganancia Proporcional P	-	0	5	1
T_i	Tiempo de acción Integral I	s	0	5	1.5
T_d	Tiempo de acción Derivativa D	s	0	5	0.5
N	Factor del filtro de la acción D	-	2	20	10

b	Constante de peso en la referencia para la acción P	-	0	1	1
c	Constante de peso en la referencia para la acción D	-	1	1	1
T_t	Constante de tiempo para reiniciar la acción I	s	0	3	0.3
T_{mn}	Constante de tiempo para la referencia manual	s	0	3	0.5
bias	Factor de ajuste manual para las acciones P o PD	%	0	100	0
Sat.Max.	Límite máximo de saturación de la señal de control	%	0	100	100
Sat.Min	Límite mínimo de saturación de la señal de control	%	0	100	0
T_s	Tiempo de muestreo	s	-	-	0.6

3.3.2.5 Panel de Eventos

Este panel implementa la interfaz para el registro de eventos y configuración de alarmas, como lo muestra la Figura 3-58.



a) Panel de Eventos b) Panel de Configuración de Alarmas

Figura 3-58 Interfaz del panel de Eventos y sub-panel de Configuración

El panel de Eventos cuenta con una tabla de 21 filas y 12 columnas donde se registran 12 tipos de eventos abreviados por palabras claves, los cuales se registran dependiendo del actor que lo produjo, por la interacción del usuario o generado por el sistema (por causa de una alarma). El registro se caracteriza porque cada vez que ocurre un evento se registra la fecha y hora, el nombre y tipo de usuario. En la Tabla 3-16 se describe la palabra clave que caracteriza a cada evento. Adicionalmente este panel permite reconocer las alarmas y por medio del sub-panel de configuración se establecen los límites altos y bajos de alarma para las variables MV y PV.

Tabla 3-16 Descripción del tipo de evento

Actor	Evento	Descripción
Usuario	LOG_ON	Ingreso al sistema
	LOG_OFF	Salida del sistema
	START	Inicio del proceso
	STOP	Parada del proceso
	C_PER	Cambio de estado de un permiso, otorgado o negado
	AUTO	Selección del modo de operación Automática del controlador
	MANUAL	Selección del modo de operación Manual del controlador
	C_PARAM	Cambio de un parámetro en el Controlador, Válvula o Transmisor
	G_DAT	Reporte de Datos generado
	G_EVEN	Reporte de Eventos generado
	G_CONF	Reporte de Configuración generado
Sistema	ALARMA	Las variables MV y PV alcanzaron un límite de Alarma de tipo baja o alta.

Cuando ocurre un evento se ejecuta el algoritmo de la Figura 3-59.

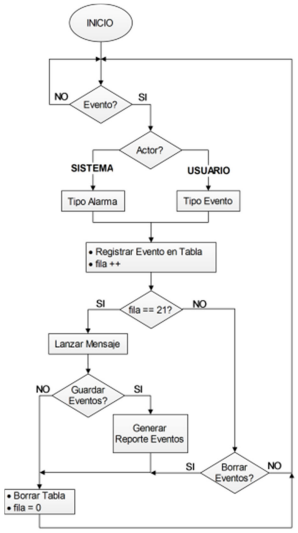


Figura 3-59 Algoritmo para registrar un evento

Inicialmente la ejecución del algoritmo permanece en espera por el suceso de un evento, cuando ocurre se registra el evento en la tabla según el actor que lo produjo; si la tabla alcanza el tamaño máximo de eventos registrados, se lanza un mensaje en una ventana emergente que informa lo sucedido y por medio del cual se permite guardar el registro de eventos antes de que estos sean borrados, como lo muestra la Figura 3-60; posteriormente el algoritmo continua con su ejecución, se borra la tabla y se empieza un nuevo registro de eventos.

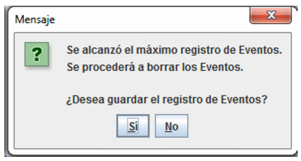


Figura 3-60 Ventana emergente con mensaje sobre el registro de eventos

En el algoritmo también se permite borrar de manera manual la tabla de eventos o generar el reporte, por medio de los botones “Borrar” y “Guardar” en el panel de eventos de la Figura 3-58.

3.3.2.6 Panel de la Válvula de Control

Este panel implementa la interfaz para configuración, ajuste de parámetros y visualización de comportamientos de la válvula de control, como lo muestra la Figura 3-61.

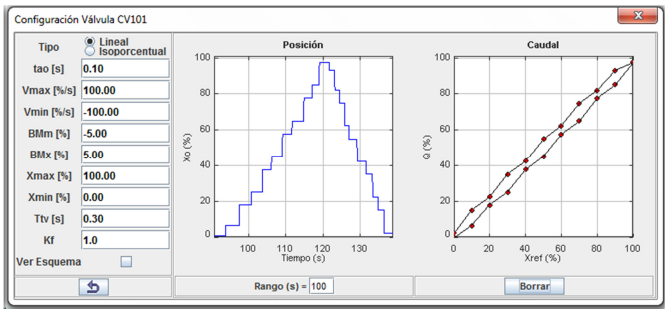


Figura 3-61 Interfaz de configuración de la Válvula de Control

Cuenta con campos numéricos para ajustar los parámetros del modelo de la válvula y dos áreas de registro gráfico: la primera registra la evolución temporal del desplazamiento porcentual de salida X_0 del vástago de la válvula, y la segunda registra el comportamiento estático del caudal Q versus el desplazamiento de referencia X_{ref} del vástago de la válvula.

Para implementar la dinámica de la válvula de control en EJS se debe definir el *modelo matemático* que simule su comportamiento, como se explica a continuación.

3.3.2.6.1 Modelo de la válvula

La válvula neumática es el elemento final de control más usado en el lazo de regulación, ya que realiza la función de variar el caudal del fluido de control que modifica a su vez el valor de la variable del proceso. Es un sistema que exhibe inherentemente una dinámica de segundo orden [60]. Por tanto, la definición del modelo matemático se basa en analizar el comportamiento de la válvula neumática de la Figura 3-62.

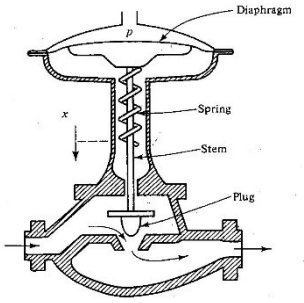


Figura 3-62 Partes de una válvula neumática. Imagen tomada de [60]

La posición del vástago (Stem), o de manera equivalente, la posición del asiento (Plug) al final del vástago, determina el tamaño de apertura para el flujo y por tanto la cantidad de flujo que pasa a través de ella. La posición del vástago es determinada por el balance de todas las fuerzas actuando sobre él:

$pA =$ Fuerza que actúa hacia abajo y es ejercida por la compresión del aire sobre el diafragma; donde p es la presión que abre o cierra la válvula y A es el área del diafragma.

$Kx =$ Fuerza que actúa hacia arriba y es ejercida por el resorte (muelle, Spring) unido al vástago y al diafragma; K es la constante de Hooke del resorte y x es el desplazamiento.

$C\frac{dx}{dt} =$ Fuerza de fricción ejercida hacia arriba por causa del contacto del vástago con el empaque de la válvula; donde C es el coeficiente de fricción entre el vástago y el empaque.

Aplicando la segunda ley de Newton, organizando y utilizando la transformada de Laplace, se obtiene la función de transferencia entre el desplazamiento $x(s)$ y la señal de presión $p(s)$ ejercida en la válvula.

$$\frac{x(s)}{p(s)} = \frac{A/K}{(M/Kg_c)s^2 + \frac{C}{K}s + 1}, \tag{78}$$

donde M es la masa del vástago y g_c es la gravedad. Usualmente, $M \ll Kg_c$ y la dinámica de la válvula se puede aproximar a un sistema de primer orden. Por tanto, la FDT que representa el modelo de la válvula viene dada por:

$$\frac{x_0(s)}{x_{ref}(s)} = \frac{1}{\tau s + K_f}, \tag{79}$$

donde, x_0 es el desplazamiento porcentual del vástago y x_{ref} es la señal de control; el área se considera unitaria, $\tau = C$ es la constante de tiempo, y $K_f = K$ es la ganancia estática de la válvula. Adicional al modelo, se consideran comportamientos no-lineales como:

- Histéresis
- Banda muerta
- Saturación

Estas alinealidades junto con el modelo de la ecuación (79) conforman la estructura de la válvula, la cual se implementó como lo muestra el esquema de la Figura 3-63, donde además se incluyen características inherentes del caudal como Lineal o Isoporcentual [61].

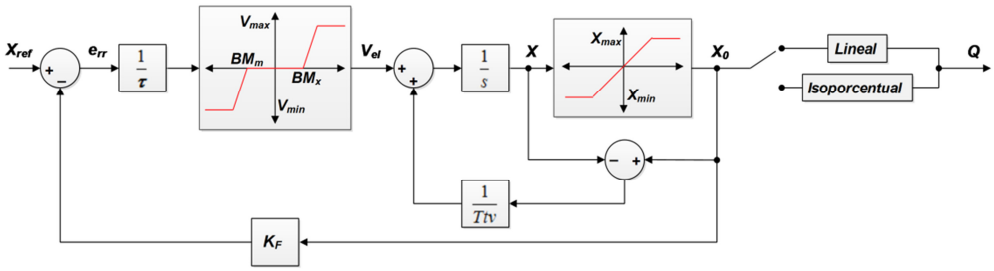


Figura 3-63 Modelo de la válvula de control con alinealidades

Donde, B_{Mm} y B_{Mx} definen los límites inferior y superior de la zona muerta, y V_{max} y V_{min} los límites máximos y mínimos de saturación de la velocidad; X_{max} y X_{min} corresponden a los límites máximos y mínimos de saturación de la posición, siendo T_{tv} la constante de tiempo que determina con cuánta rapidez se reinicia la parte integral y donde la característica del caudal viene dado por:

- Lineal: $$Q(k) = X_0(k) \tag{80}$$
- Isoporcentual: $$\frac{Q(k)}{Q_{max}} = \frac{1}{R} * R^{X_0(k)}, \tag{81}$$

donde el porcentaje del caudal está en función del campo de control o *Rangeabilidad (R)* de la válvula, la cual según la referencia [61], considera un valor típico de $R = 50$.

Para implementar la estructura de la válvula de la Figura 3-63 en EJS, se representa de manera discreta de la siguiente forma:

$$V_{el}(k) = x_{ref}(k) - x_0(k)K_f \tag{82}$$

$$E_V(k) = \frac{x_0(k) - x(k)}{T_{tv}} \tag{83}$$

$$x(k) = x(k-1) + \frac{T_{SV}}{\tau}\left(V_{el}(k) + E_V(k)\right) \tag{84}$$

En el diseño de la válvula de control se asumió un rango y valor por defecto para los parámetros, como lo especifica la Tabla 3-17.

Tabla 3-17 Rangos y valores por defecto de los parámetros de la válvula de control

Parámetro	Descripción	Und.	Valor		
			Min.	Max.	Def.
τ	Constante de tiempo de la dinámica de la válvula	s	0	1	0.1
V_{max}	Límite máximo de saturación de la velocidad	%/s	0	100	100
V_{min}	Límite mínimo de saturación de la velocidad	%/s	-100	0	-100
B_{Mm}	Límite inferior de la zona muerta	%	-10	0	-1
B_{Mx}	Límite superior de la zona muerta	%	0	10	1
X_{max}	Límite máximo de saturación de la posición	%	0	100	100
X_{min}	Límite mínimo de saturación de la posición	%	0	100	0
T_{tv}	Constante de tiempo para reiniciar la integral	s	0	3	0.3
K_f	Ganancia estática de la válvula	-	1	2	1

3.3.2.7 Panel del Transmisor de Temperatura

Este panel implementa la interfaz para configuración, ajuste de parámetros y visualización de comportamientos del transmisor de temperatura, como lo muestra la Figura 3-64.

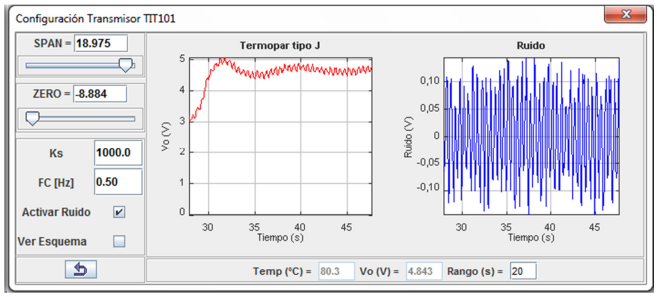


Figura 3-64 Interfaz de configuración del Transmisor de Temperatura

Cuenta con campos numéricos, para ajustar los parámetros del modelo del transmisor y dos áreas donde se registra gráficamente los comportamientos temporales, del voltaje de salida del termopar, y de la magnitud del Ruido presente en el transmisor.

Para implementar la dinámica del transmisor de temperatura en EJS se debe definir el *modelo matemático* que simule su comportamiento, como se explica a continuación.

3.3.2.7.1 Modelo del transmisor

La estructura del Transmisor de temperatura se implementa con cuatro bloques: Acondicionamiento, Ruido, Filtro y Calibración, como lo muestra la Figura 3-65.

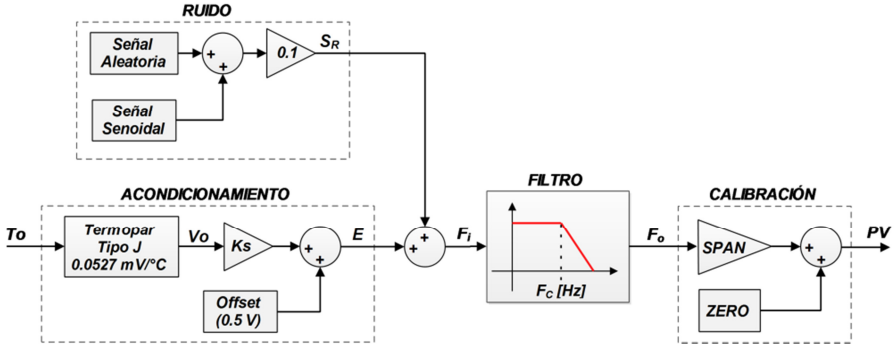


Figura 3-65 Estructura del transmisor de temperatura

Acondicionamiento:
El bloque acondicionamiento se conforma por el sensor de temperatura, una ganancia y un offset. El sensor es un termopar, el cual consiste de dos metales diferentes unidos en los extremos que genera una diferencia de potencial del orden de los milivoltios, siempre y cuando las uniones se mantengan a diferentes temperaturas, efecto *Seebeck* [61].

El tipo de termopar consiste en un tipo J, el cual se conforma por lo metales Hierro y Constantán. En la Tabla 3-18, se presentan las especificaciones técnicas más relevantes de este termopar [61].

Tabla 3-18 Especificaciones del termopar tipo J

Parámetro	Und.	Valor	
		Min.	Max.
Temperatura de entrada	℃	-40	750
Voltaje de salida	mV	-2	40
Rango	℃	-40 a 750	
Alcance	℃	790	
Error	-	± 1.5℃	

Para determinar la relación entre la temperatura del termopar y el voltaje generado se utilizó la tabla de valores del termopar tipo J citado en [61], de donde teniendo en cuenta que la temperatura del proceso está en el rango de 0 a 100℃ se tomaron los valores de los respectivos voltajes generados y se tabularon en la Tabla 3-19.

Tabla 3-19 Voltaje generado en función de la temperatura para el termopar tipo J

T_o [℃]	0	10	20	30	40	50	60	70	80	90	100
V_o [mV]	0.0	0.507	1.019	1.536	2.058	2.585	3.115	3.649	4.186	4.725	5.268

Se realizó una regresión lineal por el método de mínimos cuadrados con el comando "*polyfit*" de Matlab, obteniendo la ecuación que representa el comportamiento del termopar.

$$V_0(k) = (0.0527T_o(k) - 0.0318) * 0.001 \tag{85}$$

De esta ecuación se determina la sensibilidad del termopar la cual corresponde a la pendiente, siendo igual a *0.0527mV/℃*.

En el acondicionamiento, la ganancia *Ks*, se encarga de amplificar el voltaje de salida del termopar a una señal adecuada, en este caso se le asigna un valor de 1000 con el fin de representar la salida de la señal V_0 en voltios, además se considera un offset interno del sensor de 0.5 Voltios, con el fin de implementar posteriormente un esquema de calibración. Finalmente la señal acondicionada se representa por la siguiente ecuación:

$$E(k) = V_0(k)K_S + 0.5 \tag{86}$$

Ruido:
El ruido se genera por la suma entre una señal aleatoria y una senoidal. La señal aleatoria consiste de una función *random()* que entrega valores entre -0.5 y 0.5, y la señal senoidal presenta una amplitud unitaria y una frecuencia f_R de 2 Hertz. Por tanto, el comportamiento del ruido se representa por la siguiente ecuación:

$$\begin{gathered} S_R(k) = \big(sen\big(2\pi f_R t_R(k)\big) + random\big) * 0.1 \\ t_R(k) = t_R(k) + T_{SV} \end{gathered} \tag{87}$$

Filtro:
Para evitar el plegamiento de las señales de alta frecuencia de la señal medida en los sistemas de control digitales, se debe incluir un pre-filtro paso-bajo que limite el ancho de banda de la señal de realimentación. Sin embargo se debe considerar que el filtro es un elemento dinámico adicional en el lazo, el cual puede generar atrasos y afectar el desempeño del sistema de control [53].

En el transmisor se implementa un filtro de *Bessel*, ya que se caracteriza por tener una curva plana de magnitud versus la frecuencia lo que no distorsiona las señales; la función de transferencia que define este filtro viene dada por:

$$\frac{F_o(s)}{F_i(s)} = \frac{3}{\frac{s^2}{{\omega_C}^2} + \frac{3s}{\omega_C} + 3}, \tag{88}$$

donde ω_C es la frecuencia de corte del filtro en rad/s, la cual se seleccionó teniendo en cuenta el siguiente criterio [53]:

$$\frac{2\pi\omega_{Tb}}{3} \leq \omega_C \leq 2\pi\omega_{Tb} \tag{89}$$

donde ω_{Tb} es el ancho de banda del sistema en red cerrada, el cual se estableció en la ecuación (61) igual a *1rad/s*. Se obtiene entonces:

$$2.0943 \leq \omega_C \leq 6.2831 \tag{90}$$

Expresando en Hertz

$$0.333 \leq F_C \leq 1 \tag{91}$$

Se seleccionó entonces una frecuencia de corte para el filtro de $F_C = 0.5\,Hz$. Para implementar la ecuación (88) en EJS, esta se representó de manera discreta, de la siguiente forma:

$$F_o(k) = F_o(k-1) + x_1(k)T_{SV}$$

$$x_1(k) = x_1(k-1) + a_f\left(F_i(k) - \frac{x_1(k)}{2\pi F_C} - F_o(k)\right) \tag{92}$$

$$Con \quad a_f = 3T_{SV}(2\pi F_C)^2$$

Calibración:
A la salida del filtro se tiene una señal de voltaje que se debe escalar para representar la temperatura medida en una escala porcentual del 0 al 100% (señal PV). Además por efecto del offset presente en el bloque acondicionamiento es necesario calibrar el transmisor de temperatura. Por tanto se implementó la siguiente ecuación lineal que relaciona la variación porcentual con el voltaje generado por el sensor:

$$P_V(k) = SPAN * F_o(k) + ZERO, \tag{93}$$

donde con los parámetros *SPAN* y *ZERO* se ajusta la inclinación y punto de partida de la recta que representa esta ecuación. Para determinar los valores de ajuste de estos parámetros, se calculó el voltaje de la señal acondicionada (ver ecuación (86)) para un rango de temperatura de 0 a 100℃ y se tabularon los valores obtenidos en la Tabla 3-20 relacionándolos con su respectivo valor porcentual.

Tabla 3-20 Valor porcentual en función del voltaje del sensor

E [V]	0.468	0.995	1.522	2.049	2.576	3.103	3.630	4.157	4.684	5.211	5.738
PV [%]	0	10	20	30	40	50	60	70	80	90	100

Se realizó una regresión lineal por el método de mínimos cuadrados con el comando "*polyfit*" de Matlab, obteniendo los valores de los parámetros de la ecuación (93):

$$SPAN = 18.975 \qquad ZERO = -8.884$$

En el diseño del transmisor de temperatura se asumió un rango y valor por defecto para los parámetros, como lo especifica la Tabla 3-21.

Tabla 3-21 Rangos y valores por defecto de los parámetros del transmisor de temperatura

Parámetro	**Descripción**	**Und.**	**Valor**		
			Min.	**Max.**	**Def.**
SPAN	Constante de calibración de la multiplicación	-	10	20	18.975
ZERO	Constante de calibración del cero	-	-10	10	-8.884
K_S	Ganancia de amplificación	-	1	1000	1000
F_C	Frecuencia de corte del filtro	Hertz	0.1	1	0.5

3.3.2.8 Panel de Fallas

Este panel implementa la interfaz para la generación de 10 fallas típicas en el lazo de control, como lo muestra la Figura 3-66.

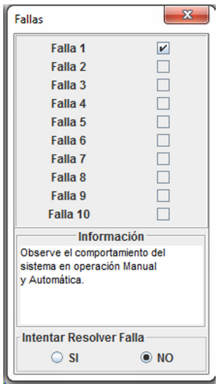


Figura 3-66 Interfaz de selección de las Fallas

Cuenta con casillas de verificación para seleccionar entre diez fallas, un área de mensajes donde se muestra información de ayuda, que indica como observar el efecto de la falla en el comportamiento del sistema y un área de selección para intentar resolver la falla.

3.3.2.8.1 Tipos de Fallas

Las fallas básicamente intentan simular comportamientos de desgaste y de desajuste de parámetros en los elementos: TIC-101, CV-101 y TIT-101 del lazo de control (ver Figura 3-46). Para simular el efecto de la falla sobre estos elementos, se modifica el valor de algunos de sus parámetros con el fin de generar un comportamiento inadecuado del elemento y por tanto del sistema de control; el cambio del valor de los parámetros además permite una posterior reparación de la falla. Las diez fallas se agrupan según el elemento que afectan de la siguiente manera:

Fallas en el Transmisor de Temperatura

Consisten de tres tipos de fallas que afectan los bloques que conforman la estructura del transmisor (ver Figura 3-65):

1. Pérdida de medida: Para simular este comportamiento se modifica la ganancia *Ks* en el bloque de Acondicionamiento, asignándole un valor de cero, simulando la apertura del lazo de realimentación.

2. Amplificación de ruido: Para simular este comportamiento se modifica el valor de la frecuencia de corte F_C en el bloque Filtro, asignándole un valor de 1.35 Hz, amplificando el efecto del ruido en la variable del proceso PV.
   La selección de este valor se debe a que con 1.35Hz se tiene una constante de tiempo equivalente para la FDT del filtro paso-bajo de la ecuación (88), igual a $\tau_F = 0.078\,s$, valor que está muy cerca del periodo de muestreo T_{SV} con que se simula el modelo del filtro y por tanto para valores mayores se obtendrá un comportamiento inadecuado del sistema de control.

3. Calibración inadecuada: Para simular este comportamiento se modifica el valor de los parámetros SPAN y ZERO en el bloque Calibración, asignándoles un valor de 20 y 10 respectivamente, cambiando la inclinación y punto de partida de la recta que representa la ecuación (93) modificando la calibración que se tenía por defecto.

Fallas en la Válvula de Control

Consisten de tres tipos de fallas que afectan el comportamiento dinámico y estático del modelo de la válvula (ver Figura 3-63):

4. Envejecimiento del resorte: Para simular este comportamiento se modifica la ganancia *Kf* asignándole un valor de 2, siendo la ganancia estática del modelo de la válvula igual a 0.5, consiguiendo que la posición de salida de la válvula X_0 sea la mitad de la referencia deseada X_{REF}. De esta manera se simula el comportamiento del envejecimiento del resorte de la válvula ya que se modifica su constante de elasticidad al aumentar su rigidez, obteniendo un apertura total del 50%.

5. Histéresis: Para simular este comportamiento se modifica el valor de los límites mínimos B_{Mm} y máximos B_{Mx} de la banda muerta, asignándoles un valor porcentual de -10 y 10 respectivamente, obteniendo una zona muerta del 20% en la cual la válvula no responderá a las órdenes de la señal de control, simulando el comportamiento de una fuerte histéresis en la válvula.

6. Desgaste del vástago: Para simular este comportamiento se aumenta el valor de la constante de tiempo τ asignándole un valor de 10 segundos, obteniendo una dinámica muy lenta del modelo de la válvula. De esta manera se simula el desgaste del vástago por el aumento de la fricción viscosa, lo que disminuye la rapidez de apertura de la válvula, afectando la velocidad de respuesta del sistema.

Fallas en el Controlador

Consisten de cuatro tipos de fallas que afectan el desempeño del controlador PID (ver Figura 3-56), por ajuste inadecuado de sus parámetros:

7. Sintonía muy suave: Para simular este comportamiento se modifica el valor de los

parámetros asignándoles los valores $K_p = 0.1$, $T_i = 1$ y $T_d = 0.1$, obteniendo una respuesta lenta de la señal de control y por tanto una respuesta del sistema de larga duración y sobre-amortiguada.

8. Excesiva acción proporcional: Para simular este comportamiento se modifica el valor de los parámetros asignándoles los valores $K_p = 1.8$, $T_i = 1.5$ y $T_d = 0.5$, obteniendo un gran esfuerzo de control y por tanto una respuesta del sistema con excesivo sobre-paso y de comportamiento oscilatorio.

9. Excesiva acción derivativa: Para simular este comportamiento se modifica el valor de los parámetros asignándoles los valores $K_p = 1$, $T_i = 1.5$ y $T_d = 2$, obteniendo un gran esfuerzo de control con disminución del amortiguamiento del sistema.

10. Tiempo de muestreo inadecuado: Para simular este comportamiento se aumenta cuatro veces el valor del parámetro T_s asignándole un valor de 2.4 segundos, valor inadecuado ya que esta por fuera del rango establecido en la ecuación (61), obteniendo una señal de control menos suave y poco definida, afectando el comportamiento y la velocidad de respuesta del sistema.

3.3.2.8.2 Diagnóstico y reparación

Con el fin de generar habilidades de diagnóstico y reparación, cuando se genera una falla se implementa un manejo especial de los sub-paneles de configuración de los elementos: TIC-101, CV-101 y TIT-101, como lo muestra el algoritmo de la Figura 3-67.

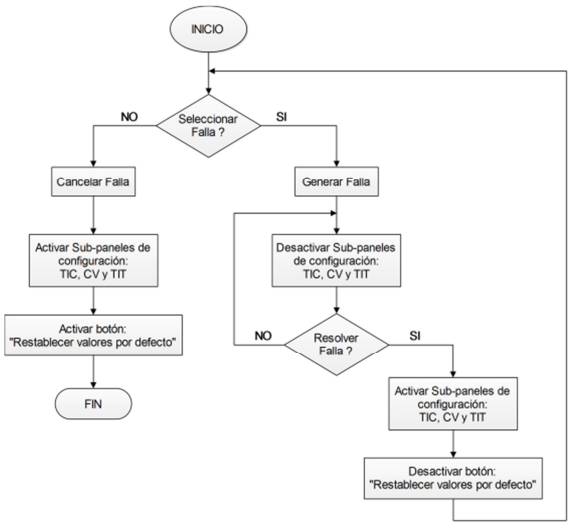


Figura 3-67 Algoritmo que implementa HAPID, para el manejo de los sub-paneles ante una falla

Si se selecciona una falla, inmediatamente se genera y se desactivan los sub-paneles de configuración de los elementos: TIC-101, CV-101 y TIT-101, siendo imposible acceder a los parámetros de ajuste, con el objetivo de que inicialmente se observe el comportamiento del sistema en falla para generar un posible diagnóstico. Posteriormente, si se decide reparar o resolver la falla se activan nuevamente estos sub-paneles y sus parámetros con el fin de realizar las respectivas correcciones o ajustes. Finalmente se selecciona otra falla e inicia de nuevo el ciclo. Para cancelar el efecto de la falla sobre el comportamiento del sistema, simplemente se desactiva la casilla de verificación correspondiente.

3.3.3 Pruebas de funcionamiento

En los sistemas de control, la implementación práctica de la solución del problema se realiza en dos fases; la primera fase consiste en el análisis, diseño e implementación de la estrategia de control, y la segunda fase incluye aspectos de operación, supervisión y mantenimiento [54].

Uno de los aspectos más destacados de la aplicación HAPID, es el de simular efectos de falla en los elementos que conforman el lazo de control, debidos a desgaste o desajuste del instrumento. La posibilidad de generar fallas fomenta el aprendizaje y genera habilidades en aspectos de mantenimiento como: la detección, el diagnóstico y la reparación.

Con el fin de ilustrar el uso y prueba de la aplicación HAPID, a continuación se muestra el diseño e implementación de la solución de un problema de control; y se simula la tarea de mantenimiento del sistema al generar la falla "Perdida de medida". El efecto de las restantes fallas en el comportamiento del sistema y su detección, diagnóstico y reparación se reportaron en el documento: *Guía reparación fallas*, el cual se encuentra disponible en el Anexo B.

3.3.3.1 Problema de control y solución

3.3.3.1.1 Problema

En HAPID, el proceso consiste de un intercambiador de calor cuyo propósito es calentar el fluido de entrada desde una temperatura $T_i(t)$, a una cierta temperatura deseada del fluido a la salida, $T_o(t)$. Se debe diseñar un sistema de control que cumpla las siguientes especificaciones de desempeño:

- Error de estado estable e_{ss} nulo
- Respuesta transitoria con sobrepaso máximo del 25%
- Tiempo de estabilización t_s menor o igual a 10 segundos (criterio del 2%)

3.3.3.1.2 Análisis

Para cumplir con las especificaciones deseadas inicialmente se debe determinar el modelo del proceso. El procedimiento consiste en acceder al panel "Control" y operar el sistema en modo "Manual"; ajustar la variable MV en el 50% y cuando se estabilice la respuesta del sistema realizar un incremento del 30%, como lo muestra el comportamiento de la Figura 3-68.

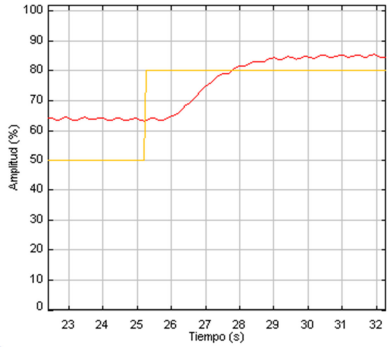


Figura 3-68 Comportamiento del proceso en modo "Manual"

Al analizar la respuesta obtenida, se observa un tiempo muerto de 0.5 segundos, una ganancia estática de 0.7 y una constante de tiempo de 1 segundo; aproximando el comportamiento del proceso a una FDT de primer orden con tiempo muerto, se obtiene:

$$G(s) = \frac{0.7e^{-0.5s}}{s+1}$$

3.3.3.1.3 Diseño e implementación

Después de definir el modelo del proceso, se diseña la estrategia de control PID según las especificaciones de desempeño establecidas. El controlador se sintonizo usando el método de diseño por asignación de polos [53], obteniendo el valor de los parámetros $K_p = 1$, $T_i = 1.5\ s$ y $T_d = 0.5\ s$. Estos parámetros se ajustaron en el panel "Configuración PID", como lo muestra la Figura 3-69.

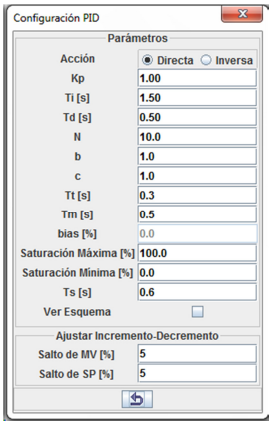


Figura 3-69 Parámetros de ajuste del controlador PID

Finalmente se verifica el comportamiento del sistema controlado. El procedimiento consiste en realizar la transferencia al modo "Automático"; ajustar la variable SP en el 65% y cuando se estabilice la respuesta del sistema realizar un incremento del 20%, como lo muestra el comportamiento de la Figura 3-70.

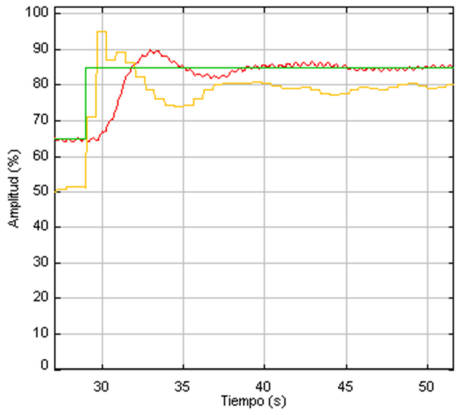


Figura 3-70 Comportamiento del proceso en modo "Automático"

Al analizar el comportamiento obtenido, se observa una respuesta transitoria amortiguada con un sobrepaso máximo del 25%, con t_s de 9 segundos y e_{ss} nulo; verificando de esta manera que se cumple con las especificaciones de desempeño establecidas.

3.3.3.2 Mantenimiento

Para simular una tarea de mantenimiento, a continuación se simula el efecto de la falla "Perdida de medida"; y se muestra el procedimiento de detección, diagnóstico y reparación.

Cuando se genera esta falla, HAPID muestra el siguiente mensaje de información (ver sección 3.3.2.8): "*Observe el comportamiento del sistema en operación automática*". Por tanto, en el panel "Control" se selecciona el modo de operación "Automático" y se observa el efecto de esta falla en el comportamiento del sistema, como lo muestra la Figura 3-71.

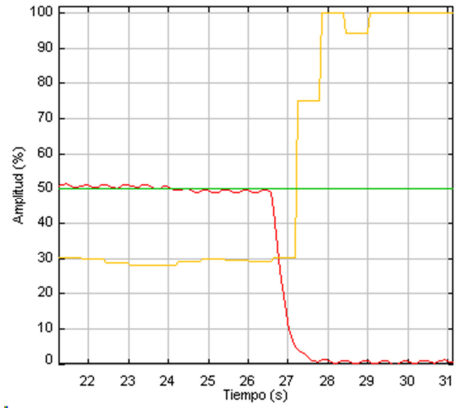


Figura 3-71 Efecto de la falla "Perdida de Medida" en el comportamiento del sistema

3.3.3.2.1 Detección y diagnóstico

En este caso el sistema presenta una condición inicial de comportamiento normal al permanecer la variable PV en el valor de SP establecido del 50%. Aproximadamente a los 26 segundos de simulación se genera la falla, y se observa una desviación de la variable PV del valor de referencia establecido del 50% al 0%; y de la variable MV del 30% hasta alcanzar el valor máximo de saturación del 100%.

Este comportamiento inadecuado del sistema se debe a un mal funcionamiento del transmisor de temperatura, ya que se pierde la medida de la variable del proceso PV al tomar el valor de 0%, lo cual ocasiona una señal de error máxima en el controlador que genera un gran esfuerzo de control hasta saturar la variable MV en el 100%.

3.3.3.2.2 Reparación

El procedimiento para reparar la falla es el siguiente:

1. En el panel de "Fallas" se selecciona la opción SI del campo "Intentar resolver falla", como lo muestra la Figura 3-72.

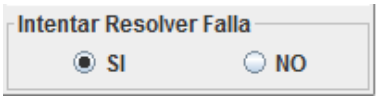


Figura 3-72 Campo para resolver la falla del panel Fallas

2. Se accede al panel de configuración del transmisor de temperatura, como lo muestra la Figura 3-73.

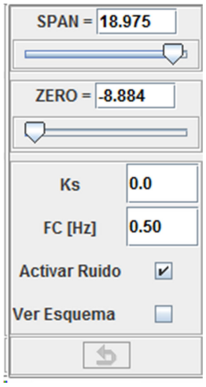


Figura 3-73 Parámetros de configuración del transmisor TIT-101

Al analizar los parámetros se observa que el campo K_s correspondiente a la ganancia de amplificación del sensor tiene por valor cero, lo cual da una indicación de la inoperancia del transmisor.

3. Se corrige el valor del parámetro K_s al valor que tenía por defecto, se ajusta en 1000.

4. Finalmente se verifica el comportamiento del sistema, como lo muestra la Figura 3-74, observando que cuando se corrige la falla (aproximadamente a los 40 segundos de la simulación), el sistema retorna a su estado de comportamiento normal.

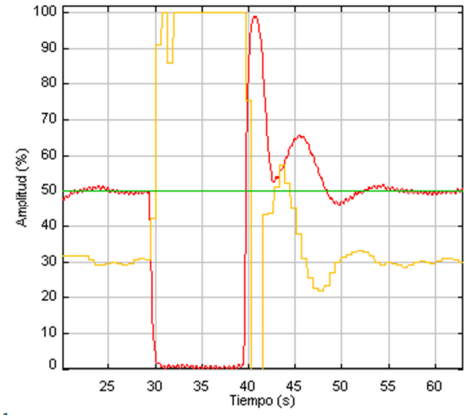


Figura 3-74 Comportamiento del sistema después de corregir la "Perdida de medida"

Comentario: Las pruebas realizadas basadas en el diseño, implementación y mantenimiento de la solución de un problema de control fueron satisfactorias; ya que permitieron obtener el modelo que representa el proceso, observar el correcto funcionamiento de la solución de control diseñada y el efecto de la falla "Perdida de medida" en el comportamiento del sistema.

3.4 Resumen

En este capítulo se presentó el diseño y desarrollo de un sitio web con un esquema de navegación que permite un fácil acceso y una correcta ejecución de las aplicaciones informáticas.

El diseño y desarrollo de 15 Lúdicas Animadas, dirigidas a facilitar el aprendizaje de los conceptos más importantes de la teoría de los sistemas de control en temáticas como: realimentación, modelado matemático, análisis temporal y frecuencial, estabilidad, compensación, reconstrucción de señales, estructuras de control, controlabilidad y observabilidad. Las Lúdicas Animadas se organizaron en páginas web con una estructura que consiste en una estrategia didáctica para facilitar el aprendizaje del concepto, a través de la comprensión del funcionamiento de la animación, del desarrollo de unas actividades y del análisis de unas preguntas de discusión. Para uso de las Lúdicas Animadas por un docente, se elaboró el documento: *Guía profesor animaciones*, el cual se encuentra disponible en el Anexo A; donde se presentan los objetivos y resultados de aprendizaje esperados del uso de las Lúdicas Animadas como recurso de apoyo para el aprendizaje de conceptos en la educación en control. Las pruebas de funcionamiento realizadas basadas en el análisis de la Lúdica Animada "Válvula y Tanque" fueron satisfactorias; ya que permitieron conocer el concepto de modelado matemático de un sistema.

El diseño y desarrollo de la Herramienta de Análisis del Controlador PID HAPID como soporte para el análisis de un controlador PID práctico y aprendizaje de las diversas funciones de operación y de detección de fallas en lazos de control. HAPID recrea un ambiente de *Supervisory Control And Data Acquisition SCADA* para un intercambiador de calor y presenta funcionalidades como: control de acceso con tres niveles de jerarquía (Ingeniero, Supervisor y Operario), gestión de permisos, generación y reconocimiento de alarmas, ajuste de parámetros del PID, modo de operación (Manual - Automático), configuración de la válvula de control y del transmisor de temperatura, registro de eventos, elaboración de tres tipos de reportes (configuración, datos y eventos), y generación de diez fallas típicas que afectan los elementos (controlador, actuador y sensor) del lazo de control. Cuenta con aspectos prácticos de implementación como: acción de control PID en estructura ISA, filtrado de la derivada, antiembalamiento de la acción integral, transferencia suave en el cambio del modo de operación, modelos matemáticos que simulan comportamientos como histéresis, banda muerta y características del caudal en la válvula de control; y ruido, filtrado y calibración en el transmisor de temperatura. La implementación y generación de las fallas fomenta el aprendizaje y genera habilidades en aspectos de mantenimiento como: la detección, el diagnóstico y la reparación; con el fin de ilustrar el diagnóstico y procedimiento de reparación de las 10 fallas, se elaboró el documento: *Guía reparación fallas*, el cual se encuentra disponible en el Anexo B. Para el correcto manejo y operación de la aplicación, se elaboró el documento: *Guía HAPID usuario*, el cual se encuentra disponible en el Anexo C. Las pruebas de funcionamiento realizadas basadas en el diseño, implementación y mantenimiento de la solución de un problema de control fueron satisfactorias; ya que permitieron observar el correcto funcionamiento de la solución de control diseñada y el efecto de la falla "Perdida de medida" en el comportamiento del sistema.

4 DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE LA HERRAMIENTA DE ANÁLISIS MATEMÁTICO

En el diseño e implementación de un sistema de control es necesario obtener el modelo de la planta, analizar su dinámica y diseñar la estrategia de control. Una vez se tiene el modelo matemático que representa el proceso, se realiza el procedimiento de análisis donde se observa y evalúa el comportamiento dinámico del sistema.

Los análisis más comunes son [62]:

- Análisis en el dominio del tiempo: consiste en evaluar la respuesta temporal del sistema a señales de entrada particulares, como: el escalón, el impulso y la rampa.
- Análisis del lugar geométrico de las raíces: consiste de un plano complejo de abscisa Real y ordenada Imaginaria donde se ubican los polos y ceros del sistema en red abierta, por medio del cual se analiza la estabilidad del sistema a partir de las trayectorias que describen los polos en red cerrada.
- Análisis en el dominio de la frecuencia: Es otra forma de analizar la estabilidad del sistema por medio de trazas de magnitud y fase de Bode o graficas polares como el diagrama de Nyquist.

La herramienta de cálculo matemático consiste de la Herramienta de Análisis de Sistemas de Control HASC, la cual se desarrolló con las herramientas EJS y Scilab; con EJS se implementó la interfaz gráfica de usuario y se usó Scilab como motor de cálculo matemático. La aplicación HASC diseñada permite realizar el análisis de los sistemas de control [62] de manera remota, como el temporal, frecuencial y el lugar de las raíces de un sistema continuo o discreto representado por su función de transferencia FDT o por su modelo en el espacio de estados EE; además realiza transformaciones entre planos (del plano continuo al discreto, o del plano discreto al plano ω) y entre representaciones (de FDT a EE y viceversa), mostrando los coeficientes calculados en una ventana emergente.

En este capítulo inicialmente se describe la arquitectura software de HASC, la cual se basa en dos aplicaciones, HASC-Cliente y HASC-Servidor para realizar un determinado análisis y/o transformación de manera remota; además se presenta la interfaz *Javasci* que permite realizar la conexión entre una aplicación Java con Scilab y de esta manera llevar a cabo el procesamiento matemático. Posteriormente se establecen los requerimientos generales de funcionamiento y de aspecto gráfico de las dos aplicaciones. Seguidamente se muestra el diseño y desarrollo de cada aplicación, la implementación del protocolo TCP/IP para la comunicación entre el servidor y los clientes a partir del intercambio de *tramas de datos*, y se presentan los algoritmos de ejecución concebidos para procesar las peticiones y los resultados. Finalmente se muestran pruebas de funcionamiento realizadas a las dos aplicaciones, en HASC-Servidor se prueba la conexión y en HASC-Cliente se prueba su uso al realizar el análisis temporal de un sistema continuo. Las pruebas de los restantes análisis y transformaciones se reportaron en el documento: *Guía HASC usuario*, el cual se encuentra disponible en el Anexo D.

El documento *Guía HASC usuario*, además ilustra el correcto manejo y operación de HASC. En esta guía se presentan los requerimientos de ejecución, especificaciones y aspectos generales de la aplicación; además por medio de ejemplos se muestra el procedimiento adecuado para realizar un determinado análisis y/o transformación.

4.1 Arquitectura Software de HASC

Para realizar el análisis de un sistema de control de manera remota a través de Internet, HASC implementa una arquitectura Cliente – Servidor con el protocolo de comunicación TCP/IP, como lo muestra la Figura 4-1.

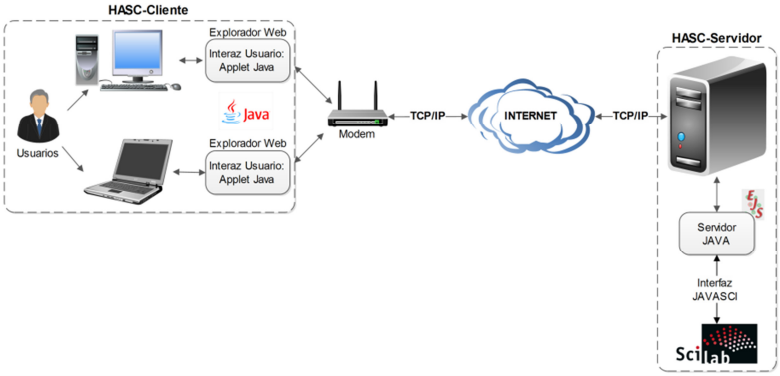


Figura 4-1 Arquitectura software de HASC

La arquitectura cuenta con dos aplicaciones, una cliente HASC-Cliente y una servidor HASC-Servidor, que utilizan enlaces de tipo TCP/IP para la comunicación y transferencia de datos. En este caso los usuarios utilizan HASC-Cliente como interfaz gráfica para el envío de solicitudes y visualización de los diferentes análisis y/o transformaciones; y HASC-Servidor para procesar estas solicitudes en Scilab, por medio de una interfaz denominada *Javasci* y retornar los resultados.

4.1.1 Interfaz JAVASCI

Desde la versión 4.0, una interfaz Java fue incorporada en Scilab. Esta interfaz permite llamar el motor computacional de Scilab desde programas realizados en Java. La interfaz se denomina *Javasci V2* [63], la cual proporciona algunos objetos y métodos para llevar a cabo la conexión con Scilab. En la interfaz *Javasci V2* las clases se encuentran divididas en dos librerías (paquetes):

- *org.scilab.modules.javasci*
  Este paquete proporciona la manera de utilizar y llamar el motor Scilab desde una aplicación Java. Contiene las clases: *Call_Scilab, Scilab y ScilabVariablesJavasci. Scilab* constituye la clase principal ya que proporciona los métodos necesarios para realizar tareas de conexión, envío/recepción de datos, gestión de errores y ejecución de código (script) de Scilab. A continuación se describen los principales métodos que contiene esta clase:

  - *open ():* Se abre y establece una conexión con el motor de Scilab.
  - *close ():* Se cierra la conexión con Scilab.
  - *exec ():* Envía y ejecuta uno o varios comandos y scripts .sce o .sci, de Scilab.
  - *get ():* Obtiene el valor de una variable almacenada en Scilab.

- *org.scilab.modules.types*
  Este paquete proporciona una representación de la mayoría de los tipos de datos de Scilab. Se utiliza la clase *ScilabDouble* para representar el tipo de dato Double de Scilab y el método *getRealPart()* para obtener la parte real del dato.

4.2 Requerimientos

Como se mencionó anteriormente HASC implementa una arquitectura Cliente – Servidor y por tanto requiere el desarrollo de dos aplicaciones: una cliente HASC-Cliente y una servidor HASC-Servidor. A continuación se presentan los requerimientos funcionales y no funcionales, y las interfaces graficas de usuario de estas dos aplicaciones.

4.2.1 Aplicación HASC-Cliente

A continuación se establecen los requerimientos funcionales y no funcionales, se definen los rangos, unidades y restricciones permitidas, y se presenta el esquema de la interfaz gráfica de usuario de la aplicación cliente.

4.2.1.1 Requerimientos funcionales

- Debe solicitar conexión de tipo TCP con la aplicación HASC-Servidor.

- Permitir el ingreso de modelos continuos o discretos representados por:
  - Función de transferencia FDT.
  - Espacio de estados EE de sistemas mono-variables y multi-variables.

- Permitir realizar tres tipos de análisis:
  - Temporal, a través de la respuesta al escalón, impulso o rampa.
  - Lugar Geométrico De las Raíces LGDR.
  - Frecuencial, a través de diagramas de Bode o Nyquist.

- Los análisis solicitados deben ejecutarse remotamente en el servidor de aplicación que contiene Scilab.

- Se debe visualizar la respuesta gráfica del análisis solicitado.

- Permitir el ajuste de parámetros de entrada como: amplitud, duración, pendiente, ganancia máxima, frecuencia inicial y final y tiempo de muestreo.

- Permitir realizar dos tipos de transformaciones:
  - Entre planos:
    - Del plano continuo al discreto, por medio de los métodos ZOH (retenedor de orden cero), Mapeo polo-cero y Bilineal (S-Z).
    - Del plano discreto al plano ω, por medio del método Bilineal (Z-ω).
  - Entre representaciones:
    - De FDT a EE.
    - De EE a FDT.

- Las transformaciones solicitadas deben ejecutarse remotamente en Scilab.

- Se deben visualizar los coeficientes de la FDT o del modelo en EE obtenidos como resultado de una transformación entre planos o representaciones.

- Permitir copiar los coeficientes de la representación FDT discreta de salida (obtenidos como resultado de una discretización continua a discreta) en los coeficientes de la representación FDT discreta de entrada.

- Permitir reiniciar a condiciones iniciales y valores por defecto.

4.2.1.2 Requerimientos no funcionales

- Se debe utilizar el software EJS, versión 5.0 como herramienta de desarrollo.
- Se debe utilizar el entorno de ejecución Java (JRE), versión 1.7 o superior.
- Se debe utilizar el protocolo de comunicación TCP/IP.

4.2.1.3 Rangos, unidades y restricciones

Para un correcto manejo y operación de la aplicación, se plantean las siguientes restricciones:

- En la representación en función de transferencia se tienen:
  - Debe ser propia o estrictamente propia, es decir, el grado del polinomio denominador debe ser mayor o igual que el del polinomio numerador.
  - El orden del sistema o el grado del polinomio denominador debe ser menor o igual a cuatro.

- En la representación en modelo de espacio de estados se tienen:
  - Se debe permitir el ingreso de sistemas mono-variables de hasta cuatro estados, una entrada y una salida.
  - Se debe permitir el ingreso de sistemas multi-variables de hasta dos estados, dos entradas y dos salidas.

- En cuanto a los parámetros de entrada, en la Tabla 4-1 se muestran las unidades, rangos y valores por defecto permitidos para cada análisis.

Tabla 4-1 Valores permitidos de los parámetros de entrada en HASC-Cliente

Análisis	Parámetro	Unidades	Rango		Valor por defecto
			Min.	Max.	
Temporal, LGDR, Frecuencial	Tiempo de muestreo	Segundos (seg)	0.01	1	0.01
Temporal	Amplitud	Unidades (Und)	-10	10	1
	Duración	Segundos (seg)	1	100	10
	Pendiente	Und/seg	-10	10	1
LGDR	Ganancia	--	0	1000	1000
Frecuencial	Frecuencia Inicial	Hertz (Hz)	0.001	1	0.001
	Frecuencia Final	Hertz (Hz)	1	1000	1000

4.2.1.4 Interfaz gráfica de usuario

Teniendo en cuenta las funcionalidades establecidas se determinó un panel principal y un panel emergente, conformados por las siguientes áreas:

- Área 1: Botones para seleccionar el tipo de modelo, continuo o discreto.
- Área 2: Botones para seleccionar el tipo de transformación entre planos.
- Área 3: Campos para el Ingreso de coeficientes según el tipo de representación.
- Área 4: Botón para seleccionar el tipo de transformación entre representaciones.
- Área 5: Botones para seleccionar el tipo de análisis.

- Área 6: Campos para el ajuste de parámetros.
- Área 7: Botones para ejecución y reinicio.
- Área 8: Campos para el ajuste de ejes de la gráfica.
- Área 9: Respuesta gráfica del análisis.
- Área 10: Campos para el registro de los coeficientes calculados.

Por tanto, la organización y presentación gráfica general que debe tener la aplicación HASC-Cliente, se muestra en la Figura 4-2.

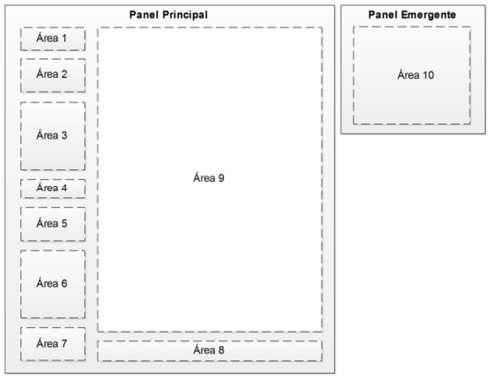


Figura 4-2 Esquema de la interfaz gráfica de usuario de la aplicación HASC-Cliente

4.2.2 Aplicación HASC-Servidor

A continuación se presentan los requerimientos funcionales y no funcionales, y se presenta el esquema de la interfaz gráfica de usuario de la aplicación servidor.

4.2.2.1 Requerimientos funcionales

- Debe estar a la espera por solicitudes de conexión de los clientes.
- La conexión y la transferencia de datos se debe realizar por medio de *Sockets* TCP en Java.
- Por cada usuario que solicite conexión, un nuevo hilo de ejecución se creará.
- Debe establecer conexión con Scilab.
- Debe procesar peticiones de los diferentes análisis.
- Debe procesar peticiones de las diferentes transformaciones.
- Debe procesar datos en Scilab y obtener resultados.
- Debe retornar los resultados obtenidos al cliente.

4.2.2.2 Requerimientos no funcionales

- Se utilizará el software EJS versión 5.0, como herramienta de desarrollo.
- Se debe utilizar el entorno de ejecución Java (JRE), versión 1.7 o superior.
- Se debe utilizar el protocolo de comunicación TCP/IP.
- Se utilizará el software Scilab versión 5.3.3, como herramienta de cálculo y procesamiento matemático.
- Se utilizará la interfaz *Javasci v2* para realizar la conexión entre Java y Scilab.
- Se requiere tener 300Mbytes en memoria RAM debido a la sobrecarga que representa Scilab.

4.2.2.3 Interfaz gráfica de usuario

Con la finalidad de iniciar la ejecución del servidor y verificar el estado de las conexiones; se determino la organización y presentación gráfica general que debe tener la aplicación HASC-Servidor, como lo muestra la Figura 4-3.

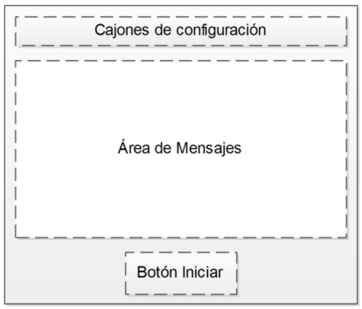


Figura 4-3 Esquema de la interfaz gráfica de usuario de la aplicación HASC-Servidor

4.3 Diseño y Desarrollo

La aplicación HASC diseñada se basa en una arquitectura "Cliente – Servidor" que establece enlaces de comunicación TCP/IP por medio de *Sockets* en *Java*, e implementa un protocolo de comunicación a través del intercambio de *tramas de datos*. El cliente se denomina HASC-Cliente y se encarga de la implementación y envió de solicitudes a través de una *trama de petición*, la cual presenta la estructura de la Figura 4-4.

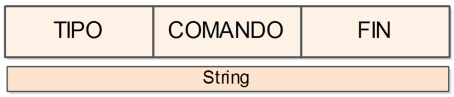


Figura 4-4 Trama de petición que envía HASC-Cliente

Esta trama se conforma por tres campos, donde se almacenan datos de tipo cadena de caracteres o "*String*" con información acerca del análisis y/o transformación solicitada:

- Tipo: Contiene información que especifica el tipo de análisis solicitado.
- Comando: Contiene información que define el tipo de función a ejecutar en Scilab.
- Fin: Contiene información para finalizar la conexión.

El servidor se denomina HASC-Servidor y se encarga del procesamiento de las solicitudes y retorno de resultados a través de una *trama de respuesta* la cual presenta la estructura de la Figura 4-5.

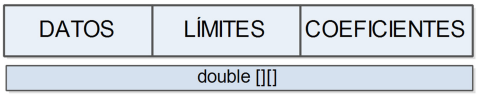


Figura 4-5 Trama de respuesta que envía HASC-Servidor

Esta trama se conforma por tres campos donde se almacenan arreglos de datos de tipo "*double*" con valores del análisis y/o transformación procesada:

- Datos: Contiene datos para construir la representación gráfica del análisis.
- Límites: Contiene los valores máximos y mínimos de los datos.
- Coeficientes: Contiene los coeficientes de las transformaciones.

4.3.1 Aplicación HASC-Cliente

La aplicación cliente implementa la interfaz gráfica de usuario, la cual se desarrolló en EJS y se encarga de la implementación y envió de la solicitud y visualización de resultados, como lo muestra la Figura 4-6.

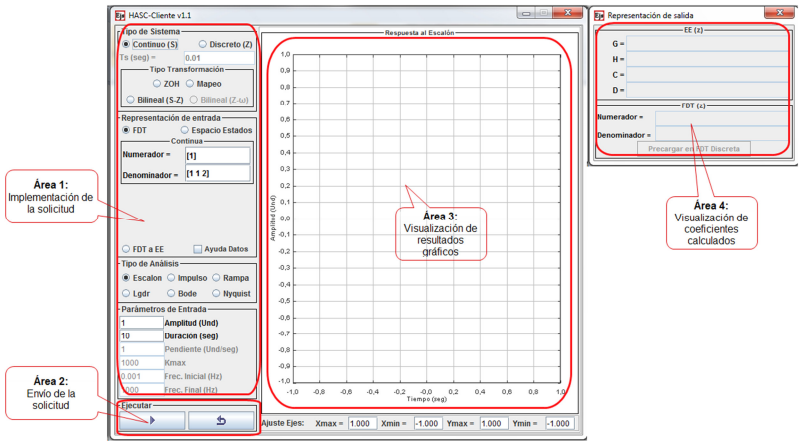


Figura 4-6 Interfaz de usuario de la aplicación HASC-Cliente

Para realizar un determinado análisis o transformación la interfaz cuenta con cuatro áreas bien diferencias:

Área 1: Donde por medio de opciones de selección y campos numéricos se implementa la solicitud o *trama de petición*, la cual se conforma por el tipo de análisis y/o transformación deseada. Esta área cuenta con las siguientes características para construir la solicitud:

- Selección del tipo de sistema continuo S o discreto Z.
- Selección del tipo de Transformación entre planos:
  - S a Z, por medio de los métodos ZOH (retenedor de orden cero), Mapeo polo-cero o Bilineal (S-Z).
  - Z a ω, por medio del método Bilineal (Z- ω).
- Selección del tipo de representación del sistema:
  - Función de transferencia FDT o modelo en el espacio de estados EE.
- Campos numéricos para el ingreso de coeficientes de la FDT o del modelo en EE.
- Selección del tipo de Transformación entre representaciones:
  - FDT a EE.
  - EE a FDT.
- Selección del tipo de análisis:
  - Temporal (respuesta al escalón, impulso y rampa).
  - Lugar Geométrico De las Raíces LGDR.
  - Frecuencial (Diagramas de Bode y Nyquist).
- Campos numéricos para el ingreso de parámetros de entrada como:
  - Amplitud, duración y pendiente.
  - Ganancia máxima K_{max}.

- Frecuencia inicial y final.
- Tiempo de muestreo.

Área 2: Después de implementar la trama de petición, esta se envía por medio del botón ▶ a la aplicación HASC-Servidor para el respectivo procesamiento. Con el botón ↺ se restablece a condiciones iniciales y valores por defecto la interfaz.

Área 3: Donde se visualiza el resultado gráfico del análisis solicitado.

Área 4: Donde se visualizan los coeficientes calculados según la transformación solicitada. Cuenta con el botón "Precargar en FDT Discreta" que copia los coeficientes calculados en la representación FDT de entrada.

4.3.1.1 Algoritmo de ejecución

El funcionamiento de la aplicación HASC-Cliente se basa en el algoritmo de ejecución que se muestra en la Figura 4-7.

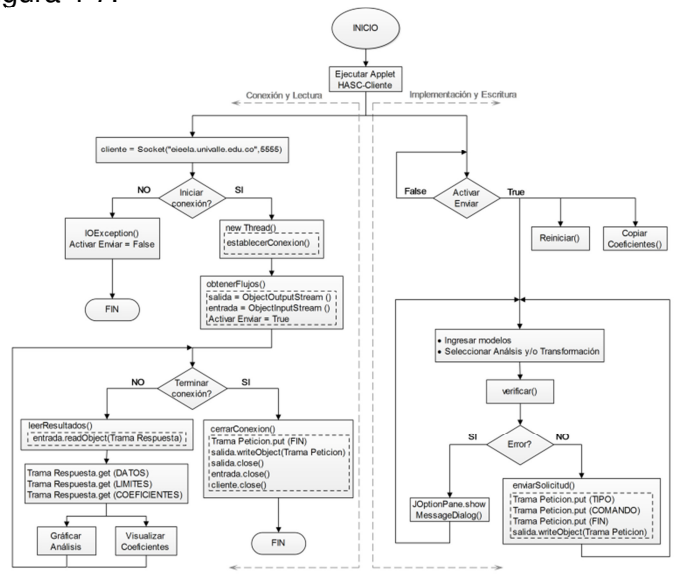


Figura 4-7 Algoritmo de ejecución de la aplicación HASC-Cliente

HASC-Cliente ejecuta dos procesos de manera independiente por medio del uso de hilos (*threads*) en Java. El primer proceso se ejecuta en el hilo principal de EJS y se encarga de la *Implementación y Escritura* (envío) de las solicitudes; el segundo proceso se ejecuta en un hilo auxiliar y se encarga del establecimiento de la *Conexión y Lectura* (recepción) de las solicitudes procesadas. A continuación se explican estos dos procesos.

4.3.1.1.1 Conexión y lectura

Al ejecutar el cliente, inicialmente HASC-Cliente realiza una petición de conexión al servidor de aplicación con URL "*eieela.univalle.edu.co*" y puerto "*5555*", si la conexión es aceptada, un *socket* se crea de forma satisfactoria para usarse en la comunicación con el servidor. Si no se establece conexión se produce una excepción de tipo *IOException* (indicación de error de entrada/salida), desactivando la interacción con los botones de selección y campos numéricos del área 1 y los botones de acción del área 2 de la interfaz de usuario (ver Figura 4-6), finalizando la ejecución del algoritmo.

Posteriormente, si la conexión es aceptada se crea un hilo que contiene el método *establecerConexion()* el cual ejecuta los métodos: *obtenerFlujos(), leerResultados() y cerrarConexion(),* los cuales se explican a continuación.

1. obtenerFlujos():

En este método se crea y asocian flujos intermedios a los flujos de entrada (InputStream) y de salida (OutputStream) asignados al socket. Además se activan las opciones de selección, los campos numéricos y los botones de acción de la interfaz de usuario.

2. leerResultados():

Mientras no se termine con la conexión, en este método se ejecuta continuamente la lectura del flujo de entrada, y la ejecución del hilo auxiliar permanece a la espera por la *trama de respuesta* enviada por el servidor. Cuando se recibe, esta trama se procesa como lo muestra el diagrama de flujo de la Figura 4-8.

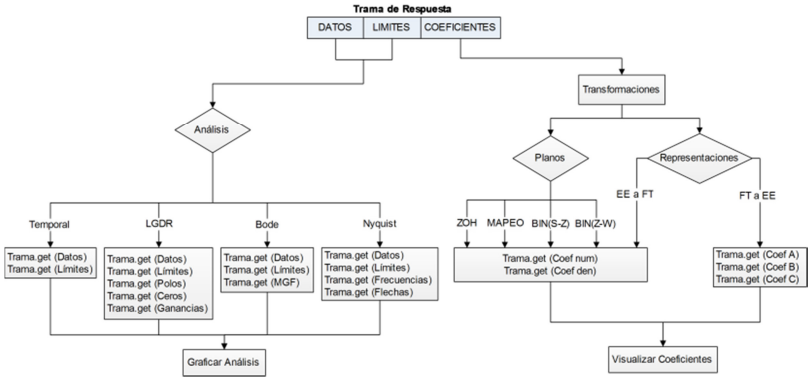


Figura 4-8 Procesamiento de la trama de respuesta en HASC-Cliente

Con los campos DATOS y LÍMITES se obtiene la respuesta gráfica de cada análisis y con el campo COEFICIENTES se obtienen los coeficientes de las transformaciones. A continuación se describe como se construye la respuesta gráfica de cada análisis y como se visualizan los coeficientes.

Graficar análisis:

Cada análisis obtiene la información necesaria de la trama y la almacena en arreglos de datos de tipo "*double[][]*", por tanto dependiendo del análisis solicitado se requieren de los siguientes datos para construir su representación gráfica:

- El análisis Temporal representa la respuesta temporal de un sistema en un gráfico de la Amplitud versus el Tiempo. Según especificación establecida en la sección 4.2.1.3 se permite obtener el comportamiento temporal de sistemas mono-variables y multi-variables de máximo dos entradas y dos salidas. Por tanto este análisis presenta como máximo cuatro trazas de salida y para construirlo se requiere de los siguientes datos:
  - Datos: Arreglo de n filas y cinco columnas. La primera columna contiene los valores de tiempo; si es mono-variable la segunda columna contiene los valores de salida del sistema, si es multi-variable las columnas 3, 4 y 5 contienen los valores de las restantes salidas del sistema. El número de n filas depende del tamaño del arreglo Datos.
  - Límites: Arreglo de una fila y dos columnas. La primera columna contiene el valor máximo de los datos de salida y la segunda columna el valor mínimo.

- El análisis del LGDR se representa en un plano complejo de abscisa Real y ordenada Imaginaria con gráficos de trayectorias, ubicación de polos y ceros. El orden del sistema determina las trayectorias o ramas del lugar y como se especificó en la sección 4.2.1.3 se permite máximo un sistema de cuarto orden. Por tanto este análisis presenta como máximo cuatro ramas y para construirlo se requiere de los siguientes datos:
  - Datos: Arreglo de n filas y ocho columnas con información de las trazas. Si el sistema es de cuarto orden las primeras cuatro columnas contienen valores de la componente Real y las columnas 5, 6, 7 y 8, valores de la componente Imaginaria de cada rama. El número de n filas depende del tamaño del arreglo Datos.
  - Límites: Arreglo de cuatro filas y una columna. Las primeras dos filas contienen el valor mínimo de los ejes Real e Imaginario y las filas 3 y 4 el valor máximo de los ejes Real e Imaginario.
  - Polos: Arreglo de n filas y dos columnas con información de la ubicación de los polos en el plano. La primera columna contiene el valor de la coordenada Real y la segunda columna el valor de la coordenada Imaginaria. El número de n filas depende del número de polos o del tamaño del arreglo Polos.
  - Ceros: Arreglo de n filas y dos columnas con información de la ubicación de los ceros en el plano. La primera columna contiene el valor de la coordenada Real y la segunda columna el valor de la coordenada Imaginaria. El número de n filas depende del número de ceros o del tamaño del arreglo Ceros.
  - Ganancias: Arreglo de n filas y tres columnas con información de la ubicación en el plano y valor de las ganancias que presenta el análisis en cada punto donde las trazas cortan el eje Real y/o Imaginario. La primera columna contiene el valor de la coordenada Real, la segunda columna el valor de la coordenada Imaginaria y la tercera columna el valor de la ganancia. El número de n filas depende del tamaño del arreglo Ganancias.

- El análisis de Bode consiste de dos gráficos de Magnitud y Fase versus el logaritmo en base 10 de la Frecuencia. Por tanto para construirlo se requiere de los siguientes datos:
  - Datos: Arreglo de n filas y tres columnas. La primera columna contiene los valores de la frecuencia, la segunda columna los valores de la magnitud y la tercera columna valores de la fase. El número de n filas depende del tamaño del arreglo Datos.
  - Límites: Arreglo de una fila y seis columnas. Las columnas 1 y 2 contienen los valores mínimos y máximos de la frecuencia, las columnas 3 y 4 contienen los valores mínimos y máximos de la magnitud, las columnas 5 y 6 contienen los valores mínimos y máximos de la fase.
  - MGF: Arreglo de dos filas y dos columnas con información de los márgenes de ganancia MG y fase MF. La primera fila contiene el valor de la frecuencia de cruce de fase y el valor del MG. La segunda fila contiene el valor de la frecuencia de cruce de ganancia y el valor del MF.

- El análisis de Nyquist se representa en un plano complejo de abscisa Real y ordenada Imaginaria en donde se grafica una traza sobre la cual se recorre la frecuencia como un parámetro, por lo que a cada frecuencia le corresponde un punto de la gráfica. Por tanto para construirlo se requiere de los siguientes datos:
  - Datos: Arreglo de n filas y dos columnas con información de la traza de Nyquist. La primera columna contiene valores de la componente Real y la segunda columna valores de la componente Imaginaria. El número de n filas depende del tamaño del

arreglo Datos.
- Límites: Arreglo de una fila y cuatro columnas. Las primeras dos columnas contienen el valor mínimo de los ejes Real e Imaginario y las columnas 3 y 4 el valor máximo de los ejes Real e Imaginario.
- Frecuencias: Arreglo de n filas y tres columnas con información de la ubicación en el plano y valor de las frecuencias. La primera columna contiene el valor de la coordenada Real, la segunda columna el valor de la coordenada Imaginaria y la tercera columna el valor de la frecuencia. El número de n filas depende del tamaño del arreglo Frecuencias.
- Flechas: Arreglo de n filas y cuatro columnas con información de la ubicación, dirección y tamaño de las flechas que indican el sentido de incremento de la frecuencia. Las columnas 1 y 3 contienen el valor de la posición y sentido, y las columnas 2 y 4 contienen el valor del tamaño de las flechas. El número de n filas depende del tamaño del arreglo Flechas.

Visualizar coeficientes:
Al igual que los análisis, cada transformación obtiene el valor de los coeficientes de la trama y los almacena en arreglos de datos de tipo "*double[][]*", por tanto dependiendo de la transformación se requieren de los siguientes datos para visualizar los coeficientes:

1. Las transformaciones entre planos (ZOH, Mapeo, Bilineal (S-Z) y Bilineal (Z-ω)) y la transformación entre representaciones EE a FT, visualizan los coeficientes por medio de los siguientes datos:
   - Num: Arreglo de n filas y una columna, en donde cada fila contiene el valor del coeficiente de cada posición del polinomio del numerador. El número de n filas depende del tamaño del arreglo Num.
   - Den: Arreglo de n filas y una columna, en donde cada fila contiene el valor del coeficiente de cada posición del polinomio del denominador. El número de n filas depende del tamaño del arreglo Den.

2. La transformación entre representaciones FT a EE, visualiza los coeficientes por medio de los siguientes datos:
   - A: Arreglo de n filas y una columna, en donde cada fila contiene el valor del coeficiente de cada posición de la matriz A. El número de n filas depende del tamaño del arreglo A.
   - B: Arreglo de n filas y una columna, en donde cada fila contiene el valor del coeficiente de cada posición de la matriz B. El número de n filas depende del tamaño del arreglo B.
   - C: Arreglo de n filas y una columna, en donde cada fila contiene el valor del coeficiente de cada posición de la matriz C. El número de n filas depende del tamaño del arreglo C.

3. cerrarConexion():

Cuando el usuario termina la conexión, la lectura del flujo de entrada en el método *leerResultados()* termina su proceso, continuando la ejecución del algoritmo en el método *cerrarConexion()*. En este método se cierra el socket y los flujos de entrada-salida establecidos y se envía en el campo FIN de la *trama de petición* una palabra que notifica la desconexión al servidor, finalizando la ejecución del algoritmo.

4.3.1.1.2 Implementación y escritura

Continuando con la ejecución del algoritmo de la Figura 4-7, después de establecer la conexión con el servidor de aplicación, se activa la interacción con la interfaz gráfica de usuario, siendo posible implementar y enviar la solicitud del respectivo análisis y/o transformación en la *trama de petición* para su respectivo procesamiento en el servidor. Esta trama se construye como lo muestra el diagrama de flujo de la Figura 4-9.

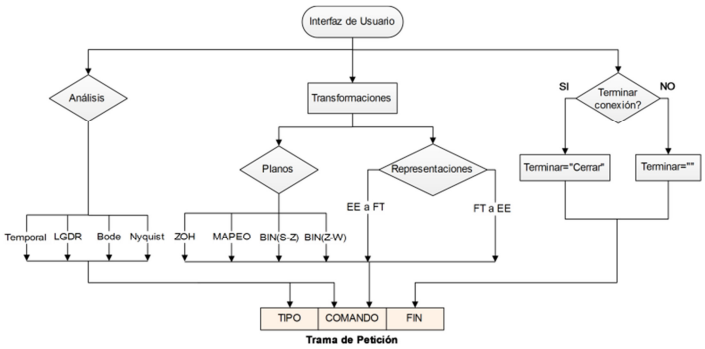


Figura 4-9 Construcción de la trama de petición en HASC-Cliente

Los campos de la trama contienen datos de tipo "*string*". El campo TIPO se construye con el tipo de análisis seleccionado, el campo COMANDO se construye también con el tipo de análisis y además con el tipo de transformación seleccionada, y el campo FIN se construye con un mensaje cuyo contenido depende de la decisión que tome el usuario sobre terminar la conexión, en este caso la palabra "Cerrar" indica terminar la conexión.

En el campo COMANDO se indica el *tipo de función* a ejecutar remotamente en Scilab a través del servidor de aplicación. Esta función presenta la siguiente estructura general:

[argumentos de salida] = nombre de la función (argumentos de entrada)

Donde los *argumentos de entrada* contienen variables que indican el tipo de análisis y/o transformación seleccionada por el usuario y en los *argumentos de salida* se obtienen los resultados de la ejecución de la función cuando esta es procesada en el servidor.

El *nombre de la función* y los *argumentos de entrada y salida* dependen del tipo análisis y/o transformación seleccionada en la interfaz de usuario, de la siguiente manera:

- Para el análisis Temporal la función se denomina *rtatem()*, y tiene la estructura:

[dat,lim,num0,den0,A0,B0,C0]=rtatem(dom,funrep,num,den,Ts,tipo,ana,tf,amp,A,B,C,D,X0,tran)

- Para el análisis del LGDR la función se denomina *lgdr(),* y tiene la estructura:

[lim,zl,pl,datosRI,pvk,num0,den0,A0,B0,C0]=lgdr(dom,funrep,num,den,Ts,tipo,k,A,B,C,D,X0,tran)

- Para el análisis de Bode la función se denomina *bod(),* y tiene la estructura:

[dat,lim,pvGF,num0,den0,A0,B0,C0]=bod(dom,funrep,num,den,Ts,tipo,fmin,fmax,A,B,C,D,X0,tran)

- Para el análisis de Nyquist la función se denomina *nyq()*, y tiene la estructura:

[dat,lim,frq,fle,num0,den0,A0,B0,C0]=nyq(dom,funrep,num,den,Ts,tipo,fmin,fmax,A,B,C,D,X0,tran)

Las variables de los argumentos de entrada y salida de estas cuatro funciones se describen en la Tabla 4-2.

Tabla 4-2 Argumentos de entrada y salida de las funciones *rtatem(), lgdr(), bod() y nyq()*

Argumento	Variable	Descripción
Entrada	*dom*	Dominio o tipo de sistema, continuo “S” o discreto “Z”
	funrep	Tipo de representación del sistema “FDT” o “EE”
	num	Coeficientes del numerador del sistema en FDT
	den	Coeficientes del denominador del sistema en FDT
	Ts	Valor del tiempo de muestreo
	tipo	Transformación entre planos: “ZOH”, “Mapeo”, “Bin(s-z)” o “Bin(z-ω)”
	ana	Tipo de análisis, “Escalón”, “Impulso” o “Rampa”
	tf	Valor de la duración del análisis (tiempo final)
	amp	Valor de la amplitud o pendiente para el análisis
	A	Coeficientes de la matriz A del sistema en EE
	B	Coeficientes de la matriz B del sistema en EE
	C	Coeficientes de la matriz C del sistema en EE
	D	Coeficientes de la matriz D del sistema en EE
	X0	Coeficientes del vector X0 (condiciones iniciales) del sistema en EE
	tran	Transformación entre representaciones: “TF2SS” o “SS2TF”
	k	Valor de la ganancia máxima para el análisis del Lgdr
	fmin	Valor de la frecuencia mínima para el análisis frecuencial
	fmax	Valor de la frecuencia máxima para el análisis frecuencial
Salida	*dat*	Valores para graficar el análisis
	lim	Límites máximo y mínimo de los valores
	num0	Coeficientes obtenidos para el numerador, según transformación
	den0	Coeficientes obtenidos para el denominador, según transformación
	A0	Coeficientes obtenidos para la matriz A, según transformación
	B0	Coeficientes obtenidos para la matriz B, según transformación
	C0	Coeficientes obtenidos para la matriz C, según transformación
	zl	Posición de los ceros para el análisis del Lgdr
	pl	Posición de los polos para el análisis del Lgdr
	datosRl	Valores para graficar el análisis para el análisis del Lgdr
	pvk	Posición de las ganancias para el análisis del Lgdr
	pvGF	Valores de los márgenes de ganancia y fase del análisis de Bode
	frq	Ubicación y valor de las frecuencias del análisis de Nyquist
	fle	Ubicación, dirección y tamaño de las flechas del análisis de Nyquist

Después de construir la *trama de petición* se verifican los valores ingresados de los argumentos de entrada por medio del método *verificar()* (ver Figura 4-7), el cual cuenta con una serie de condiciones basadas en las especificaciones y restricciones establecidas en la sección 4.2.1.3. Si existe error por medio de un mensaje se le informa al usuario que realice la respectiva verificación de los datos ingresados. Si no existe error, el usuario envía la trama por medio del botón , el cual tiene asociado el método *enviarSolicitud()*. Este método agrupa los campos de la *trama de petición* por medio de una clase en Java denominada *Hashtable* y la envía a través del flujo de salida establecido.

Por último la acción del botón  lanza el método *limpiar(),* el cual se encarga de limpiar el área gráfica y ajustar los botones de selección, parámetros de entrada, los coeficientes de las representaciones de entrada FDT y EE y ventanas emergentes de la interfaz a valores por defecto. Finalmente la acción del botón “Precargar en FDT Discreta” lanza el método *copiarCoeficientes()*, el cual se encarga de copiar los coeficientes de los campos Numerador y Denominador de la representación FDT de salida, en los campos Numerador y Denominador de la representación FDT de entrada.

4.3.1.2 Condiciones para realizar un análisis o transformación

La interfaz gráfica de usuario de HASC-Cliente implementa una serie de condiciones para realizar un determinado análisis o transformación, las cuales dependen de la manera como se represente el sistema, como FDT o EE, como lo muestra la Figura 4-10.

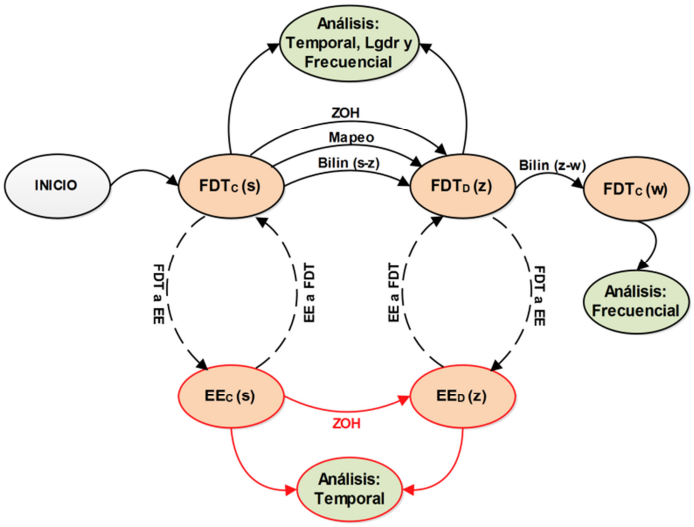


Figura 4-10. Condiciones para realizar un análisis y/o transformación en HASC-Cliente

De la figura anterior, los estados de color rosa corresponden a la representación del sistema, los de color verde al análisis permitido, el borde rojo en los estados se usa para indicar un sistema en EE multi-variable. Las flechas solidas indican los métodos usados para realizar la transformación entre planos (continuo a discreto, o discreto a plano ω) y las flechas punteadas indican los métodos usados para realizar la transformación entre representaciones (de FDT a EE y viceversa).
Por tanto se deben cumplir las siguientes condiciones:

- Para los análisis se tienen:
  - En FDT continua o discreta se permite realizar el análisis temporal, Lgdr y frecuencial.
  - En FDT en el plano ω solo se realiza el análisis frecuencial.
  - En EE mono/multi-variable continuo o discreto, únicamente se permite realizar el análisis temporal.
- Para la transformación entre planos se tienen:
  - En FDT continua se permite realizar la transformación del plano continuo al discreto por medio de los métodos: ZOH, Mapeo polo cero y Bilineal (S-Z).
  - En FDT discreta se permite realizar la transformación del plano discreto al plano ω, por medio del método Bilineal (Z-ω).
  - En EE continuo mono/multi-variable se permite realizar la transformación del plano continuo al discreto por medio del método ZOH.

- Para la transformación entre representaciones se tienen:
  - En FDT continua o discreta se permite realizar la representación a EE por medio del método FDT a EE.
  - En EE mono-variable continuo o discreto se permite realizar la representación a FDT por medio del método EE a FDT, mientras que en EE multi-variable no se permite realizar esta transformación.

4.3.2 Aplicación HASC-Servidor

La aplicación servidor implementa una sencilla interfaz gráfica, la cual se desarrolló en EJS y se encarga de establecer la conexión local con Scilab, abrir el canal de comunicación y estar a la espera por solicitudes de conexión de los clientes, como lo muestra la Figura 4-11, donde se observa que se han diferenciado tres áreas.

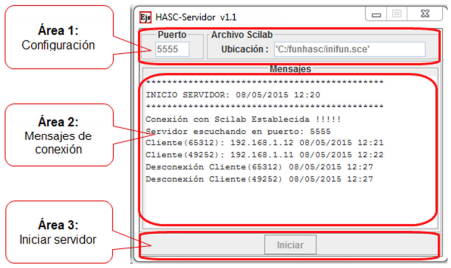


Figura 4-11 Interfaz de usuario de la aplicación HASC-Servidor

Área 1: Donde por medio de campos numéricos y de texto, se asigna el número de puerto del servidor y se establece la ruta del script de Scilab "*inifun.sce*". Este script como se explicará más adelante, contiene las funciones necesarias para procesar los análisis y transformaciones en Scilab.

Área 2: Donde se visualizan mensajes de información acerca de los clientes conectados con su dirección IP, puerto asignado, y fecha y hora de establecimiento de la conexión.

Área 3: Donde se inicia la ejecución del servidor por medio del botón "Iniciar".

4.3.2.1 Algoritmo de ejecución

El funcionamiento de la aplicación HASC-Servidor se basa en el algoritmo de ejecución que se muestra en la Figura 4-12.

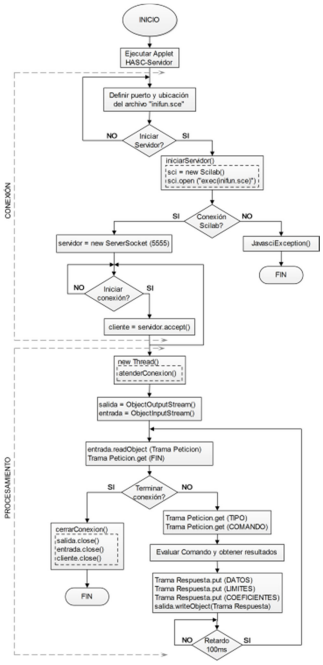


Figura 4-12 Algoritmo de ejecución de la aplicación HASC-Servidor

HASC-Servidor ejecuta los procesos de conexión y procesamiento de las solicitudes, de manera independiente por medio del uso de hilos (*threads*) en Java. El hilo principal se encarga de establecer *Conexión* con Scilab, escuchar permanente por peticiones de conexión de los clientes y asignar a cada conexión TCP/IP establecida un hilo independiente de ejecución. En cada hilo asignado, se realiza el *Procesamiento* de las solicitudes enviadas y se retorna los resultados a su respectivo cliente. A continuación se explican estos dos procesos.

4.3.2.1.1 Conexión

Al ejecutar el servidor, inicialmente se debe ajustar el número del puerto del servidor el cual por defecto tiene el valor *5555*, e indicar la ruta de ubicación del archivo *inifun.sce*, que por defecto muestra "*C:/funhasc/inifun.sce*" (si el servidor se ejecuta sobre Windows). Es importante mencionar que la apertura del puerto *TCP/5555* se llevó a cabo gracias a la gestión realizada por el administrador de la plataforma PERI [38] con la OITEL [64] (dependencia encarga de prestar servicios informáticos y de telecomunicaciones en la Universidad del Valle). Para efectos de seguridad exigidos por la OITEL se establecieron reglas de filtrado de puertos y control de acceso en el servidor PERI.

El archivo *inifun.sce* corresponde a un script de Scilab y debe ejecutarse siempre con el servidor ya que contiene 15 funciones necesarias para realizar los correspondientes análisis y/o transformaciones. Las funciones implementadas se muestran en la Tabla 4-3, donde para cada análisis se describen las funciones principales y auxiliares necesarias para calcularlo. Es importante mencionar que los argumentos de entrada y salida que conforman las funciones, en este caso las principales, se describieron en la Tabla 4-2.

Tabla 4-3 Descripción de las funciones que utiliza Scilab para un análisis y/o transformación

Análisis	Función	Descripción
Temporal	*rtatem.sci*	Función principal que contiene las funciones auxiliares: *analisc.sci*, *analisd.sci, represis.sci, represiz.sci y orcofout.sci,* que permiten calcular los datos para construir la respuesta temporal y los coeficientes de las transformaciones.
	analisc.sci	Función auxiliar que calcula los datos de la respuesta temporal de un sistema continúo.
	analisd.sci	Función auxiliar que calcula los datos de la respuesta temporal de un sistema discreto.
LGDR	*Lgdr.sci*	Función principal que contiene las funciones auxiliares: *represis.sci, represiz.sci, orcofout.sci y evansm.sci,* que permiten calcular los datos para construir el lugar de las raíces y los coeficientes de las transformaciones.
	evansm.sci	Función auxiliar basada en la función propia de Scilab *evans.sci*, modificada con el fin de obtener datos gráficos del LGDR.
Bode	*bod.sci*	Función principal que contiene las funciones auxiliares: *represis.sci, represiz.sci, orcofout.sci y bodem.sci,* que permiten calcular los datos para construir el análisis de Bode y los coeficientes de las transformaciones.
	bodem.sci	Función auxiliar basada en la función propia de Scilab *bode.sci*, modificada con el fin de obtener datos gráficos del diagrama de Bode.
Nyquist	*nyq.sci*	Función principal que contiene las funciones auxiliares: *represis.sci, represiz.sci, orcofout.sci y nyquistm.sci,* que permiten calcular los datos para construir al análisis de Nyquist y los coeficientes de las transformaciones.
	nyquistm.sci	Función auxiliar basada en la función propia de Scilab *nyquist.sci*, modificada con el fin de obtener datos gráficos del diagrama de Nyquist.
Funciones Auxiliares		
represis.sci	Construye la representación en FDT o EE de un sistema continuo.	
represiz.sci	Construye la representación en FDT o EE de un sistema discreto.	

t_ss2tf.sci	Calcula la transformación de EE a FT y sus coeficientes
orcofout.sci	Organiza los coeficientes obtenidos de las transformaciones
tranSZ.sci	Calcula la transformación entre planos y sus coeficientes
mapeo.sci	Calcula la transformación del plano continuo al discreto con el método mapeo polo-cero.

El servidor debe estar en capacidad de procesar las solicitudes de los diferentes análisis y/o transformaciones en Scilab, para ello se estableció la conexión entre la aplicación HASC-Servidor y el motor de cálculo de Scilab, por medio de la interfaz *Javasci* (descrita en la sección 4.1.1). El procedimiento utilizado para establecer la conexión se detalla en el documento: *Guía conexión EJS-Scilab*, el cual se encuentra disponible en el Anexo E.

La ejecución del servidor inicia al hacer clic sobre el botón “Iniciar”, la acción de este botón lanza el método *iniciarServidor(),* el cual contiene los métodos que implementa *Javasci* para establecer conexión y abrir una sección con Scilab; y ejecutar y cargar el archivo *inifun.sce* en la sección de Scilab abierta. Si se produce un error de conexión con Scilab, se lanza una excepción y finaliza la ejecución del algoritmo. Si no se produce error, se abre un *socket* (*ServerSocket*) al cual se le asocia el puerto *5555*, en este punto la ejecución del algoritmo permanece a la espera, escuchando a través del *socket* por peticiones de conexión de los clientes.

Cuando un cliente solicita conexión, se le asigna un nuevo *socket* y se crea un hilo de ejecución independiente que contiene el método *atenderConexion()* con el fin de atender sus peticiones. Después de llevar a cabo esta acción, la ejecución del programa nuevamente queda a la espera, escuchando por nuevas peticiones de conexión.

4.3.2.1.2 Procesamiento

Las peticiones del cliente se atienden en un hilo de ejecución que lanza el método *atenderConexion()*. En este método inicialmente se crean y asocian flujos intermedios a los flujos de entrada (InputStream) y de salida (OutputStream) asignados al *socket*. Posteriormente se lee la *trama de petición,* se procesa y se construye la *trama de respuesta* como lo muestra el diagrama de flujo de la Figura 4-13. Finalmente se realiza continuamente la lectura del flujo de entrada hasta que el cliente decida terminar la conexión.

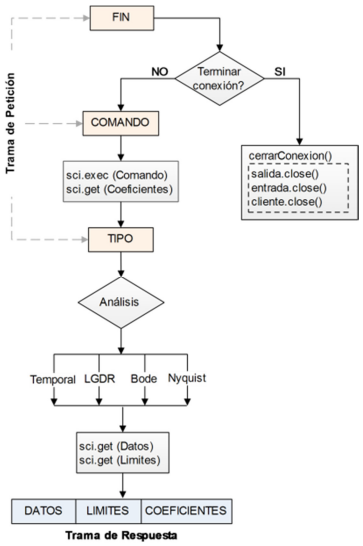


Figura 4-13 Tratamiento de las tramas de petición y de respuesta en HASC-Servidor

Inicialmente de la *trama de petición* se extrae la información del campo FIN, este campo contiene una palabra (“Cerrar”) que indica si el cliente desea terminar la conexión. Si no se termina, se extrae la función contenida en el campo COMANDO y se ejecuta en Scilab, obteniendo de los argumentos de salida de la función el valor de los Coeficientes. Posteriormente del campo TIPO se obtiene el tipo de análisis solicitado con el fin de obtener: los otros argumentos de salida de la función evaluada, el valor de los Datos y Límites de cada análisis. Los valores obtenidos de los Coeficientes, Datos y Limites, son almacenados en arreglos de tipo “*double[][]*” y organizados en los campos de la *trama de respuesta* por medio de una clase Java denominada *Hashtable*, para finalmente enviar esta trama a través del flujo de salida establecido.

Es importante destacar que el algoritmo de ejecución de HASC-Servidor (ver Figura 4-12), en el procesamiento de las tramas implementa un retardo de 100 ms entre lecturas del flujo de entrada, con el fin de evitar bloqueos del servidor al atender posibles solicitudes generadas de manera instantánea.

4.4 Pruebas de Funcionamiento

Con el fin de verificar el funcionamiento de las aplicaciones HASC-Cliente y HASC-Servidor desarrolladas, se realiza una prueba de conexión en la aplicación HASC-Servidor y una prueba de uso en la aplicación HASC-Cliente. La prueba diseñada plantea tres aspectos: objetivo, procedimiento y resultado.

4.4.1 Uso y Prueba de HASC-Servidor

Objetivo:
Probar el correcto funcionamiento de la conexión local con Scilab y la conexión remota con dos clientes.

Procedimiento:
1. Ejecutar el Applet de la aplicación HASC-Servidor.
2. Establecer por número del puerto el valor 5555, y por ruta del archivo “*inifun.sce”* la ubicación “*'C:/funhasc/inifun.sce'*”.
3. Hacer clic en el botón “Iniciar”.

Resultado:
Al iniciar el servidor automáticamente se estableció conexión con Scilab y permaneció a la escucha por solicitudes de conexión de los clientes, como lo muestran los mensajes obtenidos por la aplicación en la Figura 4-14.

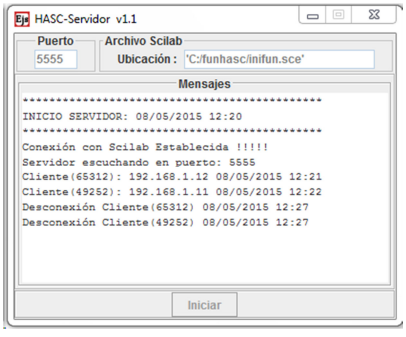


Figura 4-14 Resultado de la prueba de HASC-Servidor

Posteriormente se observa que se estableció y atendió la conexión con dos clientes; donde se muestra información de su dirección IP, puerto asignado, y fecha y hora de establecimiento de la conexión.

4.4.2 Uso y Prueba de HASC-Cliente

Objetivo:
Probar el correcto funcionamiento al realizar el análisis temporal de un sistema continuo ante una entrada de prueba tipo escalón. El sistema a analizar presenta la siguiente FDT:

$$G_P(s) = \frac{3}{s^2 + 4s + 20}$$

Procedimiento:

1. Cargar la página: http://eieela.univalle.edu.co/ludicas_animadas/menu_ludicas.html, y en la sección HASC permitir la ejecución del Applet de la aplicación.
2. En el campo Tipo de Sistema seleccionar “Continuo (S)”.
3. En el campo Representación de Entrada seleccionar “FDT”.
4. Ingresar los coeficientes de la FDT en los campos Numerador y Denominador de la siguiente manera:

$$Numerador = [3]$$
$$Denominador = [1 \quad 4 \quad 20]$$

5. En el campo Tipo de Análisis seleccionar “Escalón”.
6. En el campo Parámetros de Entrada asignar los valores de amplitud de la señal escalón y el tiempo de duración:

$$Amplitud\ (und) = 1$$
$$Duración\ (seg) = 4$$

7. En el campo Ejecutar hacer clic en el botón .

Resultados:
Se observa que se obtiene la respuesta temporal del sistema bajo análisis, como lo muestra la Figura 4-15.

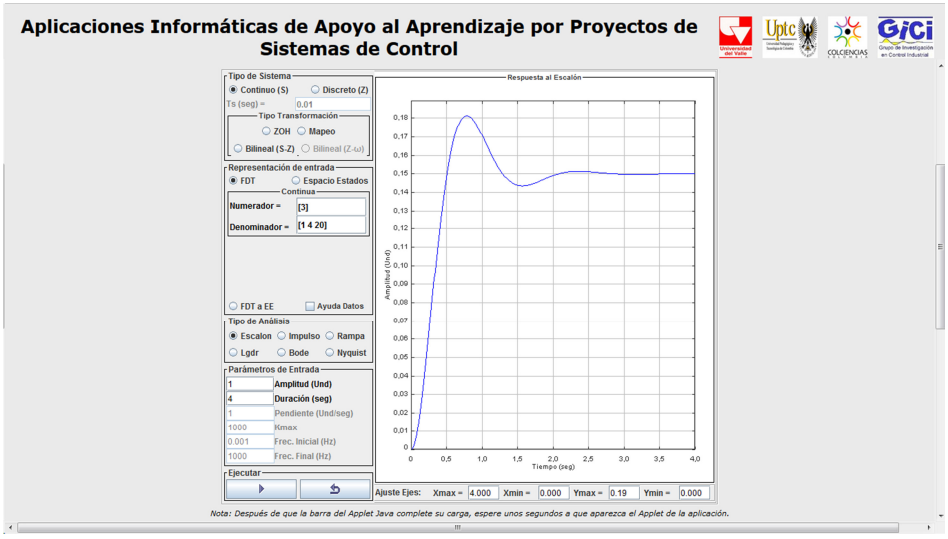


Figura 4-15 Resultado de la prueba de HASC-Cliente

Comentario: Las pruebas realizadas en conexión y uso de la aplicación fueron satisfactorias; ya que permitieron verificar el correcto funcionamiento de la herramienta HASC al obtener de manera remota el análisis temporal de un sistema continuo. Las pruebas de los restantes análisis y transformaciones se reportaron en el documento: *Guía HASC usuario*, el cual se encuentra disponible en el Anexo D.

4.5 Resumen

En este capítulo se presentó el diseño y desarrollo de dos aplicaciones HASC-Cliente y HASC-Servidor como soporte matemático para el análisis de los sistemas de control de manera remota; se basa en una arquitectura Cliente – Servidor que utiliza el protocolo de comunicación TCP/IP. La interfaz de usuario implementada cuenta con campos de ingreso de valores y opciones de selección que permiten de una manera fácil y rápida realizar análisis como: el temporal, frecuencial, el lugar de las raíces de un sistema continuo o discreto representado por su función de transferencia FDT o por su modelo en el espacio de estados EE; además realiza transformaciones entre planos (del plano continuo al discreto, o del plano discreto al plano ω) y entre representaciones (de FDT a EE y viceversa), mostrando los coeficientes calculados en una ventana emergente. En ambas aplicaciones se presentaron los algoritmos de ejecución concebidos para la implementación y envió de solicitudes; y procesamiento y retorno de resultados. Las pruebas de funcionamiento realizadas en conexión y uso de la aplicación fueron satisfactorias, ya que permitieron obtener de manera remota el análisis temporal de un sistema continuo.

Para ilustrar el procedimiento de conexión entre EJS y Scilab se elaboró el documento: *Guía conexión EJS-Scilab*, el cual se encuentra disponible en el Anexo E; y para el correcto manejo y operación de la aplicación, se elaboró el documento: *Guía HASC usuario*, el cual se encuentra disponible en el Anexo D.

5 EVALUACIÓN

En este capítulo se describe el diseño del instrumento utilizado para la evaluación de las aplicaciones informáticas desarrolladas: Lúdicas Animadas, Herramienta de Análisis del Controlador PID HAPID y Herramienta de Análisis de Sistemas de Control HASC. El análisis de los resultados se realiza a partir de los datos proporcionados por una encuesta que entrega información cuantitativa de la calificación cualitativa que los usuarios dan a cada uno de los interrogantes presentes en ésta, utilizando las opciones de respuesta del instrumento. Para cada aplicación informática se presenta el análisis de los resultados basados en aspectos como: el funcionamiento de la aplicación, su impacto en el aprendizaje y el desarrollo de habilidades transversales.

5.1 Instrumento de Evaluación

Como instrumento de evaluación se utilizaron tres encuestas orientadas a evaluar cada aplicación informática. El diseño de las encuestas se caracteriza por: presentar las *preguntas como afirmaciones* buscando que los encuestados califiquen el grado de cumplimiento o aceptación de cada afirmación utilizando una escala de 1 a 5, 1 No se cumple, 2 Bajo, 3 Aceptable, 4 Bueno y 5 Excelente; su estructura se basa en las encuestas propuestas por [5], en donde se consulta la opinión de los estudiantes y los profesores para evaluar el enfoque ABP en los cursos de control y los recursos informáticos utilizados como apoyo.

En las encuestas las preguntas se agrupan en tres aspectos: funcionamiento de la herramienta, contribución al aprendizaje y desarrollo de habilidades transversales. El funcionamiento busca evaluar características técnicas como: organización y presentación de las interfaces de usuario, comportamiento de la herramienta, usabilidad y conveniencia del material de apoyo. El aspecto de aprendizaje evalúa la percepción del encuestado sobre la utilidad, pertinencia y facilidad en la compresión de conceptos de sistemas de control. El aspecto de habilidades busca conocer las fortalezas y debilidades en el desarrollo de competencias transversales como, aprendizaje autónomo, capacidad para tomar decisiones y resolver problemas, recursividad y creatividad; con respecto a estas habilidades el enfoque ABP aplicado a los cursos de control se centra en el desarrollo de este tipo de competencias (ver sección 2.3). En el Anexo G se presentan las encuestas diseñadas para las aplicaciones: Lúdicas Animadas, HAPID y HASC. Es importante aclarar que las 15 Lúdicas Animadas cuentan con su propia encuesta, para ser evaluadas de manera específica. El diseño de la encuesta de cada lúdica presenta los mismos ítems de los aspectos Funcionamiento y Habilidades transversales de la encuesta general del Anexo 1; sin embargo dependiendo del concepto que se requiera introducir los ítems del aspecto Aprendizaje son diferentes.

La confiabilidad de las encuestas fue evaluada utilizando el coeficiente *Alpha Cronbach* [65]. Este coeficiente puede obtenerse a través de la siguiente ecuación.

$$\alpha = \left(\frac{k}{k-1}\right)\left(1 - \frac{\sum {S_i}^2}{{S^2}_{sum}}\right), \tag{94}$$

donde, k es el número de preguntas (ítems) de la encuesta, S_i^2 es la varianza de cada ítem y S^2_{sum} es la varianza de la suma de los ítems. Coeficientes entre 0.8 y 1, indican buena confiabilidad del cuestionario diseñado.

5.2 Resultados de la Evaluación

Las Lúdicas Animadas fueron evaluadas por estudiantes del curso Modelado de Sistemas de la Universidad Pedagógica y Tecnológica de Colombia UPTC, en el primer semestre de 2014. Debido al grado de complejidad en la concepción, diseño y desarrollo de las aplicaciones HASC y HAPID, se retrasó su finalización no alcanzando a ser evaluadas por estudiantes; por tanto fueron evaluadas de forma independiente por profesores de las instituciones: Universidad del Valle y UPTC en Abril de 2015.

Para el análisis de los resultados, las preguntas relacionadas con el Funcionamiento fueron reseñadas con la letra 'F', las asociadas al Aprendizaje con la letra 'A' y las de Habilidades transversales con la letra 'H'. A continuación se describen y analizan los resultados obtenidos de la evaluación de las aplicaciones informáticas.

5.2.1 Lúdicas Animadas

El cuestionario del Anexo 1 fue aplicado al final del desarrollo de cada actividad Lúdica Animada. Durante el periodo evaluado se utilizaron las Lúdicas Animadas 1, 2, 5, 6 y 7. Estas lúdicas tienen como objetivo facilitar el aprendizaje de conceptos, y son comúnmente utilizadas por el docente como estrategias didácticas dentro de las actividades desarrolladas en el salón de clase y como etapa previa al análisis matemático de los temas.

Como se mencionó anteriormente los ítems del aspecto Aprendizaje para cada lúdica son diferentes, por tanto en la Tabla 5-1 se muestran las preguntas de aprendizaje que evalúan cada una de estas lúdicas.

Tabla 5-1 Preguntas de aprendizaje para las Lúdicas Animadas evaluadas

Lúdica	No.	Preguntas de Aprendizaje
		Cree usted que la lúdica animada:
1. Plano y Rueda	QA1.1	- Permitió comprender el comportamiento de un sistema estable e inestable. - Facilitó identificar los elementos y señales que componen la bucla típica de un sistema de control. - Permitió comprender el efecto de la realimentación en el desempeño de un sistema de control.
2. Válvula y Tanque	QA2.1	- Permitió conocer el comportamiento de un sistema lineal y no lineal. - Facilitó determinar un modelo matemático no lineal del sistema. - Facilitó determinar un modelo matemático lineal del sistema.
5. Plano, Rueda y Disturbio	QA5.1	- Permitió comprender el comportamiento de un sistema estable e inestable. - Permitió comprender el desempeño ante disturbios de un sistema, con y sin realimentación.
6. Válvulas y Tanque	QA6.1	- Permitió comprender la respuesta temporal de un sistema de primer orden. - Facilitó determinar los parámetros de un sistema de primer orden a partir de su respuesta temporal.
7. Masa y Resorte	QA7.1	- Permitió comprender la respuesta temporal de un sistema de segundo orden. - Facilitó determinar los parámetros de un sistema de segundo orden a partir de su respuesta temporal.

Las preguntas de aprendizaje se plantearon a partir de la metodología propuesta y los resultados de aprendizaje esperados por los estudiantes de los cursos de sistemas lineales de control (ver sección 2.3.2).

En el caso de la Lúdica 1, 27 estudiantes contestaron la encuesta, 24 para la Lúdica 2, 22 para la Lúdica 5, 18 para la Lúdica 6 y 17 para la Lúdica 7. El promedio de calificación fue de 4.1, 4.1, 4.2, 4.3 y 4.3, respectivamente. Teniendo en cuenta que las preguntas

relacionadas con el aspecto 'Aprendizaje' son diferentes, en cada lúdica se tomó el promedio general de las preguntas asociadas a este factor para hacer la comparación y se les asigno el indicador QA(Av). En la Tabla 5-2 se muestran las preguntas diseñadas para evaluar estas Lúdicas Animadas.

Tabla 5-2 Preguntas para el análisis de las encuestas sobre Lúdicas Animadas

No.	Preguntas
QF1	La animación funcionó adecuadamente durante el desarrollo de la lúdica. (Si es Inadecuado, indique en los comentarios las posibles causas).
QF2	Aspectos gráficos como: el diseño de los botones, barras de deslizamiento, ventanas emergentes, cajones de ingreso y registro de variables, tipo y tamaño de letra, y los colores usados son adecuados.
QF3	El funcionamiento descrito es claro y suficiente para usar la animación.
QF4	La redacción y la ortografía empleada, es adecuada.
QF5	La interfaz de usuario facilita la interacción con la animación.
QF6	La velocidad de ejecución de la animación es adecuada.
QF7	Las variables graficadas son adecuadas y suficientes.
El desarrollo de la lúdica animada fortaleció mi capacidad para:	
QH13	El autoaprendizaje.
QH14	Tomar decisiones.
QH15	Recursividad y creatividad.
QH16	Resolver problemas.

Para las encuestas se obtuvo un valor promedio del *Coeficiente Alpha Cronbach* de 0.82, lo cual indica una buena confiabilidad del instrumento diseñado. La Figura 5-1 contiene los promedios por consulta para cada uno de los casos encuestados y el promedio general.

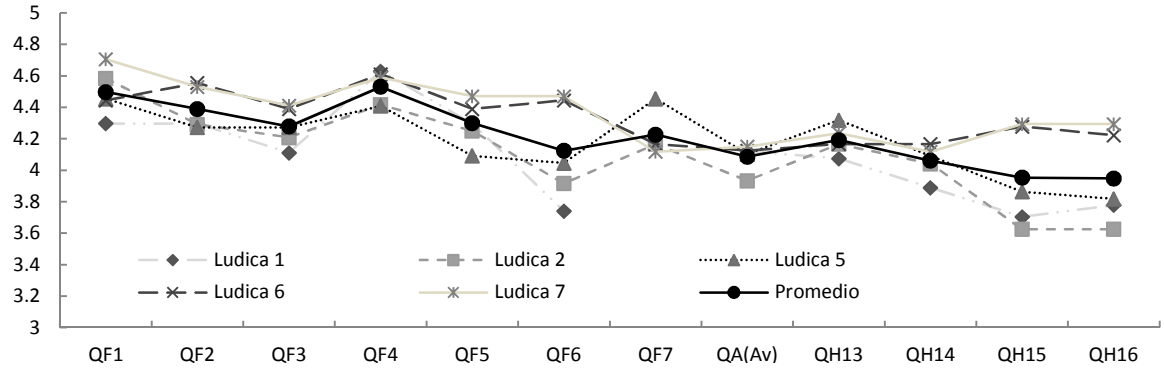


Figura 5-1 Evaluación Lúdicas Animadas, comparación de resultados
El eje horizontal corresponde al número de consulta. La escala usada es de 1 a 5, 1 = no cumplimiento y 5 = excelente nivel de cumplimiento de la afirmación

Como se observa, la mayoría de preguntas fueron calificadas con un puntaje superior de 4.0, lo que evidencia una aceptación de los estudiantes de las Lúdicas Animadas diseñadas dentro de su proceso de formación. La tendencia del promedio es similar en las respuestas asociadas a las consultas del aspecto 'Funcionamiento', observándose una menor calificación en la pregunta QF6 que hace referencia a la velocidad de ejecución. En la mayoría de los casos se observó que la velocidad de ejecución del *Applet* puede verse afectada por la versión de *Java* que se tenga instalada y las limitaciones en Hardware y Software que presente la máquina desde la cual se está ejecutando la animación. El valor promedio obtenido para el aspecto de 'Aprendizaje' QA(Av) fue de 4.1, lo cual indica que los estudiantes consideran que son útiles y facilitan el aprendizaje de los conceptos.

Con respecto a las preguntas del aspecto 'Habilidades Transversales', las consultas sobre el desarrollo de competencias como el autoaprendizaje (QH13) y la toma de decisiones

(QH14) tienen calificaciones similares en los casos evaluados; existen diferencias en los promedios de calificación de las preguntas (QH15) y (QH16) sobre recursividad y creatividad, y la capacidad para resolver problemas, estas diferencias pueden estar relacionadas con el objetivo y tema de cada lúdica, retos mayores como los presentados en las Lúdicas 6 y 7 pueden favorecer en mayor grado estas competencias.

5.2.2 Herramienta de Análisis del Controlador PID

Esta aplicación busca entrenar a los estudiantes para la fase Operativa de un proyecto, ya que permite configurar en un ambiente similar al industrial diferentes fallas, buscando que a partir del comportamiento de las variables del sistema se identifiquen la posible falla con el fin de corregirla.

El cuestionario del Anexo 2 fue aplicado por cinco usuarios para evaluar la herramienta. Las preguntas diseñadas para evaluar esta aplicación se muestran en la Tabla 5-3.

Tabla 5-3 Preguntas de la encuesta sobre HAPID

No.	Preguntas
QF1	El diseño de la plataforma de simulación facilitó la comprensión de su funcionamiento.
QF2	Los ejercicios planteados en la guía de trabajo facilitaron la comprensión de los conceptos.
QF3	Los datos obtenidos estaban disponibles en formatos adecuados para su análisis y la elaboración de informes.
QF4	Las herramientas disponibles en la plataforma son fáciles de operar y utilizar.
QF5	La plataforma de simulación funcionó correctamente durante las prácticas.
La herramienta facilitó la comprensión de conceptos en temas como:	
QA6	El modelado de los sistemas
QA7	El análisis de estabilidad
QA8	Las estrategias de control
QA9	Las características de la retroalimentación
QA10	Operación de un sistema de control
QA11	Fallas en un lazo de control
El desarrollo de la herramienta fortaleció mi capacidad para:	
QH12	Aplicar el conocimiento a la práctica
QH13	Gestionar el tiempo y la información
QH14	El autoaprendizaje
QH15	Tomar decisiones
QH16	Recursividad y creatividad
QH17	Resolver problemas

Para esta encuesta se obtuvo un valor del *Coeficiente Alpha Cronbach* de 1.054, lo cual indica una buena confiabilidad del instrumento diseñado. En la Figura 5-2 se muestran los promedios de las preguntas de la encuesta.

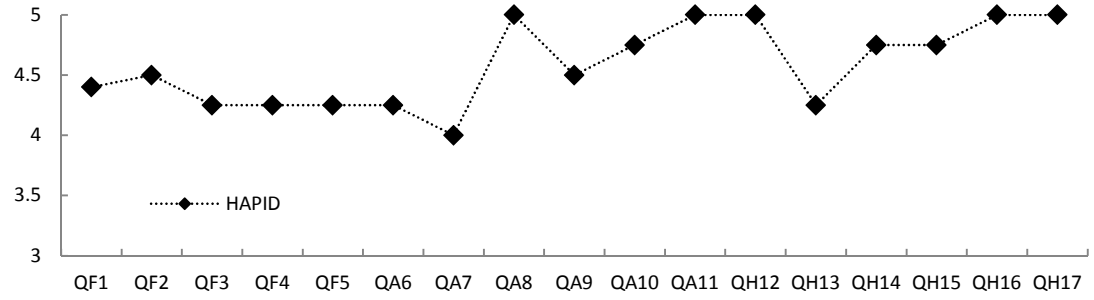


Figura 5-2 Resultado de la evaluación de HAPID
El eje horizontal corresponde al número de consulta. La escala usada es de 1 a 5, 1 = no cumplimiento y 5 = excelente nivel de cumplimiento de la afirmación

De las preguntas del aspecto Funcionamiento se obtuvo un promedio de 4.3, lo cual indica que los usuarios consideran que la herramienta se comportó adecuadamente. Con respecto al grupo de preguntas del aspecto de Aprendizaje, los usuarios calificaron con un promedio de 4.6, destacándose la calificación a las preguntas relacionadas con la operación QA10 y el diagnóstico de fallas QA11, las cuales obtuvieron un nivel de cumplimiento "Excelente". Uno de los usuarios comentó *"Me parece una excelente herramienta, además de que recrea los fenómenos de la planta, las HMI permiten interactuar sacando todos los problemas que se presentan en la planta por un mal ajuste. Considero que a pesar de estar pensada para mostrar los problemas del PID, sirve para trabajar los problemas de caracterización y modelado de la planta"*. Los usuarios destacan principalmente la similitud que existe entre el ambiente simulado y el que se encuentra en una HMI (*Human-Machine Interface*). También encuentran interesante la opción de la herramienta que permite configurar diferentes niveles de prioridad en la configuración y el acceso, ya sea como Ingenieros, Supervisores u Operarios, lo cual además de ser una presentación motivadora para el uso de la herramienta, da la posibilidad de diseñar diferentes prácticas académicas con los estudiantes. En cuanto a las preguntas relacionadas con el desarrollo de competencias transversales, todas fueron calificadas en promedio con nivel de cumplimiento 'Excelente', obteniendo una calificación promedio general de 4.8.

5.2.3 Herramienta de Análisis de Sistemas de Control

El propósito de esta aplicación es realizar de una manera fácil y rápida el análisis en el dominio del tiempo y la frecuencia, o del lugar de las raíces de un sistema continuo o discreto representado por su función de transferencia o por un modelo en espacio de estados.

El cuestionario del Anexo 3 fue aplicado por cinco usuarios para evaluar la herramienta. Las preguntas diseñadas para evaluar esta aplicación se muestran en la Tabla 5-4.

Tabla 5-4 Preguntas de la encuesta sobre HASC

No.	Preguntas
QF1	La herramienta de análisis funcionó adecuadamente. (Si es Inadecuado, indique en los comentarios las posibles fallas).
QF2	Aspectos de la interfaz como: el diseño de los botones, ventanas emergentes, cajones de ingreso y registro de valores; opciones de selección, tipo y tamaño de letra; y los colores usados son adecuados.
QF3	La interfaz de usuario facilitó la interacción con la herramienta de análisis
QF4	La herramienta de análisis fue fácil de utilizar.
QF5	La respuesta gráfica obtenida de los diferentes análisis fue clara y adecuada.
QF6	Los coeficientes obtenidos por medio de una transformación eran claros y de formato adecuado.
QF7	Guardar como archivo tipo imagen los análisis gráficos y los coeficientes obtenidos para posteriores análisis y elaboración de informes, le parece adecuado.
QF8	El material de instrucción existente fue adecuado para facilitar la comprensión de la herramienta de análisis.
Del Aprendizaje	
QA9	La herramienta de análisis es adecuada para los requerimientos de la metodología de aprendizaje por proyectos
QA10	La herramienta de análisis es pertinente para el aprendizaje de sistemas de control.
QA11	La herramienta de análisis facilitó la comprensión de conceptos en temas como: transformaciones continuo-discreto, FdT-Estado, respuesta temporal, frecuencial y lugar geométrico de las raíces.
El desarrollo de la herramienta fortaleció mi capacidad para:	
QH12	El autoaprendizaje.
QH13	Tomar decisiones.
QH14	Recursividad y creatividad.
QH15	Resolver problemas.

Para esta encuesta se obtuvo un valor del *Coeficiente Alpha Cronbach* de 0.924, lo cual indica una buena confiabilidad del instrumento diseñado. En la Figura 5-3 se muestran los promedios de las preguntas de la encuesta.

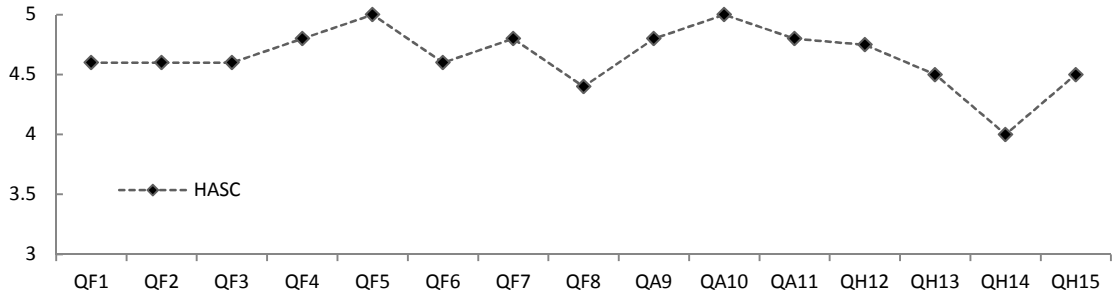


Figura 5-3 Resultado de la evaluación de HASC
El eje horizontal corresponde al número de consulta. La escala usada es de 1 a 5, 1 = no cumplimiento y 5 = excelente nivel de cumplimiento de la afirmación

Los promedios de calificación para los aspectos de Funcionamiento, Aprendizaje y Habilidades Transversales fueron, respectivamente, 4.7, 4.9 y 4.4. Todas las preguntas de la encuesta fueron calificadas con niveles de cumplimiento 'Bueno' o 'Excelente'.

Cabe resaltar, que el objetivo principal de éste recurso está orientado a apoyar el análisis matemático de los sistemas de control, especialmente aquellos que requieren alta abstracción como los relacionados con la Estabilidad. Es así como, la respuesta en el tiempo de los sistemas, la construcción de trazas de Bode, los gráficos de Nyquist y el Lugar de las Raíces facilitan la comprensión de estos temas, lo que se ratifica en la calificación de la consulta QF5, que obtuvo un promedio de 5.0. La interfaz de usuario permite modificar los parámetros de las dinámicas a analizar generando una interacción del estudiante con la herramienta en la simulación de diferentes casos y condiciones de operación, por lo cual todos los usuarios calificaron la pregunta QA10 con 5.0, relacionada con la pertinencia de la herramienta para el aprendizaje de sistemas de control.

En cuanto a las habilidades transversales se observa una disminución en la pregunta relacionada con la recursividad QH14, esta calificación se debe a que la herramienta está más enfocada al autoaprendizaje que a otras competencias, objetivo que se valida a través de la calificación obtenida por la pregunta QH12, la cual fue evaluada con un promedio de calificación de 4.8.

5.3 Resumen

En este capítulo se presentó la evaluación de las aplicaciones informáticas desarrolladas: Lúdicas Animadas, HAPID y HASC; como instrumento de evaluación se utilizaron encuestas con una escala de valoración de 1 a 5, 1 = no cumplimiento y 5 = excelente nivel de cumplimiento. Los resultados se analizaron teniendo en cuenta tres aspectos: funcionamiento de la herramienta, contribución al aprendizaje y desarrollo de habilidades transversales. Para las Lúdicas Animadas, se evidencia una aceptación de los estudiantes de las animaciones diseñadas dentro de su proceso de formación, ya que consideran que son útiles por facilitar el aprendizaje de los conceptos. Para la aplicación HAPID los usuarios la consideran como una excelente herramienta para el aprendizaje en operación y diagnóstico de fallas, y la destacan por la similitud que existe entre el ambiente simulado y el que se encuentra en una HMI de un ambiente SCADA industrial. Para la aplicación HASC los usuarios la consideran como una herramienta pertinente para el aprendizaje de sistemas de control por facilitar la compresión en temas como: respuesta temporal, frecuencial y lugar geométrico de las raíces. Del uso de las aplicaciones se evidencia que promueven el desarrollo de habilidades transversales como el auto-aprendizaje, la toma de decisiones y la resolución de problemas.

6 CONCLUSIONES Y TRABAJOS FUTUROS

6.1 Conclusiones

En este trabajo se desarrollaron tres tipos de aplicaciones informáticas: 15 Lúdicas Animadas, una Herramienta de Análisis del Controlador PID HAPID y una Herramienta de Análisis de Sistemas de Control HASC, para apoyar la educación de pregrado en sistemas de control, del área de automática de la Universidad del Valle, con el enfoque de aprendizaje por proyectos; las herramientas se desarrollaron con los software libres *Easy Java Simulations EJS* y *Scilab*; con *EJS* se implementaron las interfaces gráficas de usuario y se usó *Scilab* como motor de cálculo matemático. Las aplicaciones diseñadas presentan interfaces animadas con capacidades de interactividad, son fáciles de manejar y entender; pueden ser utilizadas por varios estudiantes en cualquier momento y lugar, ya que son de uso libre y se encuentran disponibles en un sitio web que las contiene como *Applets de Java*, el cual implementa un esquema de navegación para su fácil acceso y correcta ejecución. Por estar implementadas como *Applets* son aplicaciones que se pueden utilizar en los sistemas operativos Windows y Linux, ya que requieren únicamente del entorno de ejecución de Java para su ejecución.

Las Lúdicas Animadas están dirigidas a facilitar el aprendizaje de los conceptos más importantes de la teoría de los sistemas de control en temáticas como: realimentación, modelado matemático, análisis temporal y frecuencial, estabilidad, compensación, reconstrucción de señales, estructuras de control, controlabilidad y observabilidad. Pueden ser utilizadas por el docente como estrategias didácticas dentro de las actividades desarrolladas en el salón de clase y como etapa previa al análisis matemático de los temas, también por los estudiantes en el libro: "*Sistemas de control, enfoque de proyecto*" [53], orientado al auto estudio. En ellas los estudiantes entienden los conceptos sin una descripción matemática explícita y los motiva a aprender de una manera divertida mientras juegan con las animaciones. En la lúdica animada se describe de manera general el sistema, se dan unas actividades a realizar y se plantean unas preguntas de discusión; no tienen por lo tanto de manera explícita los objetivos y resultados de aprendizaje esperados; por lo anterior, para el uso de las Lúdicas Animadas por un docente, se elaboró el documento: *Guía profesor animaciones*, el cual se encuentra disponible en el Anexo A, donde se presentan los objetivos y resultados de aprendizaje esperados del uso de las Lúdicas Animadas como recurso de apoyo para el aprendizaje de conceptos en la educación en control.

La aplicación HAPID busca entrenar a los estudiantes para la fase Operativa y de Mantenimiento de un proyecto de control resuelto con un controlador PID. Esta herramienta simula un ambiente similar al de un *Supervisory Control And Data Acquisition SCADA* industrial para un intercambiador de calor y presenta funcionalidades como: control de acceso con tres niveles de jerarquía (Ingeniero, Supervisor y Operario), gestión de permisos, generación y reconocimiento de alarmas, ajuste de parámetros del PID, modo de operación (Manual - Automático), configuración de la válvula de control y del transmisor de temperatura, registro de eventos, elaboración de tres tipos de reportes (configuración, datos y eventos), y generación de diez fallas típicas que afectan los elementos (controlador, actuador y sensor) del lazo de control, buscando que los estudiantes a partir del comportamiento de las variables del sistema identifiquen la posible falla con el fin de corregirla. Cuenta con aspectos prácticos de implementación como:

acción de control PID en estructura ISA, filtrado de la derivada, antiembalamiento de la acción integral, transferencia suave en el cambio del modo de operación, modelos matemáticos que simulan comportamientos como histéresis, banda muerta y características del caudal en la válvula de control, y ruido, filtrado y calibración en el transmisor de temperatura. HAPID simula efectos de falla en los elementos que conforman el lazo de control, debidos a desgaste o desajuste del instrumento. La posibilidad de generar fallas fomenta el aprendizaje y genera habilidades en aspectos de mantenimiento como: la detección, el diagnóstico y la reparación. El efecto de las fallas fue cuidadosamente diseñado para evitar bloqueos totales en la ejecución de la aplicación, como es el caso de la falla "Amplificación de Ruido" donde se altera la frecuencia de corte del filtro de medida, a un valor de frecuencia que se limita por el periodo de muestreo del sistema. Con el fin de mostrar el efecto de las 10 fallas en el comportamiento del sistema, el diagnóstico y procedimiento de reparación, se elaboró el documento: *Guía reparación fallas*, el cual se encuentra disponible en el Anexo B; y para el correcto manejo y operación de la aplicación, se elaboró el documento: *Guía HAPID usuario*, el cual se encuentra disponible en el Anexo C.

La aplicación HASC facilita el procedimiento de análisis al obtener de una manera fácil y rápida el comportamiento temporal, frecuencial o del lugar de las raíces de un sistema continuo o discreto representado por su función de transferencia FDT o por su modelo en el espacio de estados EE; además amplía su utilidad al realizar transformaciones entre planos (del plano continuo al discreto, o del plano discreto al plano ω) y entre representaciones (de FDT a EE y viceversa), mostrando los coeficientes calculados en una ventana emergente. HASC implementa una arquitectura Cliente – Servidor, donde las solicitudes del cliente se manejan a través de peticiones, lo cual no representa una alta carga para el servidor ni para el usuario en términos de recursos y tráfico de red entre la aplicación cliente-servidor, permitiendo establecer conexión con varios usuarios a la vez. Para ilustrar el procedimiento de conexión entre EJS y Scilab se elaboró el documento: *Guía conexión EJS-Scilab*, el cual se encuentra disponible en el Anexo E; y para el correcto manejo y operación de la aplicación, se elaboró el documento: *Guía HASC usuario*, el cual se encuentra disponible en el Anexo D.

Las aplicaciones informáticas se evaluaron mediante encuestas con una escala de valoración de 1 a 5, 1 No se cumple, 2 Bajo, 3 Aceptable, 4 Bueno y 5 Excelente, buscando conocer la percepción de los encuestados en aspectos como: funcionamiento de la herramienta, contribución al aprendizaje y desarrollo de habilidades transversales. Los resultados del análisis de las encuestas muestran que para las Lúdicas Animadas, hay una aceptación de los estudiantes de las animaciones diseñadas dentro de su proceso de formación, ya que la mayoría de preguntas fueron calificadas con un puntaje superior de 4.0, además el puntaje promedio de 4.1 del aspecto de 'Aprendizaje' indica que las consideran útiles al facilitar el aprendizaje de los conceptos. Para la aplicación HAPID los usuarios (profesores) la consideran como una excelente herramienta para el aprendizaje en operación y diagnóstico de fallas al obtener estos aspectos calificaciones superiores a 4.5, además la destacan por la similitud que existe entre el ambiente simulado y el que se encuentra en una HMI de un ambiente SCADA industrial. Para la aplicación HASC los usuarios (profesores) calificaron con un valor promedio de 5.0 los diferentes análisis gráficos obtenidos y por tanto la consideran como una herramienta pertinente para el aprendizaje de sistemas de control por facilitar la comprensión de conceptos en temas como: respuesta temporal, frecuencial y lugar geométrico de las raíces.

Las aplicaciones informáticas desarrolladas ofrecen un entorno de aprendizaje, que permite a los estudiantes realizar diferentes tipos de actividades para comprender conceptos abstractos, analizar comportamientos y diseñar sistemas de control. Presentan interfaces animadas con capacidades de interactividad, son fáciles de manejar y entender; pueden ser utilizadas por varios estudiantes en cualquier momento y lugar, ya que son de uso libre, de fácil acceso y ejecutables en los sistemas operativos Windows y Linux. Su uso apoya el aprendizaje por proyectos, estimula el aprendizaje activo y promueve el desarrollo de habilidades transversales como el auto-aprendizaje, la toma de decisiones y la resolución de problemas.

6.2 Trabajos Futuros

Son varios los ejes de trabajo futuro para mejorar y ampliar el trabajo desarrollado; para aumentar la difusión del sitio web con las aplicaciones informáticas se puede generar una versión en inglés. Entre muchas posibilidades de ampliación se consideran:

Para las Lúdicas Animadas se pueden desarrollar más lúdicas para el aprendizaje de otros conceptos básicos de la teoría de control, como por ejemplo: animaciones de sistemas físicos que representen comportamientos de ceros de fase no mínima, tiempo muerto, ceros resonantes, etc. También pueden desarrollarse Lúdicas Animadas para conceptos avanzados de control en temáticas como: control no – lineal, optimo, etc. Igualmente, se puede crear una nueva línea de desarrollo para animar gráficos matemáticos, por ejemplo animar la correlación plano S a plano Z, el efecto de agregar polos y ceros en el lugar de las raíces, etc.

Para aumentar las prestaciones de la herramienta HAPID, se pueden implementar nuevas fallas y la posibilidad de simular otros procesos. También desarrollar un ambiente más realista para las fallas, desde un punto de vista técnico; por ejemplo para la falla “Pérdida de medida”, mostrar de manera animada la desconexión de un cable en el transmisor de temperatura y para su reparación incluir elementos animados como herramientas eléctricas; tomar como referencia el software de entrenamiento para la solución de fallas desarrollado por *Simutech Multimedia* [66].

Para mejorar la herramienta HASC en el aspecto de diseño de sistemas de control, se pueden agregar funcionalidades para realizar el análisis de un sistema en lazo abierto y en lazo cerrado, permitiendo el ingreso de la FDT de un controlador. También se puede ampliar su utilidad al incluir un entorno adicional que permita ejecutar líneas de comando compatibles con Scilab.

Para mejorar la disponibilidad de las aplicaciones informáticas, se puede adecuar su ejecución en dispositivos móviles como *Tablets y Smartphones*. Actualmente es posible realizarlo ya que el software *EJS* cuenta con una nueva versión denominada *EjsS* que desarrolla simulaciones en Javascript y HTML 5, permitiendo la ejecución de las aplicaciones sobre estas plataformas por medio del lector *Reader App* [46].

6.3 Publicaciones

"Interactive Animations for Learning by Playing Concepts of Control Systems", M. Ramírez, J. Ramírez and L. Fernandez, 21st Mediterranean Conference on Control and Automation (MED), Greece, 2013.

"*Software Tools to Support Project-Based Learning of Control Systems"*, M. Ramírez-Ramírez, J.M. Ramírez-Scarpetta and L. Fernandez. Trabajo terminado y pendiente por someterse.

ANEXOS

Anexo A: Guía para el profesor sobre las Lúdicas Animadas

Esta guía se encuentra disponible en el siguiente enlace: Guia_profesor_Animaciones.

Anexo B: Guía para la reparación de fallas en el lazo de control en HAPID

Esta guía se encuentra disponible en el siguiente enlace: Guia_reparacion_fallas.

Anexo C: Guía de usuario de la aplicación HAPID

Esta guía se encuentra disponible en el siguiente enlace: Guia_HAPID_usuario.

Anexo D: Guía de usuario de la aplicación HASC

Esta guía se encuentra disponible en el siguiente enlace: Guia_HASC_usuario.

Anexo E: Guía para establecer conexión entre EJS y Scilab

Esta guía se encuentra disponible en el siguiente enlace: Guia_conexion_EJS-Scilab.

Anexo F: Guía para la instalación y ejecución local de HASC

Esta guía se encuentra disponible en el siguiente enlace: Guia_de_Instalacion_HASC.

Anexo G: Formatos de Evaluación

Anexo 1 Formato general de evaluación para las Lúdicas animadas

	Encuesta a Estudiantes	Versión: 01	15/09/2014
	EVALUACIÓN GENERAL ACTIVIDADES LÚDICAS ANIMADAS		

Periodo Evaluado: _____ Fecha: _____

Asignatura: _____ Nombre del docente: _____

Seleccione la calificación que mejor describa **el nivel de cumplimiento** de la afirmación correspondiente de acuerdo con la tabla:

ESCALA DE VALORACIÓN				
5	4	3	2	1
Excelente	Bueno	Aceptable	Bajo	No se cumple

Aspectos Generales	**VALORACIÓN**				
Cree usted que:	**1**	**2**	**3**	**4**	**5**
1. La animación funcionó adecuadamente durante el desarrollo de la lúdica. (Si es Inadecuado, indique en los comentarios las posibles causas).					
2. Aspectos gráficos como: el diseño de los botones, barras de deslizamiento, cajones de ingreso y registro de variables, tipo y tamaño de letra, y los colores usados son adecuados.					
3. El funcionamiento descrito es claro y suficiente para usar la animación.					
4. La redacción y la ortografía empleada, es adecuada.					
5. La interfaz de usuario facilita la interacción con la animación.					
6. La velocidad de ejecución de la animación es adecuada.					
De Aprendizaje	**1**	**2**	**3**	**4**	**5**
Cree usted que:					
1. Las lúdicas animadas son adecuadas para los requerimientos de la metodología de aprendizaje por proyectos.					
2. Las lúdicas animadas son útiles para el aprendizaje de conceptos de control.					
Del desarrollo de habilidades transversales	1	2	3	4	5
El desarrollo de las actividades lúdicas animadas fortaleció mi capacidad para:					
1. El autoaprendizaje.					
2. Tomar decisiones.					
3. Recursividad y creatividad.					
4. Resolver problemas.					

Comentarios: Escriba sus aportes para la mejora de las actividades lúdicas animadas

Su evaluación contribuye a mejorar la calidad Académica de la Universidad. Por favor diligencie el cuestionario con objetividad y responsabilidad.

Anexo 2 Formato de evaluación para la aplicación HAPID

	Encuesta a Estudiantes	Versión: 01	21/11/2014
	HERRAMIENTA DE ANÁLISIS DEL CONTROLADOR PID HAPID		

Periodo Evaluado: _____ Fecha: _____

Asignatura: _____ Nombre del docente: _____

Seleccione la calificación que mejor describa **el nivel de cumplimiento** de la afirmación correspondiente de acuerdo con la tabla:

ESCALA DE VALORACIÓN				
5	4	3	2	1
Excelente	Bueno	Aceptable	Bajo	No se cumple

Aspectos Generales	**1**	**2**	**3**	**4**	**5**
Cree usted que:					
1. El diseño de la aplicación facilitó la comprensión de su funcionamiento.					
2. Los ejercicios planteados en la guía de trabajó facilitaron la comprensión de los conceptos.					
3. Los datos obtenidos estaban disponibles en formatos adecuados para su análisis y la elaboración de informes.					
4. Las herramientas disponibles en la plataforma son fáciles de operar y utilizar.					
5. La plataforma de simulación funcionó correctamente durante las prácticas.					
Del Aprendizaje	**1**	**2**	**3**	**4**	**5**
Las herramientas de la plataforma facilitaron la comprensión de conceptos en temas como:					
6. El modelado de los sistemas					
7. El análisis de estabilidad					
8. Las estrategias de control					
9. Las características de la retroalimentación					
10. Operación de un sistema de control					
11. Fallas en un lazo de control					
Del desarrollo de habilidades transversales	1	2	3	4	5
Las herramientas de la plataforma fortalecieron mi capacidad para:					
12. Aplicar el conocimiento a la práctica					
13. Gestionar el tiempo y la información					
14. El autoaprendizaje					
15. Tomar decisiones					
16. Recursividad y creatividad					
17. Resolver problemas					

Sugerencias: _____

_____

_____

Su evaluación contribuye a mejorar la calidad Académica de la Universidad. Por favor diligencie el cuestionario con objetividad y responsabilidad

Anexo 3 Formato de evaluación para la aplicación HASC

	Encuesta a Estudiantes	Versión: 01	15/09/2014
	HERRAMIENTA DE ANÁLISIS DE SISTEMAS DE CONTROL HASC		

Periodo Evaluado: _____ Fecha: _____

Asignatura: _____ Nombre del docente: _____

Seleccione la calificación que mejor describa **el nivel de cumplimiento** de la afirmación correspondiente de acuerdo con la tabla:

ESCALA DE VALORACIÓN				
5	4	3	2	1
Muy Bueno	Bueno	Aceptable	Bajo	No se cumple

Aspectos Generales	VALORACIÓN				
Cree usted que:	1	2	3	4	5
1. La herramienta de análisis funcionó adecuadamente. (Si es Inadecuado, indique en los comentarios las posibles fallas).					
2. Aspectos de la interfaz como: el diseño de los botones, ventanas emergentes, cajones de ingreso y registro de valores; opciones de selección, tipo y tamaño de letra; y los colores usados son adecuados.					
3. La interfaz de usuario facilitó la interacción con la herramienta de análisis.					
4. La herramienta de análisis fue fácil de utilizar.					
5. La respuesta gráfica obtenida de los diferentes análisis fue clara y adecuada.					
6. Los coeficientes obtenidos por medio de una transformación eran claros y de formato adecuado.					
7. Guardar como archivo tipo imagen los análisis gráficos y los coeficientes obtenidos para posteriores análisis y elaboración de informes, le parece adecuado.					
8. El material de instrucción existente fue adecuado para facilitar la comprensión de la herramienta de análisis.					
De Aprendizaje	1	2	3	4	5
Cree usted que:					
1. La herramienta de análisis es adecuada para los requerimientos de la metodología de aprendizaje por proyectos.					
2. La herramienta de análisis es pertinente para el aprendizaje de sistemas de control.					
3. La herramienta de análisis facilitó la comprensión de conceptos en temas como: transformaciones continuo-discreto, FdT-Estado, respuesta temporal, frecuencial y lugar geométrico de las raíces.					
Del desarrollo de habilidades transversales	1	2	3	4	5
La herramienta de análisis fortaleció mi capacidad para:					
1. El autoaprendizaje.					
2. Tomar decisiones.					
3. Recursividad y creatividad.					
4. Resolver problemas.					

Comentarios: Escriba sus aportes para la mejora de la herramienta de análisis

Su evaluación contribuye a mejorar la calidad Académica de la Universidad. Por favor diligencie el cuestionario con objetividad y responsabilidad.

REFERENCIAS

[1] C. A. Ramos Paja, "Sistema Remoto de Educación en Ingeniería de Control Basado en Problemas," Tesis de Maestría en Ingeniería. Universidad del Valle, 2005.

[2] B. B. Cortes, "JMat–Herramienta remota de cálculo y multiusuario para el aprendizaje basado en problemas usando Matlab," *Rev. Fac. Ing.*, vol. 59, pp. 158–168, 2012.

[3] M. Duque, L. Osorio, A. Gauthier, and F. Jimenez, "Active learning environments for automatic control courses," *Int. Conf. Eng. Educ.*, pp. 1–8, 2003.

[4] C. Beaumont, T. Owens, M. Barret-Baxendale, and B. Norton, "Blended problem-based learning for widening participation: a case study," *Innov. Teach. Learn. Inf. Comput. Sci.*, vol. 7, no. 1, pp. 44–72, Jun. 2008.

[5] L. Fernandez-Samaca, "Enfoque De Aprendizaje Basado En Proyectos Para Sistemas Lineales de Control," Tesis de Doctorado En Ingeniería Área Eléctrica Electrónica. Universidad del Valle, 2010.

[6] J. Rossiter, "New games for use in lectures to improve student learning," *Proc. Int. Conf. Eng. Educ. – ICEE 2007*, pp. 1–6, 2007.

[7] R. Leland, "A teaching module for the nyquist stability test using cooperative learning," *Proc. 2003 Am. Soc. Eng. Educ.*, pp. 5509–5518, 2003.

[8] J. A. Rossiter, "Peer Assessment and Games to Enhance Student Learning of Frequency Response Methods," *shef.ac.uk*, no. i, pp. 5087–5092, 2009.

[9] I. Santana, M. Ferre, E. Izaguirre, R. Aracil, and L. Hernandez, "Remote Laboratories for Education and Research Purposes in Automatic Control Systems," *IEEE Trans. Ind. Informatics*, vol. 9, no. 1, pp. 547–556, Feb. 2013.

[10] F. C. Herías and J. S. Moreno, "Recursos didácticos basados en Internet para el apoyo a la enseñanza de materias del área de Ingeniería de Sistemas y Automática," vol. 2, pp. 93–101, 2005.

[11] E. N. Ortiz, L. Fernandez-Samaca, and J. M. R. Scarpetta, "Aproximación al aprendizaje por proyectos en sistemas lineales de control," pp. 1–9, 2011.

[12] L. Johnson, S. Brown, M. Cummins, and V. Estrada, "The Technology Outlook for STEM+ Education 2012-2017: An NMC Horizon Report Sector Analysis," *Austin, Texas New Media Consort.*, p. 26, 2012.

[13] J. M. Ramírez Scarpetta, E. Franco Mejia, J. A. Gonzalez, J. T. Buitrago, L. Samaca Fernandez, and O. Rodriguez-Diaz, "Entorno de Aprendizaje Basado en Proyectos para Sistemas de Control. Código: 110652128453. Contrato No. 401-2011." Programa Nacional de Electrónica, Telecomunicaciones e Informática -

COLCIENCIAS, Universidad del Valle, Universidad Pedagogica y Tecnologica de Colombia. Santiago de Cali, 2011.

[14] L. Fernández-Samacá, J. M. Ramirez, and M. L. Orozco-Gutierrez, "Project-based learning approach for control system courses," *Sba Control. Automação*, vol. 23, no. 1, 2012.

[15] E. De Graaff and A. Kolmos, *Management of change: implementation of problem-based and project-based learning in engineering*. 2007.

[16] "Buck Institute for Education." [Online]. Available: http://bie.org/about/what_pbl. [Accessed: 07-Apr-2013].

[17] N. Kheir, K. Åström, and D. Auslander, "Control systems engineering education," *Automatica*, vol. 32, no. 2, pp. 147–166, 1996.

[18] L. Fernández-Samacá, O. Rodriguez-Diaz, and J. M. Salamanca, "Un Enfoque de Aprendizaje en Proyectos para Sistemas de Control," in *Reunión Nacional ACOFI 2009*, 2009.

[19] S. Bencomo, "Control learning: present and future," *Annu. Rev. Control*, vol. 28, no. 1, pp. 1639–1639, Jul. 2004.

[20] M. Johansson, M. Gafvert, and K. Astrom, "Interactive tools for education in automatic control," *IEEE Control Syst. Mag.*, vol. 18, no. 3, pp. 33–40, Jun. 1998.

[21] B. Wittenmark, H. Haglund, and M. Johansson, "Dynamic pictures and interactive learning," *IEEE Control Syst. Mag.*, vol. 18, no. 3, pp. 26–32, Jun. 1998.

[22] "The MathWorks." [Online]. Available: www.mathworks.com. [Accessed: 10-Feb-2013].

[23] H. Cheng, "Web-based control system design and analysis," *IEEE Control Syst. Mag.*, vol. 24, no. 3, pp. 45–57, Jun. 2004.

[24] "Web-Based Control System Design and Analysis." [Online]. Available: http://www.softintegration.com/webservices/control/. [Accessed: 10-Jan-2013].

[25] S. Uran and K. Jezernik, "Virtual Laboratory for Creative Control Design Experiments," *IEEE Trans. Educ.*, vol. 51, no. 1, pp. 69–75, 2008.

[26] Z. Magyar and K. Zakova, "Using SciLab for building of virtual lab," in *2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET)*, 2010, pp. 280–283.

[27] Scilab Enterprises, "Scilab," 2012. [Online]. Available: http://www.scilab.org/scilab/about. [Accessed: 02-Feb-2013].

[28] J. Sánchez, S. Dormido, R. Pastor, and F. Esquembre, "Interactive learning of

control concepts using Easy Java Simulations," *Work. Internet Based Control*, 2004.

[29] F. Esquembre, "Easy Java Simulations: a software tool to create scientific simulations in Java," *Comput. Phys. Commun.*, vol. 156, no. 2, pp. 199–204, Jan. 2004.

[30] G. Farias, R. De Keyser, S. Dormido, and F. Esquembre, "Developing Networked Control Labs: A Matlab and Easy Java Simulations Approach," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3266–3275, Oct. 2010.

[31] N. Duro, R. Dormido, H. Vargas, and S. Dormido, "The three-tank system: A remote and virtual control laboratory using Easy Java simulations," *Proc. 44th …*, 2005.

[32] R. Dormido, H. Vargas, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, F. Esquembre, and S. Dormido, "Development of a Web-Based Control Laboratory for Automation Technicians: The Three-Tank System," *IEEE Trans. Educ.*, vol. 51, no. 1, pp. 35–44, 2008.

[33] N. Aliane, "Spreadsheet-based control system analysis and design [Focus on Education]," *IEEE Control Syst. Mag.*, vol. 28, no. 5, pp. 108–113, Oct. 2008.

[34] I. Pendharkar, "Rltool for Scilab: a public domain tool for SISO system design," *IEEE Control Syst. Mag.*, vol. 25, no. 1, pp. 23–25, Feb. 2005.

[35] J. L. Guzman, K. Astrom, S. Dormido, T. Hagglund, M. Berenguel, and Y. Piguet, "Interactive learning modules for PID control [Lecture Notes]," *IEEE Control Syst. Mag.*, vol. 28, no. 5, pp. 118–134, Oct. 2008.

[36] Y. Piguet, "SysQuake 3 User Manual," 2004. [Online]. Available: http://www.calerga.com. [Accessed: 01-Feb-2013].

[37] S. Dormido and S. Dormido-Canto, "The role of interactivity in control learning," *Int. J. Eng. Educ.*, vol. 21, no. 6, pp. 1122–1133, 2005.

[38] Grupo de Investigacion en Control Industrial, "Plataforma de Educación Remota en Ingeniería," 2007. [Online]. Available: http://eieela.univalle.edu.co. [Accessed: 06-Dec-2012].

[39] C. A. Ramos-Paja, J. M. Ramirez Scarpetta, and L. Martinez-Salamero, "Integrated Learning Platform for Internet-Based Control-Engineering Education," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3284–3296, Oct. 2010.

[40] Y. Gonzalez, Á. M. Ramos, O. Montes de la Barrera, H. E. Hernandez, and J. M. López, "Juego didáctico, una herramienta educativa para el autoaprendizaje en la Ingeniería Industrial," *Rev. Educ. en Ing.*, pp. 61–68, 2011.

[41] D. Mondeja González, B. Cárdenas Zumalacárregui, M. Martín Campos, and C. Ferrer Serrano, "Juegos Didácticos : ¿Útilies en la Educación Superior?," *Pedagog.*

Univ., vol. 6, no. 3, pp. 65–76, 2010.

[42] U. Münz, C. Böhm, and J. Eck, "A Matlab-Based Game for Advanced Automatic Control Education," *IFAC Proc. Vol.*, vol. 8, no. PART 1, pp. 140–145, 2009.

[43] S. D. J. Sánchez, Sebastián Dormido Canto, G. Farias, "Understanding Automatic Control Concepts by Playing Games," *Int. J. Eng. Educ.*, vol. 27, no. no. 3, pp. 528–534, 2011.

[44] L. F. Samacá and J. M. Ramirez, "Learning Control Concepts in a Fun Way*," *Int. J. Eng. Educ.*, vol. 27, no. 1, pp. 1–13, 2011.

[45] F. Esquembre, *Creación de Simulaciones Interactivas en Java: Aplicaciones a la Enseñanza de la Física*, 1st ed. Madrid: Prentince Hall, 2005.

[46] F. Esquembre, "Easy Java Simulations," 2010. [Online]. Available: http://www.um.es/fem/EjsWiki/Main/HomePage. [Accessed: 01-Mar-2013].

[47] V. H. Oyarzún, "An Integral Web-based Environment for Control Engineering Education," National University for Distance Education, UNED, 2010.

[48] "Cecill and free software," 2013. [Online]. Available: http://www.cecill.info. [Accessed: 01-Jan-2013].

[49] UNAD, "Modelo Cliente/Servidor," *Curso Sistemas Distribuidos - Código:302090*, 2009. [Online]. Available: http://datateca.unad.edu.co/contenidos/208017/ContLin/leccin_1_modelo_clienteservidor.html. [Accessed: 15-Jan-2015].

[50] G. Coulouris, J. Dollimore, and T. Kindberg, *Sistemas Distribuidos: Conceptos y Diseño*, 3rd ed. Pearson Educación, 2001.

[51] S. González and J. Zaefferer, "jQuery UI." [Online]. Available: http://jqueryui.com. [Accessed: 10-Jun-2014].

[52] Á. Franco García, "Movimiento de rodar en un plano inclinado," *Física con ordenador*, 2010. [Online]. Available: http://www.sc.ehu.es/sbweb/fisica/solido/plano_inclinado/plano_inclinado.htm. [Accessed: 18-Mar-2015].

[53] J. M. Ramirez Scarpetta, *SISTEMAS DE CONTROL Enfoque de Proyecto*, 1st ed. Santiago de Cali: Universidad del Valle. Grupo de Investigación en Control Industrial., 2014.

[54] K. K. J. Åström and T. Hägglund, *Advanced PID Control*. United States of America: ISA - Instrumentation Systems and Automation Society, 2006.

[55] Á. Franco García, "Oscilaciones amortiguadas," *Física con ordenador*, 2010. [Online]. Available:

http://www.sc.ehu.es/sbweb/fisica/oscilaciones/amortiguadas/amortiguadas.htm. [Accessed: 18-Mar-2015].

[56] ISA 5.1, *Instrumentation Symbols and Identification*, no. September. American Nuclear Society Specificatioin, 2009.

[57] C. A. Smith and A. B. Corripio, *Control Automático de Procesos: Teoría y Práctica*, 1st ed. 1991.

[58] Siemens, "Project Planning Manual SIPART DR20." p. 136, 1990.

[59] A. B. Corripio, *Tuning of Industrial Control Systems*, Third Edit. ISA, 2001.

[60] G. N. Stephanopoulos, *Chemical Process Control: An Introduction to Theory and Practice*, 1st ed. Prentice Hall, 1984.

[61] A. Creus Solé, *Instrumentación Industrial*, 6th ed. España: Marcombo, 1998.

[62] K. Ogata, *Modern Control Engineering*, 5th ed. Pearson, 2010.

[63] A. Cornet and S. Ledru, "Javasci," 2011. [Online]. Available: http://help.scilab.org/docs/5.3.2/en_US/javasci.html. [Accessed: 01-Jun-2013].

[64] Universidad del Valle, "Oficina de Informática y Telecomunicaciones - OITEL," 2014. [Online]. Available: http://oitel.univalle.edu.co/. [Accessed: 21-Jul-2015].

[65] R. Ledesma, "AlphaCI: un programa de cálculo de intervalos de confianza para el coeficiente alfa de Cronbach," *Psico-USF (Impresso)*, vol. 9, no. 1, pp. 31–37, 2004.

[66] Simutech Multimedia Inc, "Troubleshooting Skills Training System," 2014. [Online]. Available: http://www.troubleshootingskills.com/. [Accessed: 10-Mar-2015].