



**Desarrollo de una plataforma virtual web
como apoyo a la enseñanza del control de sistemas dinámicos**

Carlos Arturo Moreno Susatama

Universidad Pedagógica Nacional

Facultad Ciencia y Tecnología

Departamento de Tecnología

Licenciatura en Electrónica

Bogotá D.C

2020

**Desarrollo de una plataforma virtual web
como apoyo a la enseñanza del control de sistemas dinámicos**

Carlos Arturo Moreno Susatama

Trabajo de grado presentado como requisito parcial para optar al título de:

Licenciado en Electrónica

Directora:

Ing. Carol Ivonn Rodríguez Feliciano, M.Sc.

Universidad Pedagógica Nacional

Facultad Ciencia y Tecnología

Departamento de Tecnología

Licenciatura en Electrónica

Bogotá D.C

2020

“En la época de mi abuelo se esperaba que un ingeniero trabajara en el mundo real diseñando máquinas, cadenas industriales, puentes y caminos. Ahora trabajaría en el mundo inmaterial, el del software, que no tiene entidad física sino que es un código, un lenguaje.”

Rosalind Williams

“Yo soy por que nosotros somos“

Ubuntu- Cultura Africana Xhosa

Agradecimientos

La elaboración de este proyecto de grado ha representado un gran esfuerzo, no solo para mí, sino también para aquellos que de múltiples formas han contribuido a que este evolucione. Si bien me he encontrado frustrado en muchas ocasiones, siempre recurro a ese comentario que una vez la maestra Claudia Rodríguez hizo en clase de pedagogía y psicología: “Si un problema no tiene solución, entonces no hay problema”, aunque cuesta mucho trabajo aceptarlo en ocasiones y parecieran existir cosas que no se pueden dejar atrás. Es decir, que a pesar de no encontrar respuestas pronto, con una constante persistencia e investigación se puede avanzar enfrentando los requerimientos que pida el camino y el desarrollo del proyecto. Particularmente le doy gracias a ella, por la cantidad de veces que cuestionó mis acciones, haciéndome dudar muchas veces de mi trabajo, siempre con el fin de formar profesionales de calidad.

Agradezco inmensamente a mi madre Ana Inés y mis hermanas: Lina y Lisbeth, que me acompañaron de manera moral, y que cuando me vieron en estados críticos que afectaron el desarrollo de este proyecto, trataron de dar lo mejor de sí para apoyarme en tan elaborado propósito dentro de la universidad.

Un fuerte saludo y mi total gratitud a cada uno de mis compañeros: Claudia López, Juan Urbina, Andrés Coronado, Sergio Duarte, Manuel Galvis, Norbey Bustamante, Sandra Aldana, Jaime Gómez Valero, Félix Ruiz; compañeros tanto de la carrera como de la Universidad Distrital, que aportaron en la validación y verificación de la plataforma *Microlablina*, y que durante mi formación fueron transmisores de experiencias, buenas y malas, que me permitieron entender la importancia de la carrera dentro de mi crecimiento como persona y profesional.

Como admiración y respeto, a aquella persona que me acompañó y me llamó la atención muchas veces por no contarle bien las situaciones, que me permitió ir moldeando mi trabajo y haciendo correcciones que agilizaran mi proceso en el proyecto, le doy gracias a la maestra Carol Ivonn Rodríguez, que más que ser mi directora de trabajo de grado, fue mi respaldo académico y un modelo a seguir.

A los maestros Hugo Marín, Luis Herrera y Vanessa Garrido por mostrarme opciones que me permitieran trabajar desde las diferentes áreas de la física y la matemática, y que incluso si perdí alguna materia con ellos, me motivaban a demostrar las capacidades que tenía para poder afrontar un reto, más que por una nota por un interés personal.

Por último y sin olvidarme de lo importantes que fueron estos tres maestros para mi formación, uno por mostrarme lo interesante de la electrónica desde la rama de las comunicaciones y los circuitos y quien en muchas ocasiones me dio recomendaciones para mejorar como estudiante, y los otros dos por mostrarme el mundo del control en su forma digital y analógica, les doy gracias a Jimmy Ramírez, Augusto Guarín y Diego Rivera.

Contenido

Agradecimientos	IV
Contenido	VII
Lista de figuras	XI
Lista de tablas	XIV
Introducción	XVII
1. El Micro-Laboratorio Virtual	1
1.1. Planteamiento del problema	2
1.2. Justificación	3
1.3. Delimitación y alcances	5
1.3.1. Sistemas a modelar	6
1.3.2. Alcances del proyecto	6
1.4. Objetivos del proyecto	7
1.4.1. Objetivo General	7

1.4.2. Objetivos Específicos	7
1.5. Marco referencial	8
2. Metodología	14
2.1. Producto a entregar	15
2.1.1. Esquema de Diseño	16
3. Diseño de Librerías y Modelo de la Plataforma	19
3.1. Programación PHP JavaScript JQuery	19
3.2. Modelo Vista Controlador MVC	21
3.3. Librerías Matemáticas y de Interfaces	22
3.3.1. Librerías de Visualización de Fórmulas Matemáticas	23
3.3.2. Planteamiento del Algoritmo de Números Complejos para una potencia de N Orden	26
3.3.3. Moldeamiento Matemático y Sistemas Dinámicos en Fun- ción de transferencia	27
3.3.4. Implementación del método de Laguerre	32
3.3.5. Librerías de Propósito General	38
3.3.6. Diagrama UML Clases	41
4. Modelos Matemáticos de la Plataforma	43
4.1. Motor de Corriente Continua	43
4.1.1. Modelo Matemático	45
4.2. Ball of Beam (Viga Bola)	47

4.2.1. Modelo Matemático	47
4.3. Aeropéndulo	50
4.3.1. Modelo Matemático	50
5. Gestión de Usuarios	53
5.1. La base de datos	54
5.2. Roles de usuario	56
5.2.1. Público	56
5.2.2. Anfitrión	57
5.2.3. Invitado	57
5.3. Sección de Preguntas	57
5.3.1. Diseño visual del gestor de preguntas	58
6. Interfaz de Usuario y Aplicaciones	62
6.0.1. Navegadores Soportados por Microlablina.	62
6.1. Pantalla de Inicio	63
6.2. Herramientas Experimentales	65
6.2.1. Funciones de Transferencia Simuladas con Diferentes Fuentes	65
6.2.2. LGR Lugar Geométrico de las Raíces	66
6.2.3. Diagrama de Bode	68
6.2.4. Experimentos Libres	68
6.3. Interfaz de modelos matemáticos de manera nativa.	72
Conclusiones y Recomendaciones	78

Conclusiones	78
Recomendaciones	80
Bibliografía	82
Apéndice A	84
A. Licencias de Terceros y Términos Legales	84
A.1. MIT	84
A.2. GNU/GPL	85
A.3. let's encrypt, uso de certificados SSL	85
A.4. MathWorks MATLAB Licencia Estudiantil, para ejecución de pruebas en ambientes académicos	87
Apéndice B	88
B. Herramientas Matemáticas	88
B.1. MÉTODO DE MULLER	88
Apéndice C	90
C. Segmentos de Códigos Fuente	90
C.1. Código Generador de Polinomios	90
C.2. Código Lazo Cerrado	92
C.3. Diagrama de Bode	94
C.4. Derivadas	98

C.5. Limites	99
C.6. Fracciones Parciales	101
C.7. Laguerre: Raíces de Polinomio	105
C.8. Discretizacion: Tustin	109

Lista de Figuras

2.1. Estructura del modelo MVC de codeigniter 3.XX	16
2.2. Esquema de MVC implementado en la plataforma de micro-laboratorio virtual	17
3.1. Modelo de Relación entre el Cliente y la página con JavaScript . .	20
3.2. Representación en bloques en lazo cerrado de un sistema $H(s)$ con un controlador $G(s)$	39
3.3. Diagrama UML de clases de las librerías desarrolladas	42
4.1. Esquema del modelo de un motor DC	44
4.2. Esquema del modelo de un motor visual del viga bola (ball of beam)	47
5.1. Pantalla de registro de usuarios nuevos	53
5.2. Pantalla de Bienvenida a usuarios registrados	54
5.3. Arquitectura de la base de datos dentro de la plataforma Microlablina	55
5.4. Ilustración de la plataforma microlablina dentro de un usuario registrado en la sección de gestor de preguntas	58

5.5. Ilustración de la plataforma microlablina dentro de un usuario registrado en la sección listado de vínculos	59
5.6. Registro de vínculos dentro de la aplicación web.	59
5.7. Ingreso del pin suministrado por el anfitrión.	60
5.8. Lista de preguntas que puede ver el invitado.	60
6.1. Esta imagen representa el estilo de diseño en el cual carga la pantalla inicio para ordenador y móvil. Esta captura corresponde a la versión 1.	64
6.2. Imagen de presentación actual de la pantalla de bienvenida, versión Beta Final.	64
6.3. Pantalla de presentación de herramientas junto con su descripción según la utilidad de cada una.	66
6.4. Simulador de función de transferencia con otras fuentes	67
6.5. Muestra el lugar geométrico de las raíces, sección en modo desarrollo	67
6.6. muestra el diagrama de bode de la función digitada	69
6.7. Interfaz antes y después de digitar la función de transferencia . . .	69
6.8. Resultados de la función de transferencia en impulso y escalón. . .	70
6.9. Resultados de la función de transferencia ante la rampa y al calcular las variables de estado.	70
6.10. Sección de lazo cerrado, plantilla explicativa	71
6.11. Ingreso de funciones de transferencia a la plataforma, para realizar la unión de lazo cerrado	72
6.12. Lista de modelos instalados en la plataforma	73

6.13. Panel de bienvenida de cada uno de los modelos nativos dentro de la plataforma.	73
6.14. Resultado del modelo en lazo abierto computado después de dar en el botón "ver solo sistema".	75
6.15. Sección de gráficas ante los diferentes estímulos de entrada de un sistema seleccionado el lazo abierto	75
6.16. presentación la plataforma de bienvenida para manejar las variables de los modelos nativos.	76
6.17. presentación la plataforma de bienvenida para manejar las variables de los modelos nativos.	76

Lista de Tablas

3.1. Tabla de la librería <code>Stringlatex</code> que convierte de cadena de fórmulas a latex.	23
3.2. Esta tabla muestra la configuración de métodos de la clase <code>text-box2tff</code> , según la construcción y el diseño de la librería para php. .	24
3.3. Esta tabla muestra una clase que permite trabajar complejos dentro de PHP. Esta clase es importante pues la mayoría de datos numéricos que se trabajan en el aplicativo son de este tipo.	25
3.4. Esta tabla muestra como la clase <code>bode</code> opera para entregar un gráfico de puntos relación magnitud-fase y magnitud-frecuencia al usuario final.	30
3.5. La librería <code>Polinomio</code> permite crear polinomios de un arreglo de caracteres, Funciona como un nanocompilador que permite compilar los datos ingresados por el usuario y entregarlos para ser operados dentro del motor PHP.	31

3.6. Esta librería es creada a partir del método numérico de Laguerre. Con el fin de crear raíces de polinomios es adaptada a PHP, su código original fue escrito en C. 32

3.7. Esta tabla permite crear límites de polinomios tanto complejos como reales desde que no tenga denominadores, es importante para el cálculo de bode y calcular fracciones parciales 36

3.8. Esta es la librería principal, la TF o transfer function se encarga de compilar y trabajar el resto de elementos matemáticos que ingresa el usuario. 37

4.1. Tabla de parámetros físicos del motor eléctrico. 45

4.2. Tabla de parámetros físicos de un viga bola. 48

4.3. Tabla de parámetros físicos Aeropendulo. 51

5.1. Tabla explicativa de los métodos existentes en la base de datos . . 55

6.1. Esta tabla muestra el funcionamiento de la página web en los navegadores más usados hasta febrero de 2019. 63

Introducción

Este trabajo de grado aplica herramientas matemáticas básicas de sistemas de control usando tecnologías de programación como: HTML5, PHP y Javascript, para ser ejecutadas desde un navegador, tanto en dispositivos móviles como computadores de mesa.

Con el auge del internet y la intención de manejar la mayor cantidad de información posible desde este medio usando múltiples plataformas, se hace indispensable programar aplicativos que sean manejables desde cualquier sistema operativo o equipo celular, preferiblemente que sean a bajo costo o libres y gratuitos. Es de mencionar, que la mayor parte de las aplicaciones para desarrollo matemático en la web no son libres, aunque tienen características de uso para el público en común.

Debido a que la mayor parte de la población de la universidad usa dispositivos móviles con diferentes recursos de hardware, resulta un poco compleja la construcción de una aplicación móvil que garantice que este tipo de cálculos que se adapte a cualquier dispositivo. Matlab, por ejemplo, es una excelente herramienta

para ejercicios de control, pero requiere el pago de la licencia para su uso, además de requerir espacio de memoria para su instalación y ejecución junto con un buen procesador.

Teniendo en cuenta las ventajas que tienen los lenguajes de programación antes mencionados se realizó la construcción de una plataforma, que permite trabajar sistemas en términos de su transformada de **Laplace** desde cualquier dispositivo, permitiendo consultas rápidas sin tener que instalar ningún programa o aplicación. Adicionalmente, se garantizó que el usuario pueda interactuar con los parámetros de la plataforma y llegar a deducciones que faciliten la interpretación de un sistema de control simple.

Para lograr la finalización de este aplicativo se implementó la metodología scrum ágil, que facilita la construcción de proyectos grandes.

Este proyecto tiene como finalidad la construcción de un micro-laboratorio virtual, que le permita al estudiante, como actividad elemental dentro del estudio de los sistemas de control, visualizar características de los modelos de sistemas en términos de función de transferencia.

Capítulo 1

El Micro-Laboratorio Virtual

La pregunta central y que motivó la construcción de este laboratorio virtual es: ¿Es posible contar con una herramienta de software libre online para facilitar al estudiante la comprensión de contenidos en el área de control? Siendo así, se pretendió con este trabajo desarrollar dicha herramienta, y para su construcción se contó con conocimientos en programación y organización de contenidos, además de reconocer los problemas propios del área de Sistemas de Control.

En la mayoría de casos, los estudiantes no comprenden los contenidos presentados por diferentes distractores de su vida cotidiana, a lo que se suma la dificultad del docente para transmitir la información correspondiente a la temática de control usando ejemplos prácticos y visibles al estudiante. Esta es una desventaja no solo para el docente del espacio académico, sino también para el maestro en formación al que le cuesta trabajo entender los contenidos asociándolos a construcciones físicas reales.

Usando este planteamiento, se decide construir un laboratorio virtual, pensando como posible escenario la Universidad Pedagógica Nacional y los estudiantes del área de control de la Licenciatura en Electrónica, donde uno de los requisitos para que el educando apruebe la asignatura es entender el concepto de sistema dinámico y las diferentes formas de estabilizarlo. Se tomó este tema para el laboratorio al presentar gran dificultad para quienes cursan esta asignatura.

1.1. Planteamiento del problema

El problema que se observa en el contexto de las asignaturas de Sistemas de Control es: la presencia de inconvenientes al momento de simular los contenidos teóricos. Aunque la Universidad Pedagógica Nacional presenta alternativas para que los docentes en formación asimilen los contenidos de las áreas del saber disciplinares hay pocas herramientas virtuales instaladas en los laboratorios, o bien son de carácter privado y solo se cuenta con licencias de uso temporal. Esto se traduce en dificultad para la visualización dinámica de un sistema físico y ocasiona que la información no se interiorice de la mejor forma, lo cual da como resultado que a la hora de implementar el montaje real se encuentren falencias y vacíos conceptuales de las temáticas teóricas tratadas sin una adecuada visualización de la dinámica de los sistemas físicos.

Por tanto, con el ánimo de ayudar al estudiante a interiorizar la importancia de

los modelos matemáticos para la identificación de sistemas dinámicos, se propone realizar una herramienta virtual de apoyo al docente y al estudiante en el proceso de enseñanza-aprendizaje de sistemas de control automático, de modo que el estudiante pueda acceder a ella permitiéndole comprender sistemas dinámicos de manera ideal. En este espacio el estudiante puede realizar algunas prácticas de control, usando modelos físicos virtuales y enfocándose solo en ver cómo se comportan las ecuaciones del sistema que desee simular, en lazo abierto y lazo cerrado con controlador.

Se propone entonces una herramienta virtual de apoyo en el proceso de enseñanza-aprendizaje de sistemas de control automático, de modo que el estudiante pueda acceder a ella de manera gratuita para la comprensión de temas básicos de sistemas dinámicos. El estudiante puede realizar algunas prácticas de control usando modelos físicos virtuales o directamente la función de transferencia, y observar el comportamiento del sistema que desee simular, ya sea en lazo abierto, lazo cerrado o con la implementación de controladores como PID.

1.2. Justificación

Como parte del plan de estudios de la Licenciatura en Electrónica ofertada por la Universidad Pedagógica Nacional y la Facultad de Ciencia y Tecnología, se encuentran espacios académicos como lo son: Sistemas de comunicaciones, Prácticas

educativas, Optativas profesionales, Sistemas de control, entre otros, cuya finalidad es formar al sujeto en la electrónica y la educación. Por esto la intencionalidad de este proyecto es orientar al estudiante en una temática disciplinar desde lo pedagógico.

En los espacios académicos de nivel avanzado se necesita una herramienta didáctica que apoye los conocimientos adquiridos por el estudiante en el área de sistemas de control, pues "Los conceptos teóricos necesitan confirmarse con experiencias realizadas en laboratorios acercándose a la realidad"[Granados, 1999], lo que indica que en lo posible los conocimientos teóricos deben apoyarse de manera práctica. Se construyó entonces una herramienta que ayude al sujeto a comprender lo mejor posible y de manera didáctica la aplicación de los temas vistos teóricamente en el área de control. Esto puede fortalecer conocimientos y saberes usando las horas de trabajo independiente (HTI).

Como principal finalidad se espera motivar el interés del estudiante, es decir, que el maestro en formación se enamore del área control automático, haciendo que esto mejore sus habilidades y competencias. Contar con una herramienta didáctica, en este caso virtual, no solo facilita al educando el aprendizaje, sino al maestro la enseñanza de temáticas que requieran ejemplos específicos.

Se tuvo como fundamento investigaciones referentes al área de control en diferentes IES como la Universidad Nacional de Colombia, la Universidad Distrital

Francisco José de Caldas y la Universidad de Antioquia, que presentaban alternativas de estudio para facilitar la comprensión de temáticas enfocadas en automatización de sistemas de control, no solo de manera textual sino virtual, más aún en un contexto como el presente en medio de la actual pandemia.

1.3. Delimitación y alcances

Se pretendió desarrollar un escenario virtual, que se pueda ejecutar en navegador web para estudiantes del área de control, donde ellos puedan simular un sistema dinámico de manera matemática y logren fortalecer conocimientos a la hora de reconocer las variables que afectan el comportamiento de una planta física. Para ello es menester considerar los siguientes elementos:

1. Se requirió contar con una serie de modelos matemáticos que representen un sistema real.
2. Las plantas se manejan con moldeamiento matemático dentro de la herramienta virtual. Si el docente o maestro del curso de control desea implementarla en la vida real, estas puedan ser usadas como ejemplo para contrastar lo real de lo idealmente modelado.

1.3.1. Sistemas a modelar

Se consideran tres tipos de plantas: modelo de posición de un motor, el modelo de ball of beam, y el modelo del sistema de posición de un Aeropéndulo. Para cada una de ellas se visualizan las gráficas de comportamiento ante la implementación de alguno de los tipos de controladores, los cuales son: control PID y control en el espacio de estados, según resultó conveniente en desarrollo del software.

Se desarrolló el laboratorio bajo una plataforma web usando programación: PHP, Jquery, entre otras, que permiten una interacción amigable para el usuario final.

1.3.2. Alcances del proyecto

La construcción de este laboratorio mediante software no incluye implementaciones físicas. Tampoco se evalúa la implementación con estudiantes en el aula. Esto queda en libertad para otros escenarios y extensiones posteriores a la realización de este proyecto, de acuerdo con las características a manejar por docentes que lo deseen implementar.

1.4. Objetivos del proyecto

1.4.1. Objetivo General

Desarrollar una plataforma virtual que facilite la contextualización en temáticas de control automático en la licenciatura en Electrónica, permitiendo realizar el modelado, análisis y simulación de los sistemas dinámicos.

1.4.2. Objetivos Específicos

1. Construir algoritmos de programación, que permitan la manipulación de parámetros característicos de cada sistema según el caso, mediante la plataforma.
2. Modelar matemáticamente los sistemas: viga bola, posición de un motor y posición de un Aero-péndulo, junto con los tipos de controladores: PID y control en el espacio de estados, para implementarlas en la plataforma web dependiendo el tipo de sistema y sus características, según sea conveniente.
3. Habilitar un espacio en la plataforma, donde se permita al docente cuantificar los conceptos aprendidos por los estudiantes mediante preguntas tipo test, formuladas por el educador.

1.5. Marco referencial

En la actualidad existen diversas formas para modelar e identificar sistemas. Una de ellas consiste en el uso de la herramienta **MATLAB**, este es un programa de escritorio y que posee herramientas web de uso privado; también maneja herramientas móviles, pero sin una licencia pagada su uso es limitado. **WÓLFRAM ALPHA** es una herramienta web que funciona para resolver o trabajar ecuaciones o modelos matemáticos, también de uso privado. Por otra parte **CircuitLab** es una plataforma virtual web orientada a la simulación y modelado gráfico de circuitos, aunque hay recursos gratuitos también requiere pago y no es netamente libre su uso.

Una de las desventajas de las plataformas nombradas es el costo y sus respectivas licencias. Adicionalmente en algunos casos, como en el uso de **MATLAB** para realizar una identificación sencilla mediante comandos, es necesario un conocimiento amplio de los comandos y un buen ordenador. Al dejar la tarea de linealización e identificación a programas de este estilo, se evita en algunos casos que el individuo entienda el proceso realizado internamente para representar o simbolizar el sistema en una ecuación.

Adicionalmente se ha estado trabajando en laboratorios virtuales por otras universidades, en temas enfocados en educación y enseñanza en plataformas web o que involucran Internet, o simplemente dentro del aula como herramienta de escritorio.

Mencionaremos a continuación 3 desarrollos que sirvieron de contexto al presente anteproyecto.

Laboratorio Virtual de Control Inteligente

Esta publicación presenta un proyecto para simular el control de sistemas mediante lógica difusa e inteligencia artificial; fue programada con recursos de java, enfocado para ser un aplicativo de escritorio. Para posteriores trabajos proponen implementar el programa, como parte de una plantilla web.

“ –Este artículo presenta un laboratorio virtual de control inteligente desarrollado con Easy Java Simulations y diseñado para tres de las plantas existentes en la Universidad Autónoma de Occidente, Cali (Colombia): Planta de nivel Lab-Volt, Servomotor SRV02 Quanser y sistema Bola Viga Quanser. Las estrategias de control implementadas en el laboratorio son control difuso, control con redes neuronales artificiales y control PID, este último con el fin de permitir al usuario comparar resultados entre control inteligente y control clásico. Esta herramienta de enseñanza en control inteligente permite a los estudiantes acercarse al mismo usando una plataforma interactiva y de fácil uso.–“

[Delgado G, 2009].

Una de las características de este trabajo es su enfoque hacia la enseñanza por computadora, usando las *TIC* como pretexto en la construcción de herramientas educativas. Sin embargo, el objetivo del proyecto está basado en usar un software para fines de aprendizaje de lógica difusa y redes neuronales *RNA*, la interfaz

presenta una apariencia cómoda para el usuario y aparentemente intuitiva para personas que tienen un conocimiento detallado de sistemas de control basado en inteligencia artificial y modelado algorítmico.

Como parte de las conclusiones de su trabajo los autores resaltan: "Los laboratorios virtuales como instrumentos modernos que permiten a los estudiantes consolidar conceptos y fusionar la teoría con la práctica de una manera sencilla y no presencial"[Delgado G, 2009], indicando que es útil para el estudiante contar con herramientas que le permitan apropiarse de contenidos de manera autónoma y por medio de la exploración.

Laboratorio Virtual para Prácticas de Control por Computador

“ Este trabajo describe un laboratorio virtual desarrollado para dar soporte a las prácticas de Control por Computador de la ETSEIB. El laboratorio está totalmente desarrollado mediante Easy Java Simulations (EJS)."[Vallv and Costa, 2007]. Este documento está basado en control discreto, y se propone como apoyo al docente en el área de control. Se utiliza como sistema de un motor y varios engranajes con un sensor para ajustar la posición de los piñones; este sistema está conectado a una tarjeta de adquisición de datos, que junto al programa diseñado por java pueden simular con controlador PID discreto, y funciona como ayuda presencial para el estudiante a la hora de realizar sus prácticas.

La aplicación está diseñada en *java* junto a las librerías de desarrollo de *MATLAB*. Esta integración permite usar elementos y funciones para lograr una simulación en tiempo real y procesamiento de datos, de tal forma que al momento de usar las funciones del programa *MATLAB*, este ejecute los cálculos y muestre las gráficas del sistema modificando los valores que involucran el controlador. Cuenta con cuatro prácticas presenciales bajo el mismo sistema, aunque es semi-virtual y permite la manipulación de hardware desde el software, pero no permite la integración con otros modelos y queda como uso e implementación para laboratorios próximos de control digital.

Desarrollo básico de un Laboratorio Virtual de Control de Procesos basado en Internet

Esta plataforma virtual corre en navegadores que usan HTML2, diseñado con recursos de la época para servidores. Cuenta con una serie de controles que permiten la manipulación de datos de un sistema ya modelado y configurado para un tipo de controlador específico. Este aplicativo se propone como alternativa a herramientas como *LabView* y *java* que son de gran utilidad en la educación para ingenieros.

“–En este artículo se presenta el desarrollo básico de un Laboratorio Virtual de Control de Procesos (LavConPro) a través de Internet, basado en la filosofía cliente/servidor y desarrollado conjuntamente entre la Universidad Nacional Experimental del Táchira (UNET) y la Universidad de Los Andes-Táchira (ULA-Táchira). Los Laboratorios Virtuales son ambientes que permiten la realización de experiencias y prácticas sin necesidad de presencia física en los laboratorios reales, con las ventajas académicas, operativas y económicas que esto implica. El área del Control de Procesos es una de las más activas en el uso de esta emergente herramienta institucional. En esta primera etapa se pretende determinar su factibilidad operativa mediante la realización de prácticas para el estudio dinámico de un sistema de control remoto de un proceso típico: un sistema de nivel de líquido.– “

[Ricardo et al., 2005]

La plataforma web es ejecutada mediante un navegador web, y su presentación aparenta ser un gran diseño. Está capacitada para controlar el caudal de líquido que cae a un tanque de manera remota, con un muestreo de datos (envío y recepción) de 1.25 muestras por segundo. Cuenta con dos tipos de prácticas: La primera es la identificación del sistema a partir de la adquisición de datos; el aplicativo recolecta los datos y se los muestra en forma de tabla, mientras el usuario deberá variar los valores que se le pide para identificar el sistema y ajustarse al modelo real. La segunda es el control en lazo cerrado, donde el estudiante deberá ajustar el sistema según los criterios que este demande y el docente establezca para su correcto funcionamiento. Aunque la aplicación mejora el proceso de enseñanza y aprendizaje y permite la interacción activa del estudiante, no se encarga de hacer

un modelado a partir de las ecuaciones ideales de la misma, pues la función es que entiendan como trabajan los controladores y muestre la gráfica del sistema controlado y en lazo abierto.

Capítulo 2

Metodología

Este proyecto de grado se realizó a partir de la metodología SCRUM ÁGIL, para el desarrollo de software bajo los siguientes criterios: Planificación, Ejecución, Seguimiento y Control, Cierre.

Actividades iniciales: Planificación y ejecución

1. Construcción de la plantilla web del usuario, que permita trabajar los algoritmos matemáticos creados mediante programación para simular la planta dentro de la plataforma. Esta actividad incluyó la adquisición de conocimientos sobre la implementación de ecuaciones matemáticas matriciales dentro de un lenguaje de programación de computadores, garantizando su funcionamiento en el navegador web independientemente del dispositivo.
2. Creación de una entidad de relación que facilite la interacción entre los usuarios según su rol, a la vez que pueda el programador entender los modelos que se requieren para organizar el código fuente. Esto implica diferenciar entre el

tipo de persona que usará la plataforma y los comandos que se requieren para que el usuario final interactúe con ella.

3. Manipulación de una base de datos que almacene tanto los usuarios y sus roles, como los resultados obtenidos en el laboratorio virtual web.

Control y seguimiento

En esta etapa se establecieron una secuencia de pasos que permitieran identificar el progreso del proyecto, con resultados cuantificables dependiendo de la creación de los hitos y los sprint que se requieren para el desarrollo de la plataforma.

Cierre

Se realiza la entrega del proyecto con pruebas técnicas de funcionamiento y soporte, donde un sujeto puede acceder a la plataforma de manera remota mediante la url, adicionalmente se entrega la documentación en la que se detallan los pasos requeridos para la realización de la aplicación web en modo de laboratorio virtual. [Brooks, 2011].

2.1. Producto a entregar

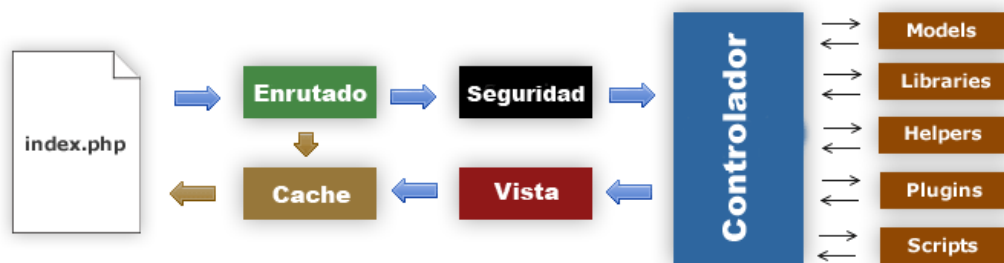
El producto final de este trabajo, es un laboratorio virtual online, que además de contar con tres prácticas de control ideal simuladas por software, tendrá una sección de preguntas donde el estudiante podrá ingresar para resolver una serie de preguntas

elaboradas por un docente con el fin de medir el desempeño del estudiante en la asignatura. La plataforma está diseñada para que los estudiantes puedan utilizar tanto el sistema modelado como el sistema controlado.

2.1.1. Esquema de Diseño

El producto se desarrolló en dos etapas principales pensadas en la interacción del usuario con la plataforma y en la solución de problemas matemáticos dentro de la página web, donde en la mayoría de los casos las operaciones y procedimientos matemáticos son ejecutados dentro del servidor.

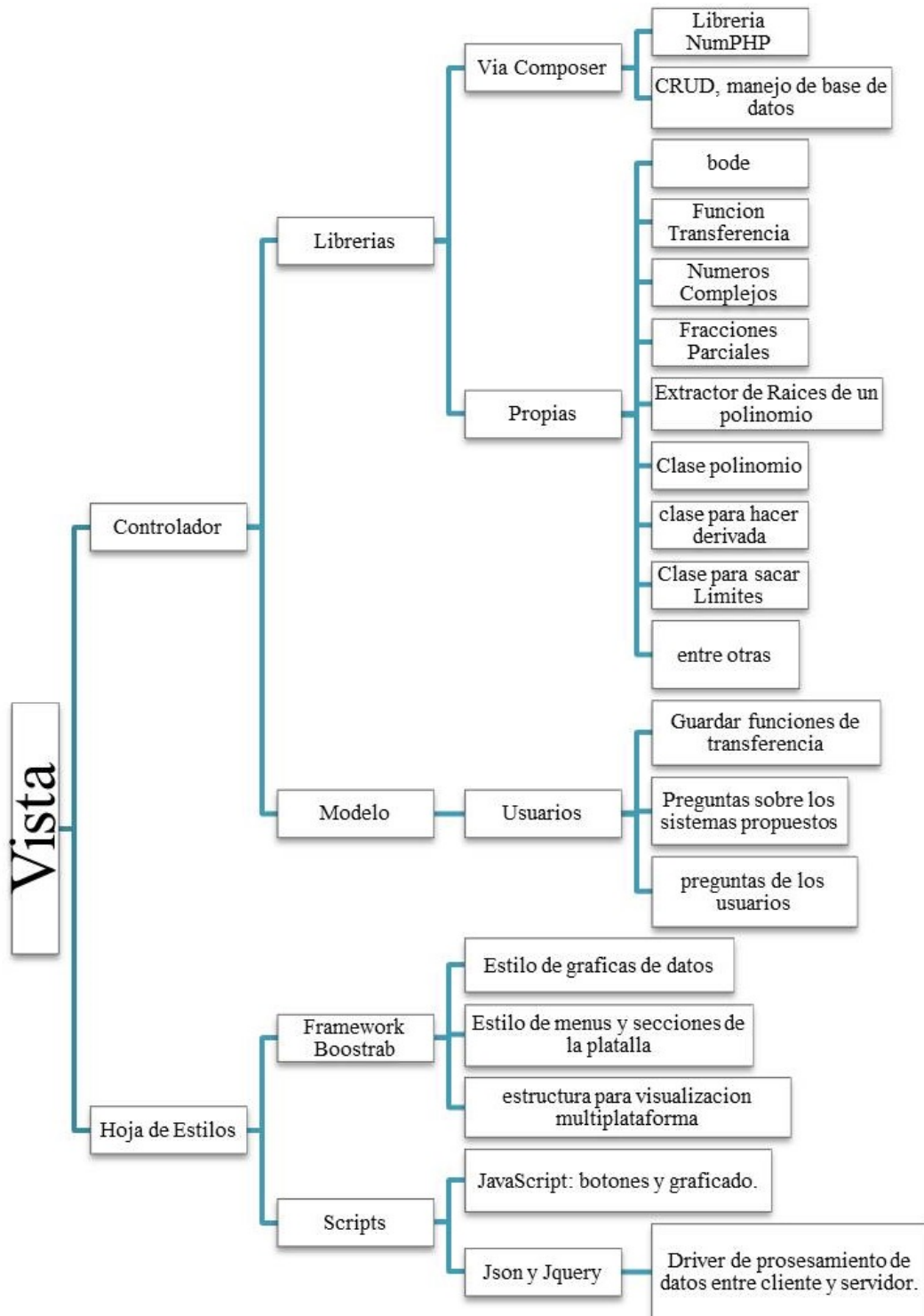
Figura 2.1: Estructura del modelo MVC de codeigniter 3.XX



La estructura en la que se desarrolla el programa es basada con el framework **Codeigniter 3.6**, esta plataforma esta basada en programación orientada a objetos y funciona bien con PHP-5 o superiores. En la figura: 2.1, se puede apreciar la estructura lógica de trabajo. Adicionalmente, por la forma en la que está organizada, la arquitectura del framework permite usar la metodología SCRUM AGILE debido a que se puede centrar en el **Front-end** (diseño y apariencia) separado de la parte

lógica **Back-end**.

Figura 2.2: Esquema de MVC implementado en la plataforma de micro-laboratorio virtual



Los criterios de programación basados en MVC se centran en la construcción de clases específicas: Modelo, Vista, Controlador. Cada una de estas etapas, hacen que el proyecto se despliegue en sectores. El mapa de la figura 2.2 muestra la propuesta estructural para el desarrollo del programa, y de esta manera: crear, encontrar y corregir fallas en las en las librerías que se desarrollaron para hacer procedimientos matemáticos e interactuar con la base de datos y la interfaz gráfica

Capítulo 3

Diseño de Librerías y Modelo de la Plataforma

Para este trabajo de grado se requieren dos saberes importantes, uno de ellos es el desarrollo de aplicaciones web, y el otro se basa en el comportamiento y modelado de un sistema dinámico para la aplicación de elementos de control. En ese sentido, se cuenta con conceptos previos que contribuyen a la creación del proyecto. En este capítulo el lector podrá informarse acerca de los sistemas y plantas a trabajar, así como de elementos web que permiten desarrollar herramientas matemáticas para la elaboración de aplicativos web.

3.1. Programación PHP JavaScript JQuery

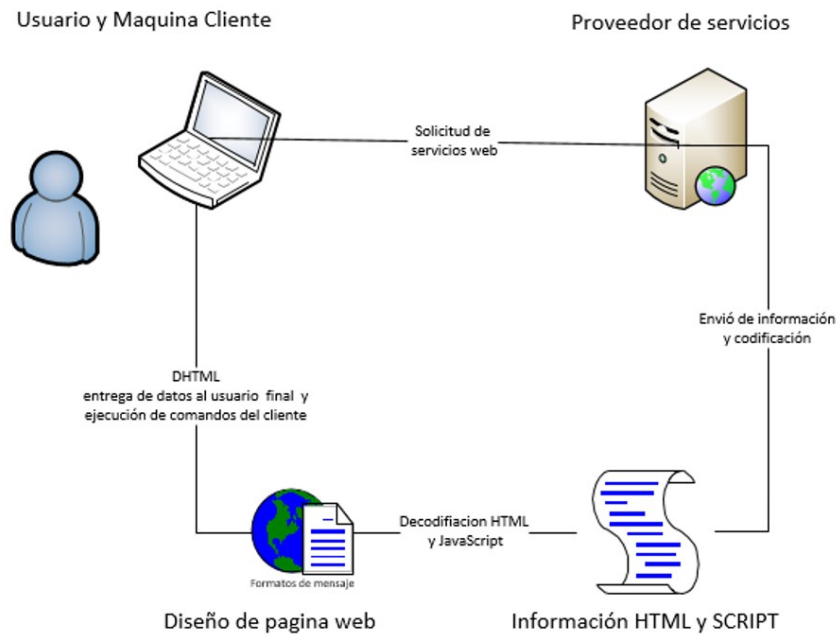
Para este trabajo fue importante conocer los lenguajes de programación web, qué es un hosting y servidor web, y recordar que existen diferentes tipos de servidor dedicado a temas específicos. En el caso de este proyecto están destinados a servicios

educativos con recursos de compilación matemática, o que permitan su creación y base de datos.

JavaScript

Este Lenguaje de programación es una herramienta útil para el desarrollo de páginas web, ya que cuenta con herramientas matemáticas como cualquier lenguaje de programación. Cuenta con un numeroso grupo de ayuda y soporte, y **DHTML** permite que la página sea más dinámica, esto quiere decir que puede hacer operaciones medianamente complejas desde el ordenador sin requerir servicios del servidor.

Figura 3.1: Modelo de Relación entre el Cliente y la página con JavaScript



Para este trabajo se usa este lenguaje como puente entre los datos enviados por

el servidor y el cliente. El servidor envía la solución de un sistema de ecuaciones diferenciales, en términos de ecuaciones en diferencias para poder trabajarlos con distintas señales de entrada, esto mejora la accesibilidad y velocidad de la página ahorrando trabajo de procesamiento al servidor.

Este lenguaje al ser diseñado para ser usado en paralelo con HTML, se ve sujeto a mejoras, y posee gran familiaridad con los navegadores actuales incluyendo dispositivos móviles. "JavaScript es regularmente utilizado para hacer extensiones para diferentes programas, como los scripts codificados"[Brooks, 2011]. Como se muestra en la figura: 3.1 se puede utilizar para construir aplicaciones con los datos que son enviados directamente con el servidor, haciendo puente entre los protocolos de comunicación de los servidores.

3.2. Modelo Vista Controlador MVC

En este proyecto se usan características de MVC y UML, que permiten que el código sea organizado, y que si hay librerías creadas por otros desarrolladores puedan ser citadas y llamadas con un estándar específico sin quebrantar los derechos de autor que le corresponde a cada creador de código fuente.

La metodología MVC, para la construcción de código fuente, para aplicativos web, consiste en una organización de contenidos en modo vista, modelo, controlador [Myer, 2008]. El controlador es la sección donde se especifican las caracte-

rísticas importantes del programa a construir, como las librerías que en este caso se encargarán de modelar matemáticamente las ecuaciones diferenciales. La vista simplemente se encarga de la visibilidad y la interfaz del usuario, pretendiendo que el individuo interactúe de manera didáctica y amigable con las herramientas que ofrece el programa. El modelo es la relación entre el usuario y el programa de manera algorítmica, un puente de comunicación entre todos los comandos, librerías y la interfaz del usuario.

UML o lenguaje de modelado unificado, es un estándar que se usa para lograr desarrollar aplicaciones de gran nivel de impacto. Esta metodología estudia los casos de uso para poder modelar las librerías y sus exigencias; esta técnica permite al programador y a los diseñadores construir un ambiente virtual amigable y sólido. Para el caso del trabajo se usa para brindar a los estudiantes un entorno virtual agradable a la vista y a los contenidos que debe interiorizar.

3.3. Librerías Matemáticas y de Interfaces

La construcción de las librerías para esta aplicación se hace importante, ya que PHP y JavaScript no portan complementos matemáticos que faciliten el modelado de sistemas en términos de Laplace. Aunque existen librerías en GIT que aportan al tema, no son suficientes para la construcción del presente proyecto, por tanto se construyeron unas nuevas desde los temas trabajados en la universidad y otras basadas en modelos de regresión matemática.

3.3.1. Librerías de Visualización de Fórmulas Matemáticas

Librería Stringlatex

La librería que se muestra en la tabla 3.1 permite al usuario final visualizar fórmulas que están escritas en forma de polinomio. Se construyó en base a la codificación \LaTeX e implementa código de \TeX . Internamente usa un compilador llamado mathjax . En la misma tabla se muestra los diferentes métodos y parámetros dentro de la librería que nos permiten la manipulación de elementos dentro de la plataforma.

Tabla 3.1: Tabla de la librería Stringlatex que convierte de cadena de fórmulas a latex.

Nombre de la clase	Stringlatex	Autoría	si	Tipo Clase	Statica	Lenguaje	Php-4 o sup
Descripción							
Permite entregar al cliente un resultado más ameno, natural y gráfico de las ecuaciones que entregan cada una de las librerías matemáticas creadas							
Método o Atributo	Tipo de Método o Atributo	dato de entrada	dato de salida	Descripción			
latexpoli	Method public static	Polinomio array unidimensional	String tex	Convierte un array con valores numéricos no complejos y en forma de polinomio en una cadena con comandos tex			
latexmeltipol	Method public static	Polinomios array unidimensionales	String tex	Convierte muchos polinomios de entrada como tipo array y los muestra como la multiplicación de los números con comandos tipo tex			
latexPrinTf	Method public static	num=array den=array fden=Int	String text_	Muestra la función de transferencia al cliente, envía una cadena tipo tex			
mostrarpolinomio	Method public static	array unidimension polinomio	String tex	Imprime el objeto polinomio en términos de comandos tex			
ImpRoos	Method public static	array unidimensiones Polos Reales	String tex	Imprime los polos de una función de transferencia en términos de string con comandos tex			
ImpRoosC	Method public static	array unidimensional Polos Complejos	String tex	Imprime las raíces de un polinomios en términos de código latex para que el usuario final pueda ver el contenido como una fórmula.			

Librería `textbox2tff` La librería `textbox2tff` incorpora un minicompilador diseñado para transformar texto en código de programación interno. Esta se encarga de verificar los datos ingresados por el usuario y transformarlos en una función de transferencia, calculando y delegando funciones a las diferentes instancias de la plataforma. En la tabla 3.2 se muestran las diferentes propiedades que posee esta librería.

Tabla 3.2: Esta tabla muestra la configuración de métodos de la clase `textbox2tff`, según la construcción y el diseño de la librería para php.

nombre de la clase	textbox2tff	Autoria	si	Tipo Clase	Publica	Lenguaje	Php
Descripción							
Convierte en arreglo los datos ingresados por el cliente y/o denominador los convierte en un arreglo llamado polinomio.							
Método o Atributo	Tipo de Método o Atributo	dato de entrada	dato de salida	Descripción			
num	Atributo privado			Guarda la cadena del numerador			
den	Atributo privado			Guarda la cadena del denominador			
factorizadord	Atributo privado			Convierte el denominador en objeto Raices2poligrup			
factorizadum	Atributo privado			Convierte el numerador en objeto Raices2poligrup			
compotexto2arreglo	Public Metodo	num string den setring	null	Recoge el den de los campos de texto y los traduce al servidor			
descompomer	Private Metodo	Cadena string	array Unidimensional	Descompone la cadena en partes y entrega un arreglo			
arpolos	Public metodo	null	array Unidimensional	Entrega el polinomio del denominador como array			
arceros	Public Metodo	null	array Unidimensional				

Librería Complejo

Aunque existen diferentes librerías externas que trabajan con números complejos, esta tiene la novedad de trabajar con complejos a la n potencia¹, esta incorpora el triángulo de Pascal para elaboración de multiplicaciones que requieren números

¹Ver la sección de anexos sobre números complejos, la lógica del algoritmo de i^n se muestra junto con un ejemplo.

complejos y conjugados dentro de un polinomio. En la tabla 3.3, se muestran las diferentes características que se diseñaron para las implementaciones dentro de la plataforma.

Tabla 3.3: Esta tabla muestra una clase que permite trabajar complejos dentro de PHP. Esta clase es importante pues la mayoría de datos numéricos que se trabajan en el aplicativo son de este tipo.

Nombre de la clase	Complejo	Autoría	Si	Tipo Clase	static	Lenguaje	Php
Descripción							
Convierte a un tipo de número complejo, hace operaciones como sumar complejos, restar, multiplicar y obtener la potencia, extraer la parte real e imaginaria, convierte a forma polar.							
Método o Atributo	Tipo de Método o Atributo	dato de entrada	dato de salida	Descripción			
r	Atributo público	NA	NA	Guarda la parte real del complejo			
i	Atributo público	NA	NA	Guarda la parte imaginaria del complejo			
_construc	Public method	Numero real, Numero Imaginario	null	Crea el objeto complejo			
tpascal	Public method	Entero orden	arreglo	Entrega el arreglo de la fila según el orden deseado, bajo los criterios del triángulo de pascal ya que permite elevar un número complejo a la n .			
powC	Public method	objeto complejo int: grado deseado	objeto complejo	Eleva un número complejo a la n			
add	Método Público	2 objetos complejos	objeto complejo	Suma dos números complejos			
sub	Método Público	2 objetos complejos	objeto complejo	Resta dos números complejos			
mul	Método Público	2 objetos complejos	objeto complejo	multiplica dos números complejos			
conjg	Método Público	objeto complejos	objeto complejo	Entrega el conjugado del número complejo			
div	Método Público	2 objetos complejos	objeto complejo	División de dos números complejos			
abs	Método Público	objeto complejos	float	Entrega la magnitud de un complejo			
sqrtz	Método Público	objeto complejos	objeto complejo	Entrega la raíz cuadrada de un número complejo			
tostring	Método Público	null	null	Entrega el número complejo visible al cliente			

3.3.2. Planteamiento del Algoritmo de Números Complejos para una potencia de N Orden

Usando el análisis del triángulo de Pascal se construye un algoritmo, de tal forma que se considera la ecuación 3.1 como una asociación de dos números dentro del programa, como reales que representan polinomio, donde $[a]$ y $[b]$ son variables que contienen números reales y se denotan con las letras griegas $[\alpha] = a$ y $[j\beta] = b$.

$$(a + b)^n \rightarrow (\alpha \pm j\beta)^n \quad (3.1)$$

Una representación general de la secuencia de $\sqrt[n]{-1}$ será:

$$Z_{num} = i^{m(mod)4} \quad (3.2)$$

La ecuación: 3.3 ilustra un ejemplo de un número complejo elevado a la 4.

$$(a \pm b)^4 = a^4 + 4a^3b + 6a^2b^2 + ab^3 + b^4 \quad (3.3)$$

En el caso de un numero complejo se extrae el valor imaginario b , y se aplica la ecuación: 3.2, creando una lista de valores para la parte imaginaria como se muestra en las ecuaciones: 3.4, donde se puede apreciar el valor de la parte imaginaria de un numero complejo según a la potencia que sea elevada. De esta lista sale una función de la ecuación: 3.2, como parámetro que se usa para la construcción central

del método de POWC dentro de la librería complejo.

$$\begin{aligned}
 i &= \sqrt{-1} = i \\
 i^2 &= -1 = -1 \\
 i^3 &= -\sqrt{-1} = -i \\
 i^4 &= 1 = 1 \\
 i^5 &= 1(-1)(-i) = i \\
 i^6 &= (-1)(-1)(-1) = -1 \\
 i^7 &= (-1)^2(-1)^2 - i = -i \\
 i^7 &= (-1)^2(-1)^2 - i = -i \\
 i^8 &= \dots = 1
 \end{aligned} \tag{3.4}$$

3.3.3. Moldeamiento Matemático y Sistemas Dinámicos en Función de transferencia

Teniendo en cuenta que toda función de transferencia suele representar el comportamiento dinámico de un sistema, bien sea antes o después de una linealización, se puede emular su trabajo en una plantilla virtual. Las plantas² pueden tener tantas salidas como se quieran controlar, así como las entradas que se deseen manipular. Con la transformada de Laplace se espera tener un panorama detallado del sistema, como los polos, los ceros, la respuesta a diferentes frecuencias (Bode), así como los criterios de estabilidad que permiten que este funcione de manera adecuada.

²Se le denomina planta a cualquier representación de un modelo matemático equivalente, que permita la descripción del sistema a nivel real.

Contando con la función de transferencia adecuada se tiene la suficiente información de la planta, de tal forma que permite crear un modelo de control adecuado, de tal manera que permita ser plasmadas en ecuaciones modeladas.

Con lo anterior en mente, las librerías de esta sección, permitirán trabajar con estos datos y enviarle un mensaje al usuario, que le ayude a visualizar lo que le está pasando al sistema desde diferentes herramientas.

Librería Fracciones Parciales (FracPart)

Es una de las librerías más significativas de la plataforma, ya que permite separar una función de transferencia en diferentes partes, esta se basa en el Método Revisite (Solución de Fracciones Parciales). La parte funcional del código que corresponde a esta librería, se encuentra en el apéndice en la sección C.6. La implementación de esta clase depende de 3 casos en particular: cuando el polinomio del denominador tiene raíces reales repetidas, cuando tiene raíces reales no repetidas, y por último, cuando estas son complejas y conjugadas. Para todos estos casos se asumen sistemas LTI, es decir, se asumen sistemas lineales invariantes en el tiempo.

Polos No reales Repetidos:

$$H(s) = A_0 + \frac{A_1}{s + a_1} + \dots + \frac{A_n}{s + a_n} = \sum_{m \pm 1}^n \frac{A_m}{s - a_m}$$

$$A_m = \lim_{s \rightarrow a_m} (s - a_m)H(s)$$

$m \Rightarrow 1, 2, 3 \dots N$ Donde n son polos diferentes

Polos Reales Repetidos:

$$H(s) = A_0 + \frac{A_1}{s - a_1} + \frac{A_2}{(s - a_2)^2} + \dots + \frac{A_n}{(s - a_3)^n}$$

$$H(s) = \sum_{m=1}^n \frac{A_m}{(s - a)^m}$$

$$A_m = \lim_{s \rightarrow a} \left\{ \frac{1}{(n - m)!} \frac{d^{(n-m)}}{ds^{(n-m)}} [F] \right\}$$

$$F = \{(a - \alpha)H(s)\}$$

Polos Complejos No Repetidos:

$$H(s) = \sum_{m=1}^N \frac{A_m}{s - a_m} + \frac{A_{\dot{m}}}{s - a_{\dot{m}}} \rightarrow N \frac{1}{(s - 2j)(s + 2j)}$$

$$A_m = \lim_{s \rightarrow a_m} (s - a_m)(H(s))$$

Librería diagrama de Bode

Esta librería permite analizar la respuesta en frecuencia de los sistemas que se están computando. Gracias a ella, la plataforma logra identificar algunos parámetros para cumplir en los gráficos con el teorema de muestreo, haciendo más eficiente la visualización de gráficos en la plataforma. En la tabla 3.4 se muestran las características que esta posee y las funciones de cada método.

Tabla 3.4: Esta tabla muestra como la clase bode opera para entregar un gráfico de puntos relación magnitud-fase y magnitud-frecuencia al usuario final.

Nombre de la clase	Bode	Autoría	si	Tipo Clase	Public Estantar	Lenguaje	Php
Descripción							
Calcula los datos de una clase de tipo tf y los convierte en gráficos de magnitud y fase							
Método o Atributo	Tipo de Método o Atributo	dato de entrada	dato de salida	descripción			
Polifase	Public static	array num array den	string puntos gráfica "[x: , y:]"	Entrega en forma de string los datos de la gráfica del diagrama fase			
Frecuencias	Private static	array complejo polos	array float frecuencias rad/s	Calcula todas las frecuencias que afectan el graficado y los entrega en forma de arreglo			
calculador 1	Private static	arreglo de polos, arreglo frecuencias	Entrega la suma de todos los ángulos tipo float	Suma los ángulos de los polos y los devuelve en términos de un valor			
rad2deg	Private static	argumento grados	Argumento radiales	Convierte el ángulo de grados a radianes			
Calculador 2	Private static	1) polos 2) frecuencia 3) tipo o número 1 den	Entrega la suma de la magnitud de polos o ceros del sistema	Entrega la suma la magnitud de los polos y los ceros en términos de variable tipo float, Este método es de tipo privado para la clase.			
Plotbode	Public Static	array num array den	string puntos gráfica "[x: , y:]"	Este muestra los puntos de diagrama de bode en términos de variable tipo String. Es el resultado de todos los puntos calculados según la variación de frecuencias y magnitudes.			

Librería Polinomio

La librería de la tabla 3.5 permite trabajar con las características de un polinomio de n orden. Fue diseñada con el fin de molificar, sumar y restar términos de un polinomio, o directamente dos o más polinomios; adicionalmente, cuenta con funciones como normalizar polinomio, y ajusta el grado del polinomio a otro para poder calcular las raíces de dos sistemas en cascada.

Librería Raíces

Esta es la clase más importante creada para el proyecto, el Kernel de la plataforma trabaja sobre esta librería, ya que usando métodos numéricos como Laguerre y

Tabla 3.5: La librería Polinomio permite crear polinomios de un arreglo de caracteres, Funciona como un nanocompilador que permite compilar los datos ingresados por el usuario y entregarlos para ser operados dentro del motor PHP.

Nombre de la clase	Polinomio	Autoría	Tipo Clase	Público	Lenguaje	Php
Descripción						
Permite la manipulación de arreglos como polinomio; hace sumas, restas, multiplicaciones: máximo grado de polinomio potencia a la n , no opera complejos						
Método o Atributo	Tipo de Método o Atributo	dato de entrada	dato de salida	Descripción		
+Construc	Public nativo metodo	array	null	Convierte un arreglo en objeto polinomio crea el objeto		
igralader de polinomio	Private metodo	Objeto polinomio Grado deseado In	array	Agrega ceros al principio del polinomio igualándolo al polinomio mayor o al grado deseado		
normalizar polinomio	Private metodh	Objeto polinomio	Objeto polinomio	Con la funcion igualador de polinomio permite que sea posible la suma o res ta de polinomios		
sumPor	Public metodo	Objeto polinomio	Objeto polinomio	Suma el polinomio principal con otro polinomio		
ResPor	Public metodo	Objeto polinomio	Objeto polinomio	Resta el polinomio principal con otro polinomio		
Ceros	Private metodo	Int	array	Entrega una cantidad de ceros en modo arreglo		
MulPor	Public metodo	Objeto polinomio	Objeto polinomio	Multiplica el polinomio principal con otro		
Grado	Public metodo	Objeto polinomio	Int	Entrega el grado del polinomio		
toSting	Metodo publico nativo	null	null			

Müller,³ se pueden calcular polos⁴ de orden superior; la desventaja por el momento de este método es la exactitud con la que el compilador de PHP calcula los datos, pues si son números muy pequeños el polo vale 0. En la tabla 3.6, se encuentra la descripción de los métodos y atributos que usa esta.

³En los anexos se puede encontrar la descripción del método de Müller, ya que solo se usa para sistemas de primer orden.

⁴Recordar que los polos del sistema son las raíces del denominador de una función de transferencia.

Tabla 3.6: Esta librería es creada a partir del método numérico de Laguerre. Con el fin de crear raíces de polinomios es adaptada a PHP, su código original fue escrito en C.

Nombre de la clase	Raices		Autoría	Med	Tipo Clase	Statica	Lenguaje	Php-6
Descripción								
De un polinomio de n orden, existe todas las raíces posibles y devuelve el resultado usando el método numérico llamado Laguerre.								
Método o Atributo	Tipo de Método o Atributo	dato de entrada	dato de salida	Descripción				
EPSS	Consta	_____	_____	Segundo factor de interacción				
MR	Constante	_____	_____	Número de comparaciones				
MT	Constante	_____	_____	Factor de error				
MAXIT	Constante	_____	_____	Maximos iteraciones puras				
EPS	Constante	_____	_____	Primer factor de iteración				
MaXM	Constante	_____	_____	Iteraciones máximas de los cálculos				
Laguer	Metodo Estatico Publico	a= m= x=	null	Ejecuta el método de Laguerre planteado en las ecuaciones anteriores				
Fmax	Metodo Estatico Publico	a= b=	array bidimensional Raices	Muestra las raíces posibles del polinomio				
zroots	a m Polish	Metodo F Publico Estatico	array bidimensional Raices	Compara las raíces posibles con los nuevos cálculos y entrega un resultado aproximado real al cliente con un margen de error del 0,001 %				

3.3.4. Implementación del método de Laguerre

El método de Laguerre es, por lejos, el más directo de estos métodos generales y complejos, debido a que requiere cálculos computacionales que dependen de librerías como: **limiteP**, **complejo**, **derivadaP**. Cuando converge el polinomio con una raíz real salta automáticamente a un loop o bucle, donde empieza a tomar límites por la izquierda para detectar el punto a donde el sistema hace curvas; sin embargo, para polinomios con todas las raíces reales, se garantiza que converge a una raíz desde cualquier punto de partida sacando los máximos y mínimos con la librería de derivadaP. En caso de encontrar polinomios con raíces complejas, se realiza un arreglo mediante un esquema simple para superar un ciclo límite no convergente

donde entra a un bucle cuando la pendiente de la segunda derivada tiende a 0.

Un ejemplo de un polinomio que requiere este esquema de interrupción del ciclo es uno de alto grado (≥ 20), con todas sus raíces justo fuera del círculo unitario complejo, aproximadamente equidistante a su alrededor. En caso de que el polinomio cuente con raíces complejas, se convierten todos los datos del polinomio en un nuevo vector complejo para poder tener una aproximación de los coeficientes complejos.

Desarrollo matemático de la implementación del método de la librería Raíces

A continuación se encuentran los modelos matemáticos que permiten la obtención de un polinomio como se muestra en la ecuación: 3.5, siendo x el valor de los puntos de corte, y x_n las iteraciones que se acercan al valor deseado de la raíz, en el caso de la librería estos números están contemplados en su forma compleja.

$$P_n(x) = (x - x_1)(x - x_2)\dots(x - x_n) \quad (3.5)$$

Para encontrar los puntos de inflexión que indiquen los máximos y mínimos de un polinomio de n orden digitado por el usuario, se calcula la derivada del logaritmo de la diferencia de las iteraciones obteniendo el valor de G , tal como se puede apreciar en la ecuación: 3.6

$$\frac{d \ln(P_n(x))}{dx} = +\frac{1}{x - x_1} + \frac{1}{x - x_2} + \dots + \frac{1}{x - x_n} = \frac{P_n'}{P_n} \equiv G \quad (3.6)$$

Para la detección del cambio de sentido en la curva, se aplica la segunda derivada de la ecuación: 3.5, obteniendo el valor H , que servirá como referente para descartar valores que estén lejos del punto de corte, como se muestra en la ecuación; 3.7

$$-\frac{d^2 \ln |P_n(x)|}{dx^2} = +\frac{1}{(x-x_1)^2} + \frac{1}{(x-x_2)^2} + \dots + \frac{1}{(x-x_n)^2} = \frac{P_n^2}{P_n} \equiv H \quad (3.7)$$

Una aproximación que se usa forzando a $Xn \Rightarrow \infty$, es la que se muestra en las ecuaciones: 3.8 y 3.9

$$\frac{1}{a} + \frac{n-1}{b} = G \quad (3.8)$$

$$\frac{1}{a^2} + \frac{n-1}{b^2} = H \quad (3.9)$$

Al terminar de calcular los valores anteriores n pertenece a todos los números enteros y va desde 0 hasta el máximo grado del polinomio. Al sustituir los valores por los valores obtenidos por las ecuaciones: 3.8 y 3.9, se puede tener una aproximación de los valores de las raíces del polinomio digitado; en el caso de que $nH - G^2$ sea negativo se aplica un nuevo criterio de iteración basado en el método de Müller.

$$a = \frac{n}{G \pm \sqrt{(n-1)(nH - G^2)}} \quad (3.10)$$

Las ecuaciones que se implementan permiten al método de Laguerre encontrar una raíz de un polinomio dado de grado N , donde sus coeficientes pueden ser

complejos, elemento que es una ventaja para la plataforma porque ayuda a la precisión de los cálculos de modelado de los sistemas. Dentro de la plataforma el primer coeficiente es para la potencia [0] y es una propiedad constante dentro del programa, mientras que en [m] está el coeficiente de la potencia más alta que puede tener el sistema. La programación de la librería raíces, proporciona una versión simplificada del criterio de detención de sub-polinomios debido a que cada 5 iteraciones rectifica los puntos de inflexión que hace la curva con una precisión de $1 * 10^{-10}$, que equilibra cuidadosamente el deseo de lograr la precisión completa del sistema, manteniendo el polinomio original digitado por el usuario intacto. Por un lado, se cuenta con el peligro de iteración, para funciones que cuenten con números muy pequeños y demasiado grandes en cada una de los términos del polinomio, pero para corregir este error, se compara y calcula la diferencia con el método de Müller.

Librería Límite Polinomio

Esta librería es una subclase de la clase raíces, debido a que de ella depende determinar las asíntotas a las que tiende un polinomio, devolviendo el cálculo de la cantidad de iteraciones para encontrar los polos o ceros de una función de transferencia. La tabla 3.7, muestra las características y funciones que esta posee.

Tabla 3.7: Esta tabla permite crear límites de polinomios tanto complejos como reales desde que no tenga denominadores, es importante para el cálculo de bode y calcular fracciones parciales

Nombre de la clase	LimiteP	Autoría	si	Tipo Clase	Statve	Lenguaje	Php
Descripción							
Permite encontrar el límite de una función con números reales o complejos desde que sea un polinomio de grado N							
Metodo o Atributo	Tipo de Método o Atributo	dato de entrada	dato de salida	Descripción			
LimitePlotComplex	Public Static	array polinomio, tienda a objeto complejo	número complejo objeto	Halla el límite de una función en términos complejos			
Limite PlotCC omplex	Public Static	objeto comlejo tienda a objeto complejo	objeto numero complejo	Halla el límite de una función en términos complejos con tendencia a ser número complejo			
LimitePol	Public Static	array polinomio Int tendencia	float	Encuentra el límite de un polinomio con tendencia a un número real			

Librería tf2fd

Esta librería se caracteriza por ser la clase padre de las demás librerías. Se encarga de ayudar al compilador principiar en la interpretación de comandos, diseñada para convertir los datos ingresados por el usuario en un sistema. La tabla: 3.8, muestra las diferentes características de esta clase.

Librería Tustin

La representación matemática de esta librería, permite a la plataforma modelar sistemas ante diversas señales, adicionalmente mostrar mejores gráficas en las representaciones en tiempo discreto como: Forward o Backward, sin contar que tiene en cuenta mejores detalles a la hora de trabajar con un modelo de tiempo continuo.

La representación de s en una función de transferencia de tiempo continuo, es

Tabla 3.8: Esta es la librería principal, la TF o transfer function se encarga de compilar y trabajar el resto de elementos matemáticos que ingresa el usuario.

Nombre de la clase	TF	Autoría	si	Tipo Clase	Public	Lenguaje	PHP5
Descripción							
Convierte dos array de tipo polinomio en una función de transferencia en términos de Laplace							
Método o Atributo	Tipo de Método o Atributo	dato de entrada	dato de salida	Descripción			
num	Atributo Publico	NA	NA	Guarda el polinomio del numerador			
den	Atributo Publico	NA	NA	Guarda el polinomio del denominador			
ident	Atributo Private	NA	NA	Guarda una variable visible para el usuario tipo string			
_constuct	Método público oculto	Obj->polinomio obj->polinomio	null	Crea el objeto tf quiere decir crea una función de transferencia			
getcerostf	Método publico	null	array: polos->array ceros->array sring	Obtiene los ceros de la función de transferencia y entrega los ceros y los polos en forma de arreglo junto con la descripción visible al cliente			
respusys	Método privado	null	Float: muestras Float: tiempo Repuesto Aproximado	Obtiene el polo más cercano al origen y calcula el tiempo aproximado de respuesta del sistema a 200 metros			
fracparcial	Metodo privado	null	Array de fracciones parciales	Calcula las fracciones parciales de la función de transferencia			
getfracpar	Método público	null	array: string: fractor string: Ilaplace	Entrega una cadena de datos con fracciones parciales inversas de Laplace ante el impulso para que los vea el cliente			
getbode	Método público	null	array: String: magnitud String: fase	Entrega los puntos de la gráfica de bode de la función de transferencia para que puedan ser mostrados al cliente			
impulsoftf	Método público	null	array: string: puntos string: tiempos	Entrega al cliente el gráfico en termino de puntos a razón del tiempo de la función de transferencia de una señal impulso			
gettf	Método público	null	objeto tf	Construye un nuevo objeto de la función transferencia			
tostring	Método publico oculto	null	null	Método por defecto que usa php, para imprimir una cadena dentro de una clase o constructor interno			

reemplazada como se muestra en la ecuación: 3.11, permitiendo generar un patrón para sistemas de n orden usando el triángulo de Pascal.

$$s = \frac{2(z - 1)}{T(z + 1)} \quad (3.11)$$

Para ecuaciones con polos conjugados, se usa la conversión que se muestra en

la ecuación 3.12. Que es un triángulo de Pascal, teniendo en cuenta los términos negativos.

$$\begin{aligned}(z - p)^2 &= z^2 - 2pz + p^2 \\(z - p)^3 &= z^3 - 3pz^2 + 3p^2z - p^3 \\(z - p)^4 &= z^4 - 4pz^3 + 6p^2z^2 - 4p^3z + p^4\end{aligned}\tag{3.12}$$

El código que se muestra en el anexo: C-Codigo1, muestra la forma en la que se estableció el patrón y se programó la parte principal de las librerías usando la clase principal polinomio.

3.3.5. Librerías de Propósito General

El uso de estas librerías ayuda a la interfaz ser más amigable con el usuario final, permitiendo la interacción con las bases de datos que se encuentran dentro de la sección modelos (como se mostró en la figura: 2.2). Ese esquema de organización de la plataforma basado en MVC, permite enfocar la plataforma en un elemento orientado por secciones, que facilita a la metodología la libertad de diseñar secciones según su nivel de prioridad, como las clases anterior mente explicadas.

Librería PID

Para el desarrollo de esta librería, se trabajará el modelo de control PID por medio del ajuste de polos, esto facilita al programa la operación de polinomios y así calcular las constantes: proporcional, integral y derivativa de la función $G(s)$. Teniendo en cuenta lo anterior, la representación en bloques de lazo cerrado como

muestra la figura: 3.2, se asume como la composición de un nuevo sistema o función de transferencia.

Figura 3.2: Representación en bloques en lazo cerrado de un sistema $H(s)$ con un controlador $G(s)$



Teniendo en cuenta la función de transferencia de la ecuación 3.13 en lazo cerrado, se puede pensar en la construcción de una librería que permita operar dos sistemas en lazo cerrado, esto quiere decir, un controlador PID para el caso de esta plataforma y respectiva planta.

$$H_n(s) = \frac{Y(s)}{U(s)} = \frac{1}{\frac{1}{H(s)*G(s)*\dots*n(s)} + 1} \quad (3.13)$$

A continuación en la ecuación: 3.14, se puede ver la representación de un PID usando un filtro pasa bajos en el término derivativo ($0 \leq \alpha \leq 1$), debido a que en un sistema real el denominador debe ser de un orden igual o superior al denominador, para que sea viable y calculable por la plataforma web⁵, con el fin de garantizar el desarrollo y utilidad de esta librería se usará $a = 0$.

$$G(S) = k_c \left(1 + \frac{1}{\tau_i s} + \frac{\tau_d s}{\alpha \tau_i s + 1} \right) \quad (3.14)$$

⁵Esta técnica se usa en otros casos, para eliminar un poco el ruido del sistema

Sintetizando $a = 0$ en la ecuación 3.14, se obtiene

$$G(S) = \frac{k_c \tau_d s^2 + k_c S + \frac{k_c}{\tau_i}}{s} \quad (3.15)$$

Remplazando los términos: $d_1 = k_c \tau_d$, $d_2 = k_c$ y $d_3 = \frac{k_c}{\tau_i}$ de la ecuación: 3.15, se obtiene la función de transferencia 3.16, donde: D es el polinomio numerador y E es el polinomio denominador de el PID

$$G(S) = \frac{d_1 s^2 + d_2 S + d_3}{s} = \frac{D}{E} \quad (3.16)$$

se puede decir que se puede convertir una planta H(S) como dos variables tipo polinomio A y B, usando las librerías de este capítulo, permitiendo crear una representación matemática como se muestra en la ecuación: 3.17.

$$H(s) = \frac{P_{num}}{P_{den}} = \frac{m}{s^2 + bs + c} = \frac{A}{B} \quad (3.17)$$

Cerrando el lazo con las ecuaciones: 3.16 y 3.17, se obtiene la nueva función de transferencia 3.18, esta permitirá de manera general entregar la función de la planta controlada a la plataforma.

$$P(s) = \frac{DA}{EB + DA} \quad (3.18)$$

La representación del código que se encuentra en el anexo: C-cod:1, es el algoritmo de la ecuación: 3.18, este código comparte características que se encuentran dentro de la librería polinomio, que se encarga de hacer las operaciones básicas

entre cada una de las funciones de transferencia.

Librería NumPHP

Esta es una librería de uso libre basada en la licencia MIT ⁶, se usó para hacer representaciones matemáticas optimizando recursos computacionales, en este caso la declaración de números con decimales tipo Long y creación de objetos de forma matricial matemática, la desventaja de esta librería es la ausencia de funciones matriciales que no cuenta con soporte para números complejos.

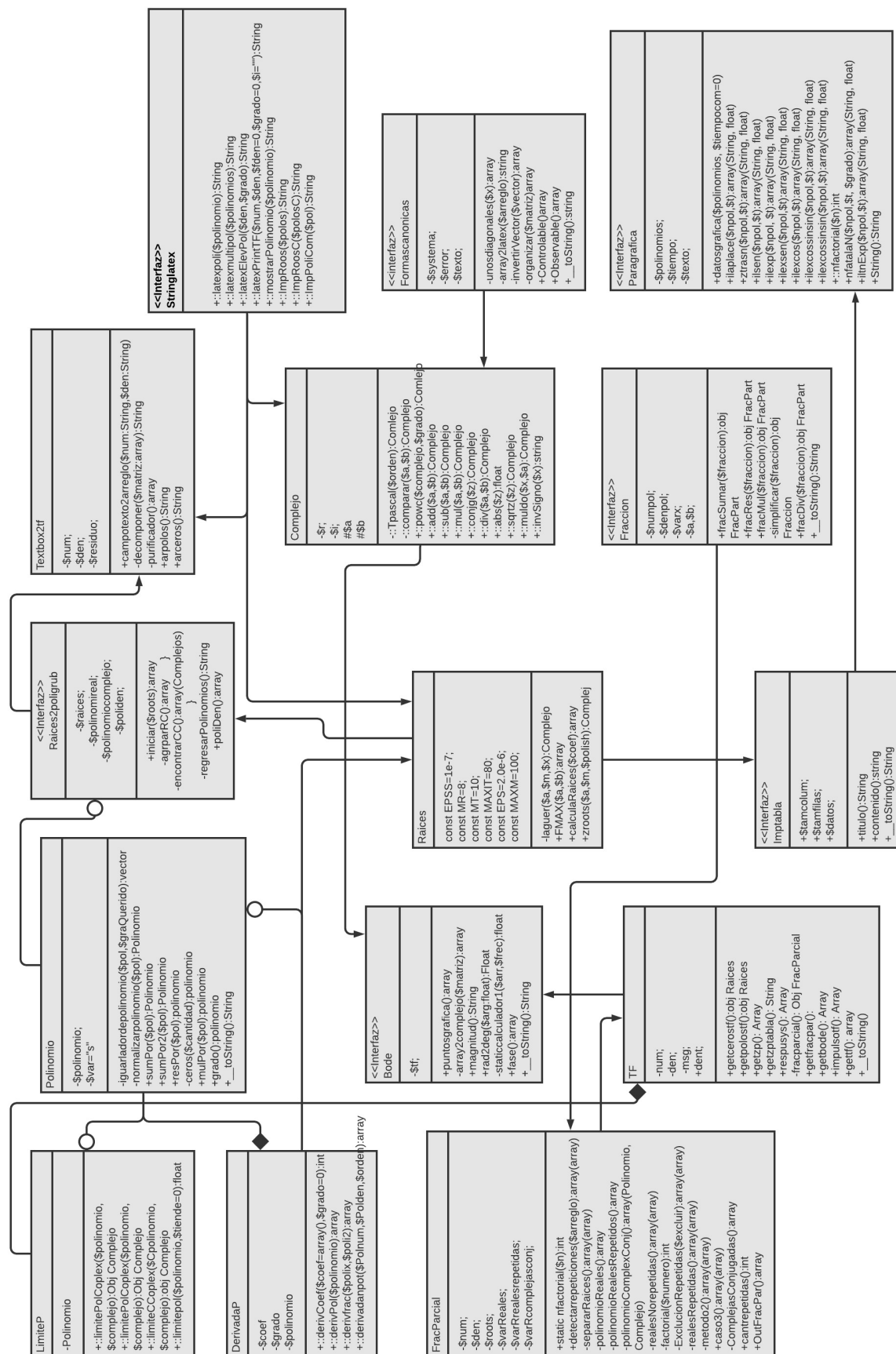
3.3.6. Diagrama UML Clases

En la figura: 3.3 se puede observar el diagrama de clases, donde indica la cantidad de librerías y la relación que tiene cada una para mostrar los resultados al usuario final.

Las clases de interfaz son las que se encargan de compilar los resultados para cuando el usuario final hace solicitud de los datos. Al enviar al servidor elementos tipo POST se muestran resultados ligados a las peticiones de cada sección de las herramientas según las características solicitadas por el front-end.

⁶ver anexos licencias

Figura 3.3: Diagrama UML de clases de las librerías desarrolladas



Capítulo 4

Modelos Matemáticos de la Plataforma

Unas de las metas importantes en la plataforma, es la implementación de sistemas dinámicos en términos de Laplace, en este caso se manejan tres tipos de sistemas con sus respectivos controladores. Para ello este capítulo está dedicado a la explicación general de estos modelos y como se implementarán dentro de la aplicación. Adicionalmente se tendrá en cuenta, que el usuario pueda aplicar otros tipos de sistemas y se visualicen resultados que sean útiles para que ayuden a entender su dinámica.

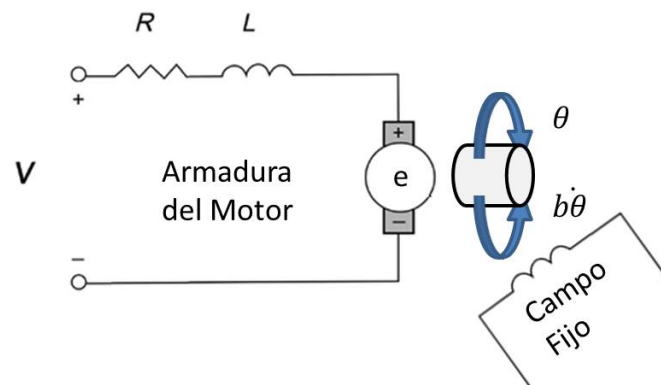
4.1. Motor de Corriente Continua

El motor eléctrico de corriente directa es uno de los sistemas que más se usa dentro de la elaboración de prácticas de control; entender su comportamiento dinámico es, quizá, uno de los elementos más importantes a la hora de realizar proyectos. Como es de saber, no todos los motores se comportan igual, depende de sus componentes y de la energía requerida para que este funcione de la manera correcta y

sea aplicable a un proyecto.

Aunque se han evaluado varios autores para la realización de este modelo, por flexibilidad se han decidido implementar los modelos planteados por el profesor **Bill Messner** del Departamento de Ingeniería Mecánica de la Universidad de Tufts. Esto se debe a que la mayor parte de implementaciones realizadas con estos modelos han sido experimentaciones físicas comparadas con elementos en Matlab 9.2. En la Figura 4.1, se pueden apreciar la estructura del motor, esto permitirá la contracción de las ecuaciones dinámicas.

Figura 4.1: Esquema del modelo de un motor DC



Los parámetros que se muestran en la tabla 4.1, son las variables internas que afectan directamente el comportamiento del motor, de ello dependerá la función de transferencia y los estados de operación del actuador.

Tabla 4.1: Tabla de parámetros físicos del motor eléctrico.

Letra Asignada	Nombre Representativo	Unidad de Medida
L	Inductancia Eléctrica	H
R	Resistencia Eléctrica	Ω
k_e	Constante Par Motor	$\frac{N}{Amp}$
k_t	Constante Fuerza Electromotriz	$\frac{V * seg}{rad}$
b	Constante Fricción Viscosa del Motor	$N * m * s$
J	Momento Inercia del Rotor	$Kg * m^2$

4.1.1. Modelo Matemático

Velocidad

Considerando la tabla 4.1, se procede a relacionar las variable del sistema, comenzando por el par motor $[T]$ que depende de la corriente que circula por la armadura $[i_a]$ y de la Fuerza Electromotriz $[K_t]$ generada por los imanes, como se ilustra en la ecuación 4.1.

$$T = k_t i_a \quad (4.1)$$

Los elementos de la armadura $[e]$ son proporcionales a la velocidad angular del eje $[\dot{\theta}]$ por un factor constante de Par Motor $[k_e]$, como se muestra en la ecuación: 4.2.

$$e = k_e \dot{\theta} \quad (4.2)$$

Al usar la segunda ley de newton que indica "si se aplica una fuerza a un cuerpo, éste se acelera"[Tippens, 2010], $f = ma$, entonces se pueden derivar las ecuaciones 4.3. Esto adicionalmente permite usar las leyes de voltaje de Kirchhoff

para construir la relación entre la tensión [V] y el ángulo [T].

$$\begin{aligned} j\ddot{\theta} + b\dot{\theta} &= k_i i \\ L \frac{di}{dt} + Ri &= v - k_e \dot{\theta} \end{aligned} \quad (4.3)$$

Para la facilidad del proyecto y la construcción del modelado, se puede usar la unidades SI, indicando que $K_t = K_e$ creando una nueva variable denominada M_k . Reemplazando y haciendo la transformación de Laplace queda la ecuación: 4.4.

$$\begin{aligned} j\theta(s)s^2 + b\theta(s)s &= M_k i(s) \\ Lsi(s) + Ri(s) &= v(s) - M_k s\theta(s) \end{aligned} \quad (4.4)$$

Factorizando obtenemos la ecuación: 4.5

$$\begin{aligned} \theta(s)(js^2 + bs) &= M_k i(s) \\ i(s)(Ls + R) &= v(s) - M_k s\theta(s) \end{aligned} \quad (4.5)$$

Al despejar $i(s)$ de las dos ecuaciones y reemplazarlas, se puede obtener la función de transferencia que relaciona la velocidad angular [θ] con la fuente de alimentación continua [V], tal como se muestra en la ecuación: 4.6.

$$\frac{\dot{\theta}(s)}{v(s)} = \frac{M_k}{JLs^2 + s(JR + bL) + Rb + M_k^2} \quad (4.6)$$

Posición

Como es de saber la velocidad es la derivada de la posición, $v(t) = \frac{dx}{dt}$, al tener la ecuación 4.6, que es la velocidad angular del eje del motor, se puede calcular su

posición integrando la función de transferencia, quedando la ecuación 4.7.

$$\frac{\theta(s)}{v(s)} = \frac{M_k}{JLs^3 + s^2(JR + bL) + s(Rb + M_k^2)} \quad (4.7)$$

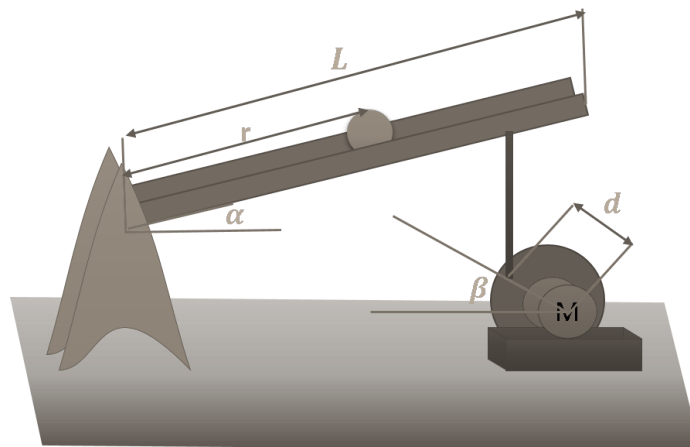
4.2. Ball of Beam (Viga Bola)

4.2.1. Modelo Matemático

El viga bola es un sistema medianamente complejo por su grado de inestabilidad, contemplando el sistema de posición del motor eléctrico es un sistema de orden 5 aproximadamente, pero puede aproximarse en la vida real a un sistema de orden dos usando la región lineal.

El modelo que se contempla dentro de la plataforma es el que se muestra en la figura: 4.2, debido a su flexibilidad de implementación en código. La identificación matemática es realizada por **Creative Commons Attribution**.

Figura 4.2: Esquema del modelo de un motor visual del viga bola (ball of beam)



Para evaluar los parámetros del sistema, se contemplan los datos de la tabla:

4.3, que son los datos que el usuario final ingresará para evaluar las condiciones dinámicas.

Tabla 4.2: Tabla de parámetros físicos de un viga bola.

Letra Asignada	Nombre Representativo	Unidad de Medida
g	aceleración gravitacional	$9,8 \frac{m}{s^2}$
β	ángulo del engranaje del servo	radianes
r	coordenada de posición de la pelota	m
α	coordenada del ángulo del haz	radianes
d	desplazamiento del brazo de palanca	m
L_D	longitud del haz	$m \rightarrow 0,1 \leq m < 1$
m	masa de la pelota	k_g
J	momento de inercia de la bola	$k_g \cdot m^2$
R	radio de la pelota	m

Al analizar el comportamiento del sistema, cuando se cambia el ángulo $[\alpha]$, se puede ver una variación que afecta directamente la coordenada de la esfera $[r]$ dentro de la viga. Al calcular la segunda derivada de $[r]$, se puede apreciar la aceleración de la esfera, esto permite aplicar las ecuaciones de **LaGrange**, como se muestra en la ecuación: 4.8.

$$0 = \left(\frac{J}{R^2} + m \right) \ddot{r} + mg \sin \alpha - mr\dot{\alpha}^2 \quad (4.8)$$

Al encontrar la aproximación lineal del sistema haciendo $\alpha = 0$, aparece la ecuación 4.9. Esta permite entender las regiones de trabajo dentro del viga bola, es decir, que la esfera se mantenga en una región de operación determinada.

$$\left(\frac{J}{R^2} + m \right) \ddot{r} = -mg\alpha \quad (4.9)$$

Para complementar el sistema se debe hacer lineal el valor de α , esto se logra mediante la aproximación de la ecuación: 4.10.

$$\alpha = \frac{d}{L}\beta \quad (4.10)$$

Remplazando en la ecuación: 4.9 el valor lineal de α , obtenido en la ecuación: 4.10, se puede obtener la relación de la coordenada de la esfera r con el ángulo del engranaje del servo β , como se ilustra en la ecuación: 4.11.

$$\left(\frac{J}{R^2} + m\right)\ddot{r} = -mg\frac{d}{L}\beta \quad (4.11)$$

Al aplicar la transformación de Laplace de la ecuación diferencial 4.11, se puede construir la función de transferencia, donde la relación entrada-salida será la coordenada de la esfera r entre el ángulo generado por el servomotor β quedando el sistema como muestra la ecuación:

$$\frac{R(s)}{\beta(s)} = -\frac{mgd}{L\left(\frac{J}{R^2} + m\right)}\frac{1}{s^2} \quad (4.12)$$

En la ecuación: 4.12 se puede ver un sistema en el cual el ángulo ya está determinado por los engranajes de un servomotor, no se contempla la existencia de un motor, sin embargo, al acoplar la función de transferencia construida en la ecuación 4.6, indicando que $\theta = \beta$ el sistema completo queda como muestra la ecuación: 4.13, empatando la dinámica de viga y bola y el funcionamiento dinámico de un motor de corriente continua.

$$\frac{R(s)}{V(s)} = - \frac{mgd(M_k)}{Ls^2 \left(\frac{J}{R^2} + m \right) \left(JJs^3 + s^2(JR + bL) + s(Rb + M_k^2) \right)} \quad (4.13)$$

Es importante mencionar que, en la ecuación 4.13 en el motor no se está contemplando ningún tipo de controlador; se recomienda aplicar primero el controlador del motor para luego ejecutar el controlador del viga bola, para simplificar procesos matemáticos que es lo que internamente la plataforma hace para sus cálculos.

4.3. Aeropéndulo

Este un sistema, debido a su nivel de complejidad, permite trabajar las diferentes etapas de desarrollo de controladores, en los dos curso de control, tanto analógico como digital.

Aunque presenta una lógica similar a un péndulo simple, analizar el sistema por este modelo permite tener una mejor relación del sistema matemáticamente.

4.3.1. Modelo Matemático

El modelo matemático usado para esta representación fue desarrollado por **Giampero Campa**, miembro de la IEEE. Consiste en dar una demostración lineal del modelo, que se ajusta a las características de la plataforma pues es un sistema de segundo orden, y adicional a ello se puede implementar dentro de las librerías construidas dentro del programa. A continuación se definen los parámetros para la

construcción de la función de transferencia.

Parámetros de configuración:

Tabla 4.3: Tabla de parámetros físicos Aeropendolo.

Letra Asignada	Nombre Representativo	Unidad de Medida
L	longitud de la vara	$[m]$
M_g	Peso del motor	$[K_g]$
K	coeficiente de empuje	n/a
C	Coeficiente de Fricción	n/a
T	Fuerza de la hélice	n/a
μ	Energía Requerida	μ

La dinámica del comportamiento del péndulo está dada por la ecuación 4.14, esta involucra los parámetros de fricción, peso de la vara y las fuerzas que trabajan en este sistema.

$$mL^2\ddot{\theta} = -mg\text{sen}\theta - c\dot{\theta} + TL \quad (4.14)$$

Teniendo en cuenta que T es la fuerza que ejerce la hélice, será proporcional a la cantidad de energía de energía de propulsión μ por el coeficiente de empuje. Reemplazando esto en la ecuación 4.14 se obtiene la solución de la ecuación 4.15.

$$mL^2\ddot{\theta} = -mg\text{sen}\theta - c\dot{\theta} + TK\mu \quad (4.15)$$

Definiendo el sistema bajo una región de trabajo lineal diciendo que el valor de los parámetros en estado estacionario está dado por la ecuación 4.16, se puede llegar a la función de transferencia lineal.

$$\sin\theta_{ss} = \frac{K}{mg}\mu_{ss} \quad (4.16)$$

La función de transferencia del sistema está dada por la ecuación 4.17. Esta se usa dentro de la plataforma para hacer las simulaciones del sistema del Aeropéndulo $\Theta(s)$ es el ángulo de salida y $W(s)$ es la energía con la que se alimenta el sistema.

$$\frac{\Theta(s)}{W(s)} = \frac{KL}{mgL^2s^2 + cs} \quad (4.17)$$

Capítulo 5

Gestión de Usuarios

Una de las características que tiene la plataforma es poder administrar usuarios, de tal manera que este al estar registrado pueda acceder o realizar preguntas a una comunidad o miembros que el usuario desee.

Figura 5.1: Pantalla de registro de usuarios nuevos

Lina La Plataforma Modelos de Sistemas Herramientas Ingresar

Login Registrar o Ingresar a lina

[Pagina Principal](#) / Usuarios de la plataforma

Bienvenido!
Es hora de registrarte...

Email:

Nombres:

Apellidos:

Clave:

Repetir Clave:

Nivel Educativo:

Institucion:

Mas informacion...

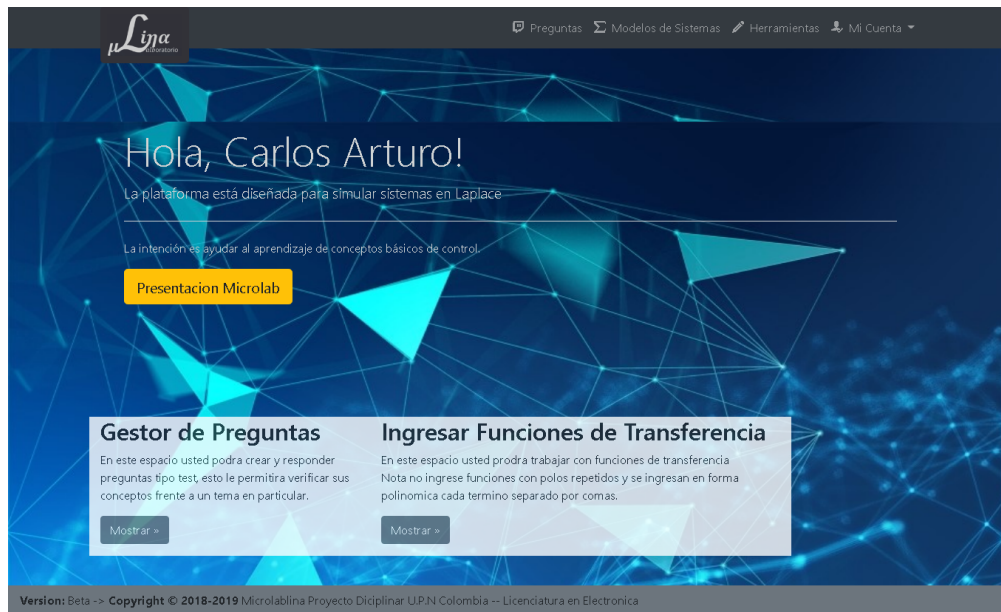
UNIVERSIDAD PEDAGOGICA NACIONAL
Facultad de Ciencia y Tecnologia
Licenciatura en Electronica

Plataforma Presentada
Como Trabajo de Grado.

Carlos A. Moreno S.
Bogota D.C. - Enero - 2019

Cuando el usuario llena el formulario de registro que se muestra en la figura: 5.1, se hace miembro de microlablina. Al introducir el usuario y contraseña, y ser confirmadas, la persona automáticamente verá la interfaz de bienvenida como se muestra en la figura: 5.2, mostrando dos opciones en su ventana principal, la sección de gestión de preguntas y la sección para trabajar con funciones de transferencia.

Figura 5.2: Pantalla de Bienvenida a usuarios registrados

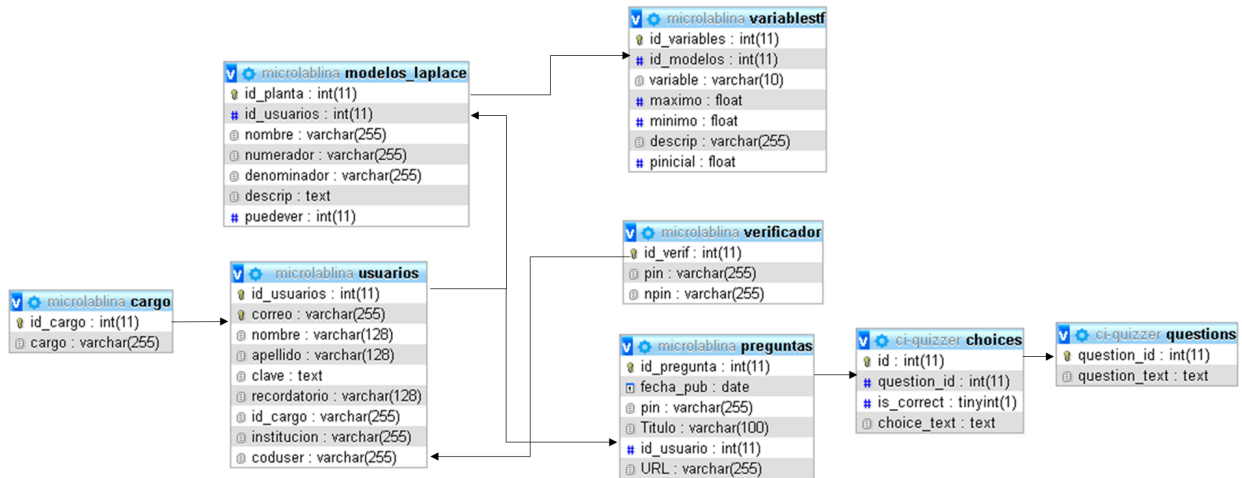


5.1. La base de datos

El gestor de bases de datos que se está usando para este proyecto es MySQL, en su versión 8.0.17. Su licencia está basada en los estándares de GNU/GPL y su uso es gratuito en el servicio de alojamiento que se está trabajando, que está por defecto pre-instalado en el servidor y posee las características necesarias para la realización de este trabajo de grado.

Se ha propuesto la estructura básica de la base de datos que se muestra en la figura: 5.3. Esta arquitectura de diseño es de un contenedor de datos de manera relacional, trabaja en modo cascada, es decir que al alterar un dato afecta el resto de contenidos de las otras tuplas aliadas a la tabla maestra.

Figura 5.3: Arquitectura de la base de datos dentro de la plataforma Microlablina



En la tabla 5.1, se muestra las propiedades de cada uno de los métodos que tiene la base de datos microlablina.

Tabla 5.1: Tabla explicativa de los métodos existentes en la base de datos

Nombre	Características
<i>cargo</i>	Esta tabla se encarga administrar los roles de usuario de la plataforma. Sin embargo, solo se trabajan 3 de los cuales se hablarán más adelante.
<i>modelos_laplace</i>	Esta tabla es la más importante de la plataforma, ya que almacena las funciones de transferencia en términos de polinomio. Aunque la plataforma no cuenta con el espacio para registrarlas se deja la posibilidad para futuras actualizaciones.
<i>preguntas</i>	Permite a los usuarios crear un espacio para agregar las rutas donde los docentes y/o estudiantes pueden agregar o acceder a las preguntas de otra plataforma.

<i>usuarios</i>	Esta tabla permite a la plataforma guardar los miembros que se registren a la misma, con sus datos básicos para acceder a ella desde diferentes lugares.
<i>variablestf</i>	Esta tabla es el complemento de la tupla modelos_laplace, guarda los parámetros de una función de transferencia y del modelo que se desea modelar dentro de la plataforma
<i>verificador</i>	Evita que los registros que se agregaron desde la sección de habilitación de vínculos se filtren en otras cuentas, ya que es una tabla auxiliar de la tupla preguntas.
<i>choises</i>	Administra preguntas específicas dentro de la plataforma, esta tabla no se encuentra habilitada pero esta en modo auxiliar para el control de bitácoras de usuarios.
Questions	Guarda preguntas en la plataforma. Esta tabla no esta habilitada debido a la ineficiencia en el almacenamiento.

5.2. Roles de usuario

Un elemento importante en el desarrollo de aplicativos es el manejo de roles de usuario, ya que permite otorgar a la plataforma algunas ventajas frente a diferentes tipos de clientes que manejen la aplicación, en esta sección se mencionarán tres roles, de los cuales el Anfitrión e invitado se manejan de manera paralela y son menesteres al momento de asignar o ver preguntas dentro de la plataforma

5.2.1. Público

Este usuario puede acceder a los parámetros externos de la plataforma, como visualizar y modelar sistemas en términos de función de transferencia, adicionalmente puede trabajar sobre los modelos que plantea de manera nativa la aplicación.

5.2.2. Anfitrión

Este usuario tiene la característica de otorgar un pin para que se vinculen a su cuenta, permitiendo compartir las preguntas o vínculos a un grupo específico de usuarios que cuenten con el pin del anfitrión.

5.2.3. Invitado

El usuario invitado solo puede acceder a los vínculos, no podrá eliminar miembros que estén registrados dentro del mismo equipo de trabajo, tampoco podrá saber qué usuarios comparten el mismo vínculo de enlace.

5.3. Sección de Preguntas

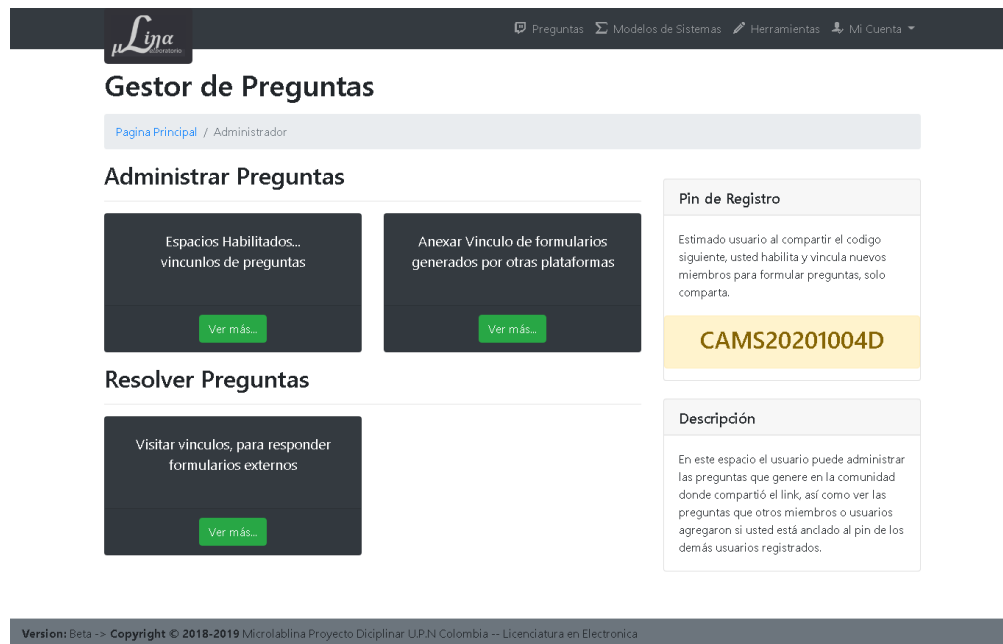
En la plataforma virtual microlablina se proyectó habilitar un escenario para que los miembros pudieran solucionar preguntas del docente, que en este caso es el anfitrión de la plataforma. Para ello se utiliza el aplicativo como un administrador de vínculos, donde el docente (anfitrión) comparte el formulario a su comunidad, y ellos pueden acceder a este mediante una ruta que solo los invitados tienen acceso. Se realizó de esta manera, debido a que hay plataformas más completas complejas y diseñadas de manera táctica para el desarrollo de test, y que son de uso común en la comunidad de educación. Se piensa de esta manera para que el maestro tenga la oportunidad de seleccionar herramientas como kahoot, google forms, entre otras que son más dinámicas, completas y no tienen una curva de aprendizaje amplia para

su uso o implementación, esto da prioridad al elemento fundamental del programa, que es servir como emulador virtual.

5.3.1. Diseño visual del gestor de preguntas

En la figura: 5.4 se puede ver la plantilla principal al acceder al vínculo de gestor de preguntas. En éste se encuentran tres elementos importantes: en los dos primeros cuadros negros se aprecia el registro de preguntas junto con su respectivo listado, tanto en el anfitrión como en docente; en el tercer cuadro se encuentra el modo invitado, que son las preguntas formuladas por otro miembro de la comunidad. En

Figura 5.4: Ilustración de la plataforma microlablina dentro de un usuario registrado en la sección de gestor de preguntas



el panel derecho inferior, se encuentra en color naranja el pin, donde el anfitrión puede compartirlo para el resto de invitados.

En la figura: 5.5, se puede ver la cantidad de preguntas que el anfitrión ha

realizado, este podrá eliminarlas o agregarlas a conveniencia, adicionalmente le permite verificar si el vínculo está funcionando.

Figura 5.5: Ilustración de la plataforma microlablina dentro de un usuario registrado en la sección listado de vínculos

Version: Beta -> Copyright © 2018-2019 Microlablina Proyecto Disciplinar UPN Colombia -- Licenciatura en Electronica

Figura 5.6: Registro de vínculos dentro de la aplicación web.

Version: Beta -> Copyright © 2018-2019 Microlablina Proyecto Disciplinar UPN Colombia -- Licenciatura en Electronica

En la figura: 5.6 se muestra la imagen del diseño del formulario para ingresar

nuevos vínculos, permitiendo colocar en sus respectivos campos el nombre de la actividad o ruta y el url de la misma.

A continuación se muestra la sección gráfica que ve el invitado. Cuando entra por primera vez el usuario, tiene que digitar el pin del anfitrión como muestra la imagen de la figura: 5.7, donde muestra el campo de texto y el cuadro informativo de la parte derecha. Por último, cuando el usuario a digitado de manera correcta

Figura 5.7: Ingreso del pin suministrado por el anfitrión.

Figura 5.8: Lista de preguntas que puede ver el invitado.

#	Titulo	Fecha	Config
0	sistema digital 2	2020-10-04	Ir
1	sistema digital 2 jmdnjdsdm	2020-10-04	Ir

el pin, el invitado podrá ver las preguntas o vínculos suministrador por el anfitrión como se muestra en la figura: .

Generalidades

En futuras actualizaciones de la aplicación se espera fortalecer en la sección de usuarios el guardado de simulaciones, así como mejoras en los formularios de preguntas, facilitando al usuario una interacción más amigable con la plataforma.

Aunque en la base de datos se guarda información sobre los modelos matemáticos, estos no cuentan con una herramienta propia que le permita al usuario final la modificación e ingreso de nuevos datos. Se deja la posibilidad de modificar la base de datos para poder agregar tuplas y mejorar la interacción con el cliente en futuros proyectos de desarrollo para esta plataforma.

Capítulo 6

Interfaz de Usuario y Aplicaciones

En el transcurso de este capítulo se visualizará la interfaz gráfica de la aplicación Microlablina. El diseño de este programa está pensado para trabajar en dispositivos de mesa y móviles, siempre y cuando manejen un navegador que cuente con los protocolos de la web 3.0. Cuenta con una interfaz y una base de datos que permiten al usuario administrar e integrar formularios de otras plataformas, con el fin de gestionar preguntas que permitan al individuo trabajar interactuar con las variables y modelos que esta presenta.

6.0.1. Navegadores Soportados por Microlablina.

En la tabla: 6.1 se puede observar el funcionamiento de la página ante los diferentes navegadores web, estas pruebas se realizaron en sistemas operativos con Windows: 7, 8 y 10, así como sistemas Linux tales como: Ubuntu 19.04 , Fedora 34, Debian 8, junto con los sistemas de los dispositivos móviles desde Android 5 hasta el 11.

Adicionalmente los valores: bajo, medio, bueno, excelente, representan el estado de funcionabilidad de cada una de las capas testeadas. El valor Bajo, significa que causa bloqueos del navegador o un aspecto visual ilegible para el usuario; Medio representa que presenta algunas inconsistencias y varían en la versión del navegador; Bueno representa que las características de las capas no presentan anomalías visuales o significativas, pero cambian de la versión planteada en la hoja de estilos del programador; Excelente, siendo la calificación más alta, representa que trabaja según lo deseado.

Tabla 6.1: Esta tabla muestra el funcionamiento de la página web en los navegadores más usados hasta febrero de 2019.

Navegador	Simulación en tiempo real	Ecuaciones matemáticas	Graficado de funciones	Estilos de interfase	Formularios dinámicos	JavaScript json
<i>Chrome PC</i>	<i>Excelente</i>	<i>Excelente</i>	<i>Bueno</i>	<i>Bueno</i>	<i>Excelente</i>	<i>Excelente</i>
<i>Chrome Móvil</i>	<i>Bueno</i>	<i>Excelente</i>	<i>Medio</i>	<i>Bueno</i>	<i>Excelente</i>	<i>Excelente</i>
<i>Opera</i>	<i>Excelente</i>	<i>Excelente</i>	<i>Bueno</i>	<i>Bueno</i>	<i>Excelente</i>	<i>Excelente</i>
<i>Opera Móvil</i>	<i>Bueno</i>	<i>Bueno</i>	<i>_Medio</i>	<i>Bueno</i>	<i>Excelente</i>	<i>Excelente</i>
<i>Microsoft Edge</i>	<i>Excelente</i>	<i>Excelente</i>	<i>Bueno</i>	<i>Bueno</i>	<i>Bueno</i>	<i>Bueno</i>
<i>Safary</i>	<i>Bajo</i>	<i>Bueno</i>	<i>Medio</i>	<i>Medio</i>	<i>Bueno</i>	<i>bueno</i>
<i>Firefox</i>	<i>Excelente</i>	<i>Excelente</i>	<i>Bueno</i>	<i>Bueno</i>	<i>Excelente</i>	<i>Excelente</i>
<i>konqueror</i>	<i>Bajo</i>	<i>Bueno</i>	<i>Medio</i>	<i>Medio</i>	<i>Bueno</i>	<i>Bueno</i>

6.1. Pantalla de Inicio

Es la presentación del programa, muestra las características centrales de la plataforma, así como para quiénes fue desarrollada.

Figura 6.1: Esta imagen representa el estilo de diseño en el cual carga la pantalla inicio para ordenador y móvil. Esta captura corresponde a la versión 1.

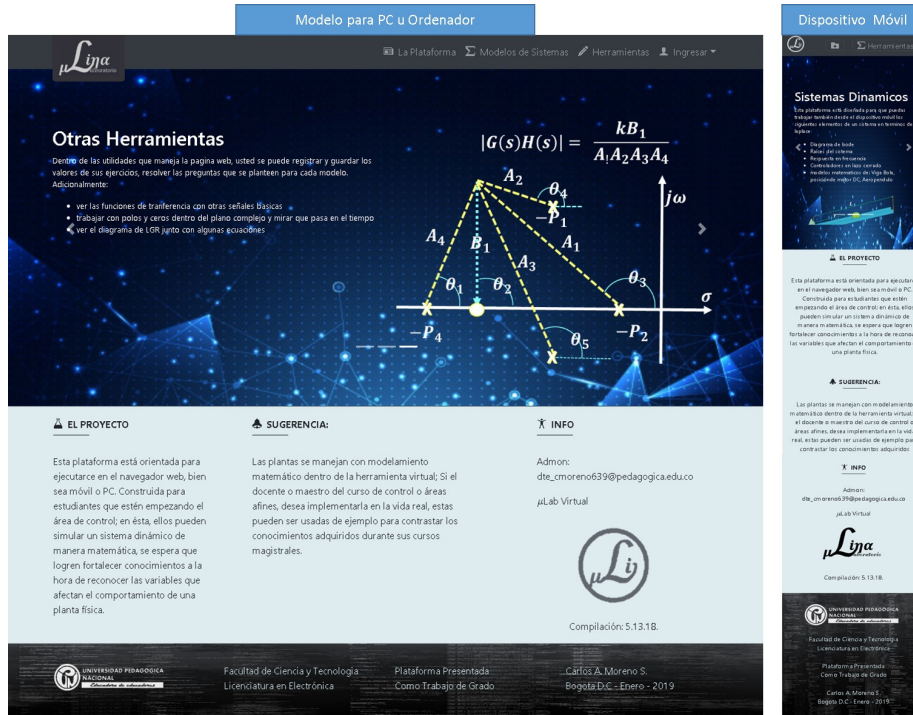
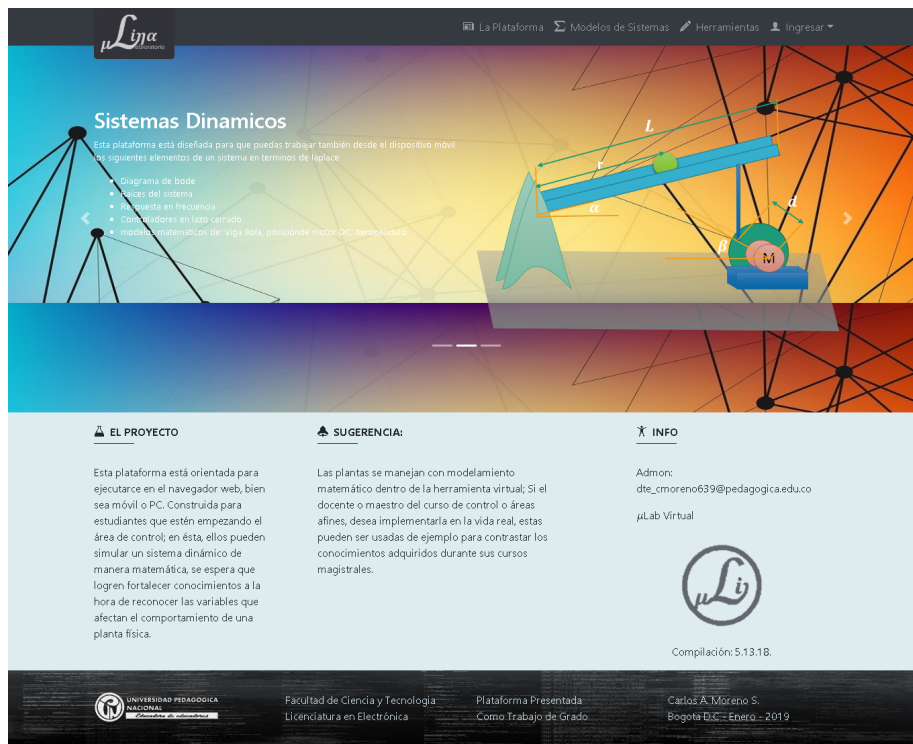


Figura 6.2: Imagen de presentación actual de la pantalla de bienvenida, versión Beta Final.



En la Figura: 6.1 se muestra cómo está distribuida la información en cada parte de la página, en la parte derecha se visualiza cómo se adapta dispositivos de pantalla de 7 pulgadas o pantallas de celulares.

En la figura 6.2 se puede apreciar la presentación actual de la portada, evidenciando las características fundamentales que han cambiado desde la versión anterior.

6.2. Herramientas Experimentales

Un elemento fundamental para corroborar el funcionamiento correcto de las librerías es el desarrollo de estas herramientas, ya que permiten tanto al programador como a los demás usuarios ingresar datos de otros modelos matemáticos en términos de función de transferencia y verificar sus resultados, se deja habilitado con el fin de que otras personas puedan trabajar con la totalidad de recursos que tiene cada una de las clases programadas dentro del micro simulador.

En la figura 6.3, se muestra la interfaz de usuario donde se encuentran cada una de las herramientas de propósito general que tiene la página web.

6.2.1. Funciones de Transferencia Simuladas con Diferentes Fuentes

Esta es una de las herramientas que permite al usuario trabajar con diferentes fuentes y verificar su comportamiento en el tiempo, usa la librería **Tustin** (ver capítulo: 3.4), para hacer una ecuación en términos de diferencias y así dar una

Figura 6.3: Pantalla de presentación de herramientas junto con su descripción según la utilidad de cada una.

Función de Transferencia
Sistemas Dinámicos (Laplace)

Herramientas:

- F(s)** Cálculo de la Función de Transferencia. Esta herramienta permite calcular la función de transferencia de un sistema a partir de sus parámetros físicos.
- H(s)** Cálculo de la Función de Transferencia. Esta herramienta permite calcular la función de transferencia de un sistema a partir de sus parámetros físicos.
- M(s)** Cálculo de la Función de Transferencia. Esta herramienta permite calcular la función de transferencia de un sistema a partir de sus parámetros físicos.
- L(s)** Cálculo de la Función de Transferencia. Esta herramienta permite calcular la función de transferencia de un sistema a partir de sus parámetros físicos.
- S(s)** Cálculo de la Función de Transferencia. Esta herramienta permite calcular la función de transferencia de un sistema a partir de sus parámetros físicos.

Introducción

En este espacio usted podrá trabajar con funciones de transferencia en términos de Laplace de la forma:

$$H(s) = \frac{P(s)}{Q(s)} = \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}$$

Obteniendo las diferentes características que describen el funcionamiento del sistema dinámico que usted digite, tales como: las raíces, el lugar geométrico, diagrama de Bode, adicionalmente ver el comportamiento del sistema ante las diferentes señales que le puedan afectar en su entrada.

Como parte del trabajo para sistemas 2250, está diseñado con fines educativos para la experimentación y comprensión de temas básicos de control sistemas dinámicos.

EL PROYECTO

Esta plataforma está orientada para que durante el desarrollo de sus trabajos de investigación, usted pueda utilizar esta herramienta para el desarrollo de sus trabajos de investigación.

SUGERENCIA:

Las herramientas manejan un modelamiento matemático dentro de la herramienta virtual. Si el docente o maestro del curso de control desea afirmar, desde la herramienta en la vida real, estas pueden ser usadas de manera constante los conocimientos adquiridos durante sus cursos magistrales.

INFO

Admin: drc@unioyca.edu.co
Juan Virella

Compañía: S.L.L.S.

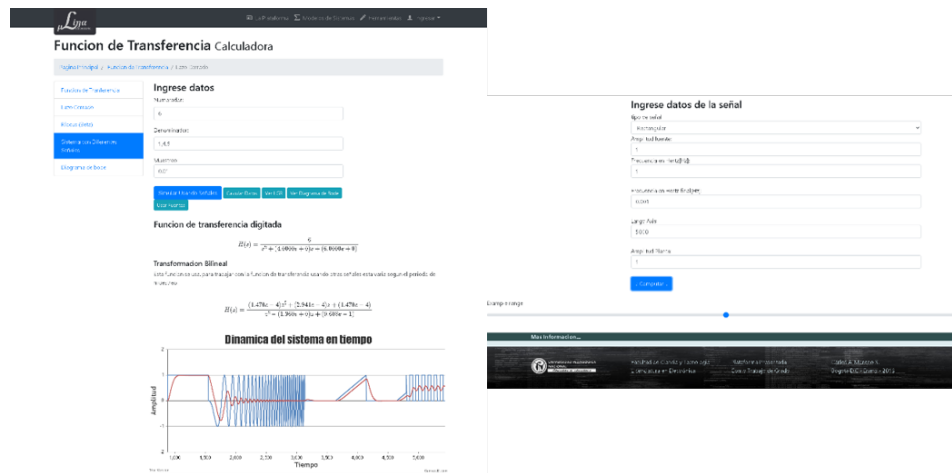
Facultad de Ciencias y Tecnología
Licenciatura en Electrónica
Plataforma Propuesta Como Trabajo de Grado
Carlos A. Moreno S.
Bogotá D.C. Enero - 2019

representación en tiempo real, el efecto de esta dependerá del dispositivo donde trabaje y está en modo piloto. En la figura: 6.4, se puede apreciar la presentación de la sección dentro de la página web, para que funcione correctamente, el usuario deberá digitar tiempo de muestreo así la función de transferencia separando el polinomio del denominador y el numerador en cada uno de los cuadros de texto que solicita el programa, adicionalmente cada termino lo deberá separar con una coma, no podrá usar letras solo números puntos y comas.

6.2.2. LGR Lugar Geométrico de las Raíces

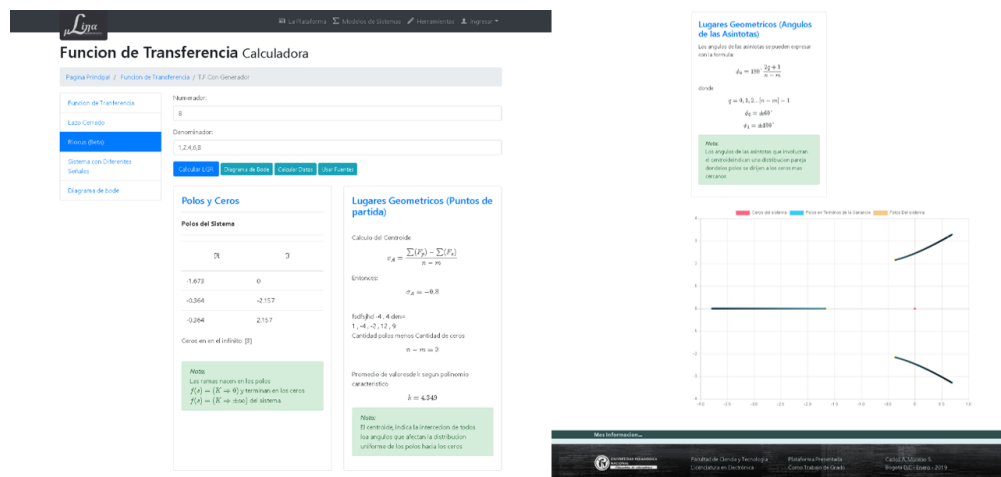
Es una herramienta experimental que muestra algunos elementos importantes para realizar el trazado del lugar geométrico de una función de transferencia, para poder usar este recurso se requiere tipear en las barras de texto la función de

Figura 6.4: Simulador de función de transferencia con otras fuentes



transferencia separando el polinomio del denominador y el numerador en cada uno de los cuadros de texto que solicita el programa, adicionalmente cada término lo deberá separar con una coma, no podrá usar letras solo números puntos y comas. En la figura: 6.5, se muestra el resultado después de digitar como se solicitó la información del sistema a modelar.

Figura 6.5: Muestra el lugar geométrico de las raíces, sección en modo desarrollo



6.2.3. Diagrama de Bode

Es la herramienta más importante de la aplicación, debido a que permite a las demás librerías saber a qué frecuencias opera el sistema y hacer los graficados más suavizados, a esta interfaz como a la librería principal se le han hecho tres actualizaciones, en las cuales se ha cambiado la forma en la que analiza la información (ver capítulo: 3). La función de transferencia se digita, separando el polinomio del denominador y el numerador en cada uno de los cuadros de texto que solicita el programa, adicionalmente cada término lo deberá separar con una coma, no podrá usar letras solo números puntos y comas. En la figura: 6.6, se muestra la forma en la que la aplicación muestra los resultados.

6.2.4. Experimentos Libres

Funciones de Transferencia Lazo Abierto

Es el corazón del sistema ya que permite simular varios modelos de transferencia, separando el polinomio del denominador y el numerador en cada uno de los cuadros de texto que solicita el programa, adicionalmente cada término lo deberá separar con una coma, no podrá usar letras solo números puntos y comas. En la figura: 6.7, se puede ver el diseño de la interfaz para ingresar los datos de la función de transferencia, así como la forma en la que muestra los resultados la cliente. En la figura: 6.8 se muestra la representación gráfica de la función de transferencia digitada ante las entradas impulso y escalón.

En la figura: 6.9 se muestra la respuesta del sistema ante el impulso y adicional

Figura 6.6: muestra el diagrama de bode de la función digitada

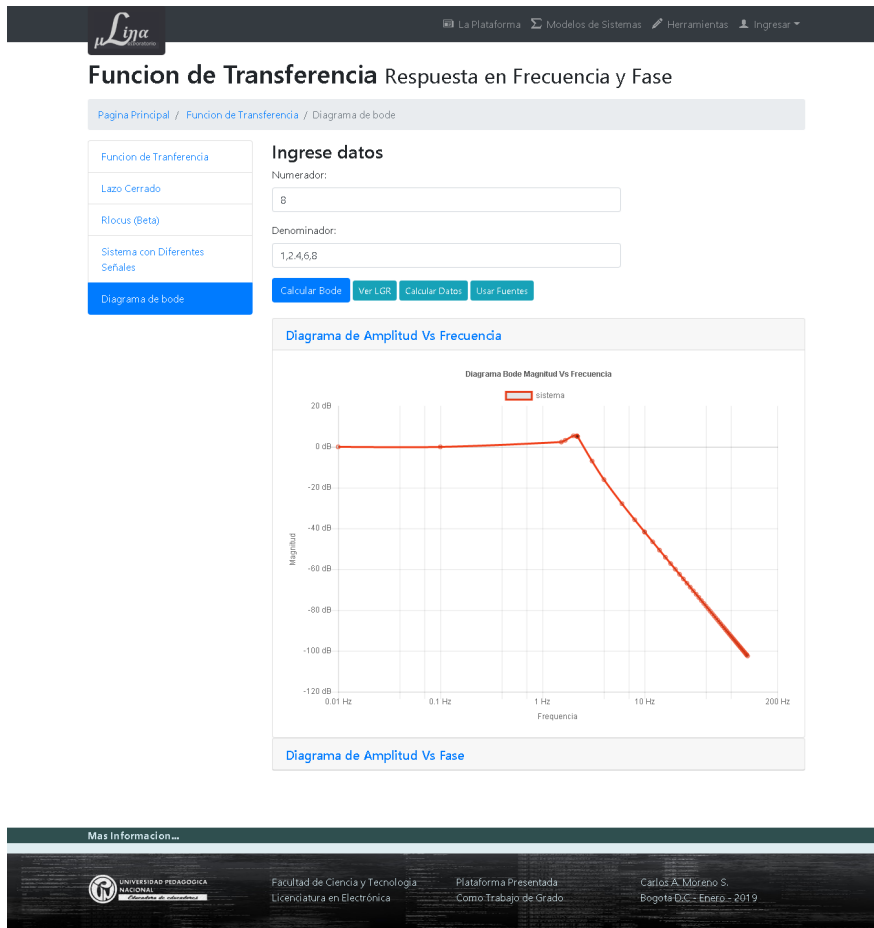


Figura 6.7: Interfaz antes y después de digitar la función de transferencia

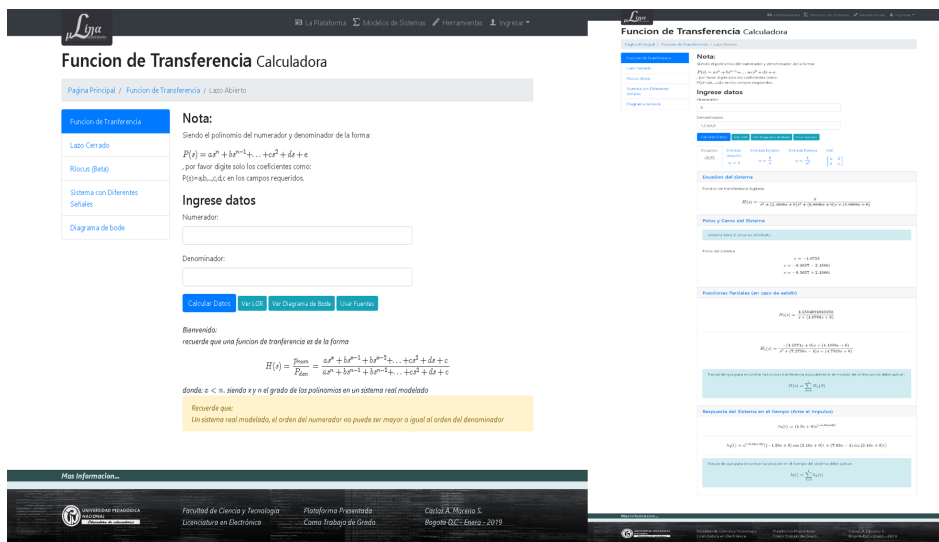
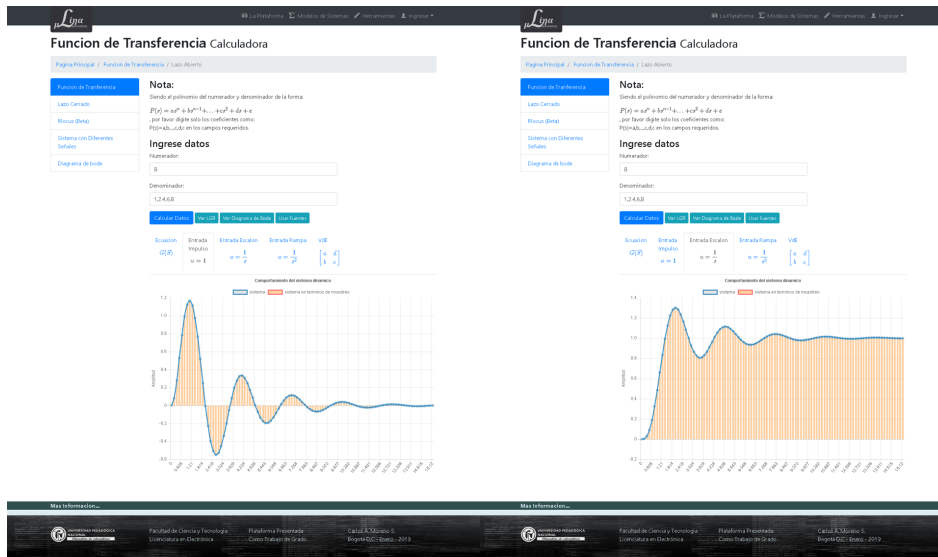
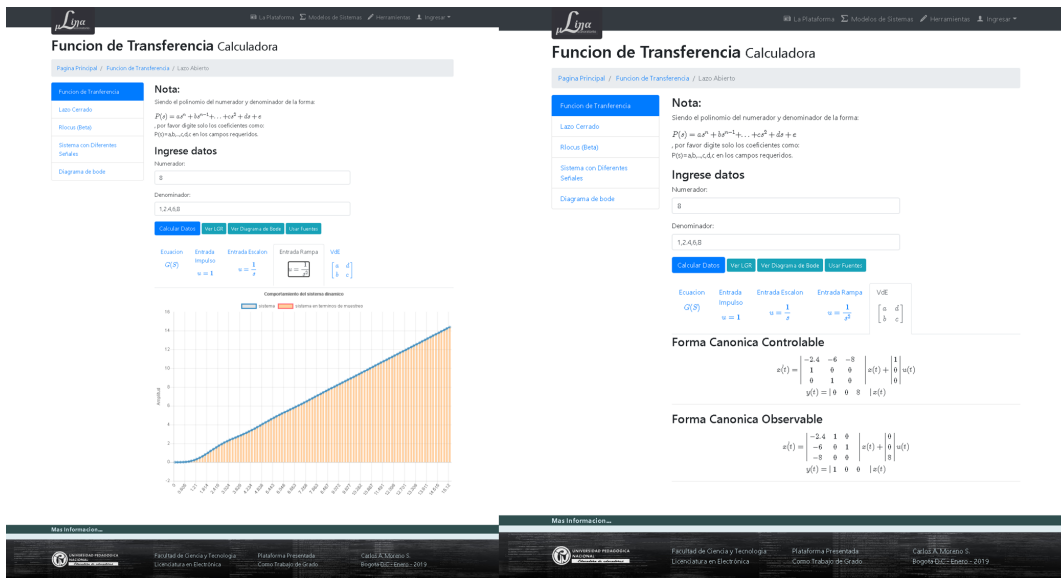


Figura 6.8: Resultados de la función de transferencia en impulso y escalón.



mete muestra la representación en variables de estados en sus formas Canónicas: Observable y Controlable, no se muestran cálculos de las mismas.

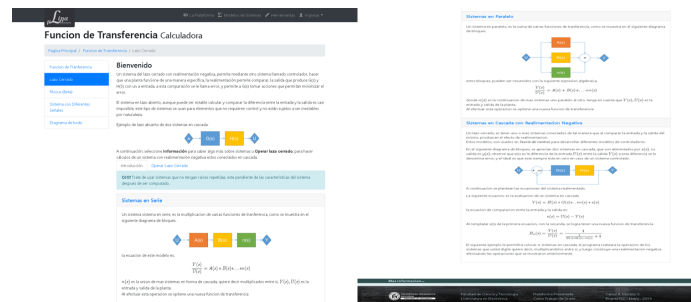
Figura 6.9: Resultados de la función de transferencia ante la rampa y al calcular las variables de estado0.



Funciones de Transferencia Lazo Cerrado Realimentación negativa

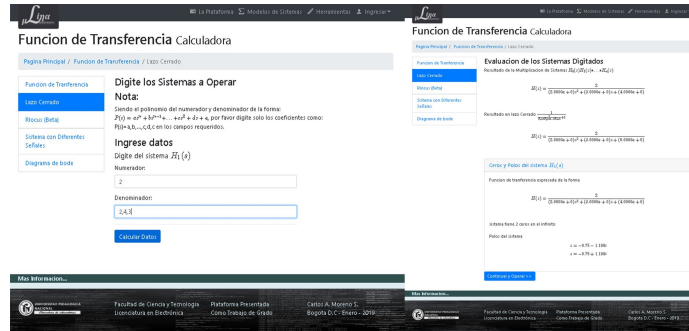
Esta es una calculadora que permite unir uno o más sistemas en forma de cascada, y luego aplica un lazo cerrado, como muestra la figura: 3.13 del capítulo 3, esta librería es importante debido a que de ella depende el trabajo con controladores para los sistemas que están por defecto dentro de la aplicación. En la figura 6.10 se muestra a grandes rasgos cómo es la interfaz gráfica, la plataforma muestra una explicación sencilla de cómo se puede realizar cálculos con cada sistema según como esté organizado cada bloque, en la última parte se centra en resaltar los cálculos que hace microlablinea, para tomar todo como un sistema final ya computado. En la figura: 6.11 se muestra en la parte izquierda la solicitud de

Figura 6.10: Sección de lazo cerrado, plantilla explicativa



datos de cada uno de los sistemas que desea introducir y operar en modo de realimentación, en el lado derecho le muestra los datos del sistema así como las operaciones que hizo para convertir todo en una nueva función de transferencia.

Figura 6.11: Ingreso de funciones de transferencia a la plataforma, para realizar la unión de lazo cerrado



6.3. Interfaz de modelos matemáticos de manera nativa.

Dentro de los objetivos que se plantearon en el proyecto de trabajo de grado se establecieron tres plantas (ver Capítulo 4). Para ello se trabajó una sección dedicada a estas, como se muestra en la figura: 6.12; esta es la lista de modelos instalados y configurados de manera nativa, ellos ingresan mediante una base de datos de MySQL, con el fin de que en futuros desarrollos se contemple la posibilidad de realizar una interfaz dedicada a la inserción de nuevos modelos y variables para cada usuario.

Al seleccionar uno de los modelos presentados en esta sección (ver figura: 6.13) se puede apreciar la pestaña de bienvenida, explicando cómo se realizó el sistema y con qué parámetros cuenta el panel de posibilidades que se muestra en la sección izquierda para ordenadores y en la sección superior para móviles.

Para poder trabajar sobre ellas de manera más cómoda se creó una plantilla general para la visualización al usuario que consta de 4 zonas:

- En la parte izquierda aparece el panel primario los parámetros pre configurado

Figura 6.12: Lista de modelos instalados en la plataforma

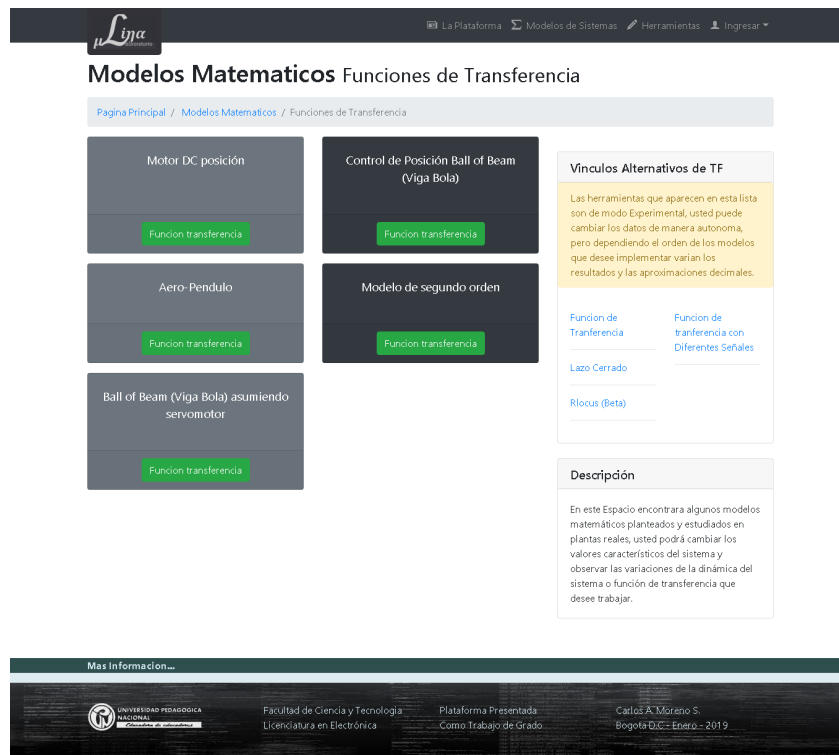
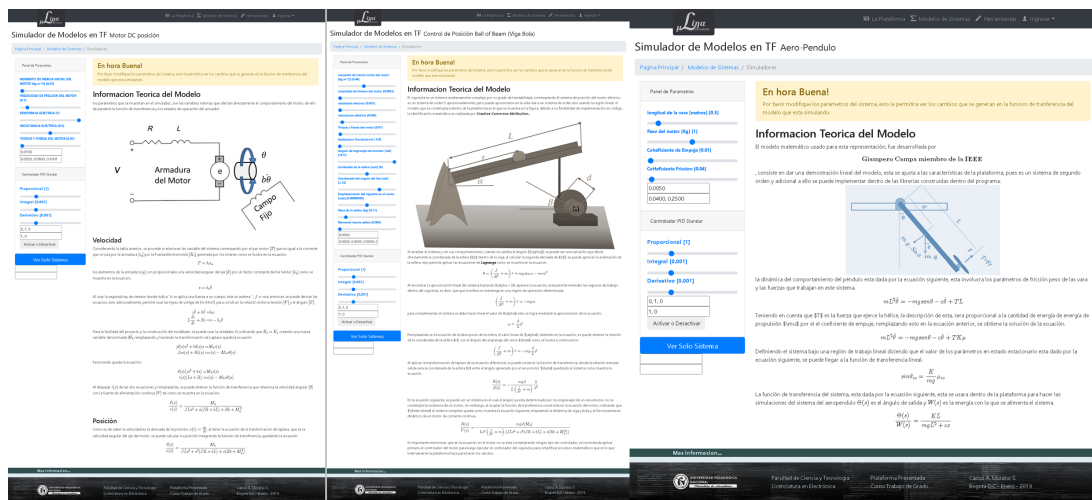


Figura 6.13: Panel de bienvenida de cada uno de los modelos nativos dentro de la plataforma.



del sistema, estos se representan con una barra de desplazamiento.

- El controlador PID por defecto esta desactivado, se encuentra debajo del panel primario izquierdo, cuenta con 3 slider que son: el proporcional y el integral y

el derivativo, adicionalmente cuenta con el botón apagado y encendido para el controlador $C(s)$.

- Panel de función de transferencia activa, este es el cuadro verde donde se muestra la información del sistema que se está computando, a medida que se varían los parámetros del controlador y de la planta esta varia.
- Panel de resultados, muestra una serie de pestañas, dependiendo la selección que el usuario marque mostrara las diferentes características del sistema que seleccione, en caso de estar la planta activa, muestra los resultados de la planta en lazo abierto sin controlador y luego la planta en lazo cerrado con controlador activo.

Sección de modelos de sistemas sin controlador activo.

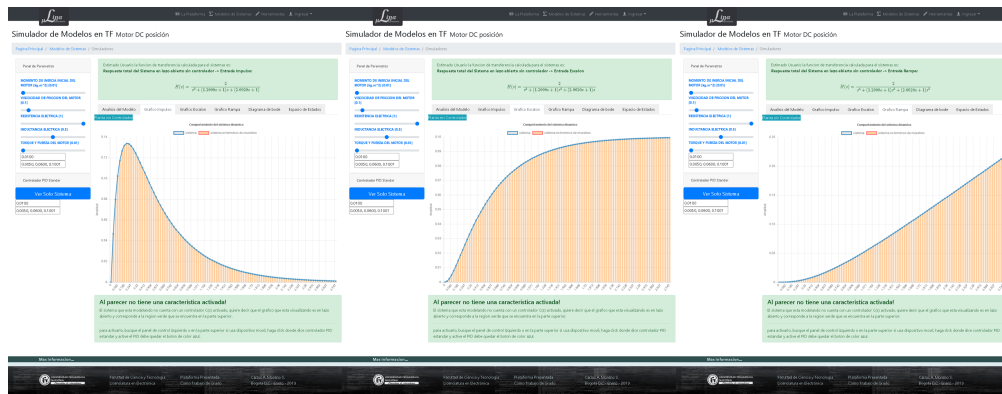
A continuación se verá la forma gráfica de la aplicación para los modelos propuestos, en este caso la vista

En la Figura: 6.14, se puede observar la interfaz gráfica del sistema modelado, se puede apreciar en la primera parte que solo se encuentra una columna con datos, mientras la otra tira una alerta naranja indicando que el controlador esta desactivado. En la Figura: 6.15, se muestran los gráficos del sistema modelado, ante las señales, impulso, escalón y rampa, en la sección de abajo muestra una alerta verde indicando que no está activado el lazo cerrado, adicionalmente le da unas orientaciones para poder apreciar mejor los resultados de la aplicación.

Figura 6.14: Resultado del modelo en lazo abierto computado después de dar en el botón "ver solo sistema".



Figura 6.15: Sección de gráficas ante los diferentes estímulos de entrada de un sistema seleccionado el lazo abierto



Sección de modelos de sistemas con controlador activo.

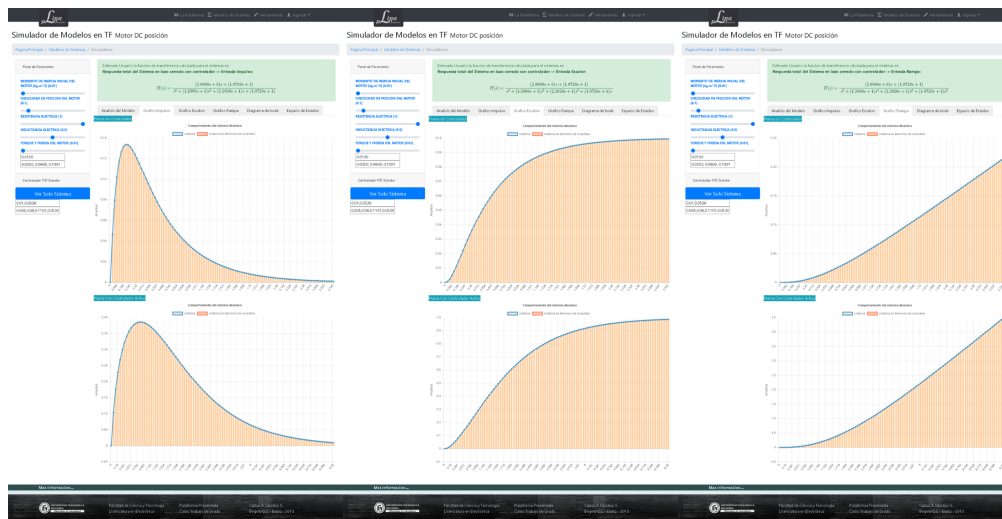
La interfaz al momento de que le es activada la acción del controlador, muestra la planta en lazo abierto sin controlador en la primera columna, en la segunda columna muestra la planta con lazo cerrado indicando que bajo las configuraciones que se están tomando los cálculos y sus respectivas soluciones, en la figura: 6.16, se muestra los resultados con controlador activado, la plataforma al usuario le indica los cálculos que está haciendo en algunos casos. Adicionalmente se muestra la comparación gráfica entre la planta en lazo abierto sin controlador y la planta

Figura 6.16: presentación la plataforma de bienvenida para manejar las variables de los modelos nativos.



en lazo cerrado con controlador, en la figura: 6.17, se puede observar el modelo original y el sistema ya con el controlador, a medida que se cambian los parámetros tanto del PID como de la planta, los gráficos van actualizando la información. En

Figura 6.17: presentación la plataforma de bienvenida para manejar las variables de los modelos nativos.



todos los sistemas modelados se activa una sección donde se muestra la pestaña de variables de estado, este solo muestra el sistema en términos de sus formas canónicas: controlable y observable, es importante que para futuros trabajos de

grados se pueda hacer uso de las otras formas canónicas ya que se requiere usar librerías que permitan trabajar transformaciones de álgebra lineal, para poder hacer librerías de uso general y aplicables a múltiples sistemas, que es lo que se realizó con las demás clases que usa el paquete de herramientas de microlablina.

Conclusiones y Recomendaciones

Conclusiones

Una vez culminada la construcción de las librerías pilares, se puede realizar la implementación de las plantas o modelos matemáticos propuestos dentro de la plataforma. Dicho desarrollo resultó una gran ventaja para el manejo de sistemas y detección de errores a la hora de compilación, obteniendo como resultado las clases que transforman arreglos delimitados por comas en polinomios para trabajar las funciones de transferencia.

Con el fin de verificar los resultados de las plantas propuestas se usó el software MATLAB, como herramienta que facilitó el contraste de resultados, evidenciando el logro de la meta propuesta en los objetivos para la simulación de sistemas.

La ventaja principal de la plataforma, consiste en trabajar y acoplarse a diferentes dispositivos, usando elementos responsive, que permiten ajustar el diseño de la aplicación a las diferentes plataformas y navegadores web, sin afectar el funcionamiento del dispositivo (celular, tablet o pc). Además, la página web no requiere

instalar programas adicionales, pues la idea fundamental es de uso rápido y flexible y de solo consulta.

Otra de las virtudes de la plataforma, es que se pueden implementar otros modelos haciendo uso de la ecuación de función de transferencia de forma polinómica extendida y separada por términos, mostrando las principales características del sistema

Una característica principal, que permite el graficado del comportamiento de sistemas, es que internamente posee una librería que hace transformaciones discretas que permite al sistema introducido enviarle diferentes señales o estímulos (Chirp, Seno, Triangular, Cuadrada, Escalón) a la planta para ver su comportamiento ante ellas.

Al crear la base de datos en MySQL como gestor, se realiza la construcción de formularios, donde funciona como herramienta para guardar preguntas y respuestas para las diferentes personas que usen la plataforma y deseen verificar sus conceptos, estas preguntas se hacen tipo test y están abiertas al todo el público registrado.

La herramientas más poderosas de la plataforma es la que se encuentra en el menú superior en la sección de herramientas llamada: "calcular la función de transferencia", debido a que permite hacer operaciones con polos repetidos en

sistemas de más de segundo orden, implementando el método de separación de fracciones parciales (heaviside), donde se requirió el diseño de librerías adicionales tales como operaciones con complejos, lectura de polinomios, límites y derivadas, para hacer funcionar esta característica.

Recomendaciones

Para futuras actualizaciones o implementaciones del laboratorio se recomienda una actualización de las librerías que trabajan con raíces y LGR, es menester adicionar más modelos matemáticos para hacer una aplicación más potente e interactiva para el usuario.

Se requiere optimizar las librerías de desarrollo gráfico para permitir la fluidez de los diferentes módulos que tiene la aplicación, es importante aplicar una implementación de la misma con estudiantes para futuros trabajos de grado.

Las herramientas matemáticas usadas en esta aplicación, contemplan la posibilidad para hacer laboratorios remotos, teniendo en cuenta que las características de toma de datos del navegador y las especificaciones establecidas anteriormente por cada librería, permiten una interacción con librerías que trabajen muestreos y cálculos desde servidor usando ajax.

Es importante mejorar el diseño visual de las librerías usadas con chartjs, ya que

la aplicación cuenta con herramientas para hacer gráficas más detalladas, pero esto consume mucho recurso del equipo del cliente así que requiere optimizar el código para su implementación.

Bibliografía

[Brooks, 2011] Brooks, D. (2011). *Guide to HTML, JavaScript and PHP: For Scientists and Engineers*. SpringerLink : Bücher. Springer London.

[Chapra and Canale, 2015] Chapra, S. and Canale, R. (2015). *Metodos numericos para ingenieros (7a. ed.)*. McGraw-Hill Interamericana.

[Delgado G, 2009] Delgado G, M. A. (2009). LABORATORIO VIRTUAL DE CONTROL INTELIGENTE. *Revista Educación en Ingeniería*, No. 8(1900-8260):102–110.

[Granados, 1999] Granados, L. F. M. (1999). Linea de inteligencia artificial y procesos de razonamiento. *Universidad Pegagogica Nacional*, 5.

[Kaçar and Bayilmis, 2013] Kaçar, S. and Bayilmis, C. (2013). A web-based educational interface for an analog communication course based on MATLAB builder NE with webfigures. *IEEE Transactions on Education*, 56(3):346–354.

[Myer, 2008] Myer, T. (2008). *Professional CodeIgniter*. Wrox Professional Guides. Wiley.

- [Ogata, 2004] Ogata, K. (2004). *Sistemas de control en tiempo discreto*. Pearson Educacion, Mexico, 4a.ed. edition.
- [Ricardo et al., 2005] Ricardo, C., Rafael, C., and Fausto, P. (2005). Desarrollo básico de un Laboratorio Virtual de Control de Procesos basado en Internet. *Accion Pedagogica, DOSSIER*, pages 58–65.
- [Tippens, 2010] Tippens, P. E. (2010). *Fisica I conceptos y aplicaciones*. McGraw-Hill Interamericana.
- [Vallv and Costa, 2007] Vallv, J. and Costa, R. (2007). Laboratorio Virtual para prácticas de Control por Computador. *Universitat Politècnica de Catalunya*.
- [Willsky, 1998] Willsky, A. S. (1998). *Senales y Sistemas*. Prentice-Hall Hispanoamericana, Mexico, 2a. ed. edition.

Apéndice A

Licencias de Terceros y Términos Legales

A.1. MIT

Esta licencia concede permiso, de forma gratuita, a cualquier persona que obtenga una copia de este software y los archivos de documentación asociados (el "Software"), para tratar en el Software sin restricciones, incluidos, entre otros, los derechos para usar, copiar, modificar, fusionar, publicar, distribuir, sub-licenciar y / o vender copias del software, y para permitir a las personas a quienes se refiere el software amueblado para hacerlo, sujeto a las siguientes condiciones: El aviso de copyright anterior y este aviso de permiso se incluirán en todas las copias o porciones sustanciales del software.

El software se proporciona tal como el programador o los desarrolladores consideren correcto, sin garantía de ningún tipo, expresa o implícito, incluidas, entre otras, las garantías de comercialización, aptitud para un propósito particular y no infracción. En ningún caso la los autores o los titulares de los derechos de autor

serán responsables de cualquier reclamación, daños u otros responsabilidad, ya sea en una acción contractual, agravio o de otro tipo, derivada de fuera de o en relación con el software o el uso u otros tratos en el software.

A.2. GNU/GPL

La Licencia Pública General, está basada en la reestructuración de la GNU o paquetes de software libre, pero contiene una cláusula adicional que permite a los usuarios interactuar con el programa licenciado a través de una red para recibir el código fuente de ese programa. Recomendamos que se considere utilizar la licencia AGPL de GNU para cualquier programa que vaya a ejecutarse a través de una red. La última es la versión 3; se resalta la implementación de esta, por la utilización de los certificados SSL que permiten cifrar y codificar la información suministrada a los usuarios que trabajen con la plataforma MICROLABLINA, sin contar que la aplicación web está trabajando bajo sistemas operativos Linux y con soporte del servidor que cuenta con esta licencia de manera nativa.

A.3. let's encrypt, uso de certificados SSL

Let's Encrypt es una autoridad de certificación (CA) libre, automatizada y abierta, dirigida al beneficio público. Es un servicio proporcionado por el Grupo de Investigación de Seguridad de Internet (ISRG). Se entrega a la gente los certificados digitales que necesitan para habilitar HTTPS (SSL/TLS) para sitios web,

de forma gratuita, de la manera más fácil que podemos. Hacemos esto porque queremos crear una web más segura y respetuosa con la privacidad.

Los principios clave detrás de Let's Encrypt son:

Gratis: Cualquiera que posea un nombre de dominio puede usar Let's Encrypt para obtener un certificado de confianza a coste cero. **Automático:** El software que se ejecuta en un servidor web puede interactuar con Let's Encrypt para obtener sin problemas un certificado, configurarlo de forma segura para su uso y encargarse automáticamente de su renovación. **Seguro:** Let's Encrypt servirá como plataforma para avanzar en las mejores prácticas de seguridad TLS, tanto del lado de la CA como ayudando a los operadores del sitio a asegurar adecuadamente sus servidores. **Transparente:** Todos los certificados emitidos o revocados serán registrados públicamente y estarán disponibles para que cualquiera los inspeccione. **Abierto:** El protocolo de emisión y renovación automática se publicará como un estándar abierto que otros pueden adoptar. **Cooperativa:** Al igual que los propios protocolos de Internet subyacentes, Let's Encrypt es un esfuerzo conjunto para beneficiar a la comunidad, más allá del control de cualquier organización. En la plataforma microblabina, la implementación de estos recursos permite garantizar que la aplicación trabajara de manera correcta con los navegadores de uso cotidiano, estableciendo un puente con la empresa cloudflare que es la entidad encargada de suministrar este tipo de licenciamientos y permisos gratuitos acompañado de las licencia de software libre AGPU/GPL.

A.4. MathWorks MATLAB Licencia Estudiantil, para ejecución de pruebas en ambientes académicos

En este caso para la corroboración de los datos obtenidos con la plataforma microlablina se usó la licencia Campus-Wide License, esta se implementó con el uso colaborativo de un estudiante de ingeniería en control de últimos semestres de la universidad distrital, llamado Norbey Bustamante, la utilización de esta consiste en el uso ilimitado de los productos de MATLAB y Simulink por parte de todo el alumnado, el Profesorado, el personal y los investigadores de una institución o campus universitario, en cualquier dispositivo dependiendo de los acuerdos internos dentro del contrato interno con cada facultad que desee adquirir el software educativo, permitiendo la utilización de recursos para apoyar la enseñanza, el aprendizaje y la investigación colaborativa, aprovechando el acceso abierto y escalado para los programas de MATLAB y las simulaciones de Simulink a clusters, nubes y centros de computación de alto rendimiento (HPC).

En la utilización de este software, se garantiza que no se usa ninguna sección de código del programa para uso interno de microlablina, este solo se implementa para cotejar resultados y se emplea netamente con fines educativos.

Apéndice B

Herramientas Matemáticas

En esta sección se muestra de manera general algunos elementos matemáticos trabajados dentro de la plataforma pero que no son tan relevantes, debido a que solo se implementan como caso de emergencia si el compilador principal encuentra un error en algunas de las librerías.

B.1. MÉTODO DE MULLER

El método de Müller aplica de manera más detallada el método secante, pero utiliza la interpolación cuadrática entre tres puntos en lugar de la línea y la interpolación entre dos. La resolución de los ceros de la cuadrática permite que el método encuentre pares complejos de raíces. Dadas tres conjeturas previas para la raíz x_{i-2} , x_{i-1} , x_i y los valores del polinomio $P(x)$ en esos puntos, el siguiente Aproximación x_{i+1} es producida por las siguientes fórmulas expresadas en la ecuación:

B.1

$$\begin{aligned}
 q &\equiv \frac{x_i - x_{i-1}}{x_{i-1} - x_{i-2}} \\
 A &\equiv qP(x_i) - q(1 + q)P(x_i - 1) + q^2P(x_i - 2) \\
 B &\equiv (2q + 1)P(x_i) - (1 + q)^2P(x_{i-1}) + q^2P(x_{i-2}) \\
 C &\equiv (1 + q)P(x_i)
 \end{aligned} \tag{B.1}$$

Donde el signo del denominador se elige para hacer su valor absoluto por el módulo, haciendo este lo más grande posible. No puede iniciar las interacciones con ninguno de los tres valores de x que desee, Tres valores igualmente espaciados en el eje real. Tenga en cuenta que debe tener en cuenta la posibilidad de un denominador complejo, la aritmética compleja posterior, en la implementación del método, ayuda a la normalización de los valores que cambian con cada una de las iteraciones que dependen del orden del polinomio..

$$x_{i+1} = x_i - (x_i - x_{i-1}) \frac{\sqrt{2C}}{B \pm \sqrt{B^2 - 4AC}} \tag{B.2}$$

Este método a veces se usa para encontrar ceros complejos de funciones analíticas (no solo polinomios) en el plano complejo, pero a nivel computacional consume mucho recurso debido a la cantidad de bucles para analizar la gráfica.

Apéndice C

Segmentos de Códigos Fuente

La mayoría de código fuente que verá en este apéndice será el encargado de hacer funcionar la plataforma, son fragmentos que dependiendo de cómo estén estos estructurados funcionarán las librerías diseñadas para las operaciones matemáticas.

C.1. Código Generador de Polinomios

Código 1: Algoritmo en PHP, para generar polinomios de N orden en la plataforma microlab.

```
1     private function operador($orden, $ac=1){  
2         $oper=array();  
3         $o=1;  
4         if($ac==1){  
5             for($i=0; $i<=$orden ; $i++){  
6                 $a=$o %2;  
7                 if($a==1) $a=1;  
8                 else $a=-1;  
9                 $oper[$i]=$a;  
10                $o++;  
11            }  
12        }  
13    }
```

```
11     }
12   } else {
13     for ($i=0; $i<=$orden ; $i++){
14       $oper[$i]=1;
15     }
16   }
17   return $oper;
18 }
19 public function Tpaschalz($orden , $ac=1){
20   $pasc=array ();
21   $n=$orden+1;
22   $x=0; $i=0; $j=0;
23   $pascx=array ();
24   // valida el numero de lineas
25   for ($i=1; $i<=$n ; $i++){
26     //Construimos el triangulo de pascal
27     for ($j=$x; $j>=0; $j--){
28       if ($j==$x || $j==0){
29         $pasc[$j] = 1;
30       }
31       else
32       {
33         $pasc[$j] = $pasc[$j] + $pasc[$j-1];
34       }
35     }
36     $x++;
37     //Truco para imprimir el triangulo
38
39     $cams=$this->operador($x,$ac);
40     for($j=0; $j<$x; $j++) {
41       $pascx[$j]=$pasc[$j]*$cams[$j];
42     }
43   }
```

```

45         return $pascx;
    }

```

C.2. Código Lazo Cerrado

código: 2 Algoritmo en PHP, construir una función de transferencia en lazo cerrado bajo el análisis de la ecuación $p(s) = \frac{DA}{EB+DA}$.

```

1 public function CompTfCer($kc, $Td, $Ti, $a){
3
4     ___if ($a!=0){
5     ___if ($Ti!=0){
6         ___if ($Td!=0){
7             ___$num=array (($a*$kc*$Ti*$Td+$kc*$Ti*$Td), ($kc*$Ti+$a*$kc*$Td), $kc);
8             ___$den=array (($a*$Td*$Ti), $Ti, 0);
9         ___} else {
10            ___$num=array ($kc);
11            ___$den=array (1);
12            ___}
13        ___} else {
14            ___if ($Td!=0){
15                ___$num=array ($a*$kc*$Td, ($kc+$Td));
16                ___$den=array ($a*$Td, 1);
17            ___} else {
18                ___$num=array ($kc+$Td);
19                ___$den=array (1);
20            ___}
21        ___}
22    ___} else {

```

```

23 _____if ($kc !=0){
_____    if ($Ti !=0){
25 _____    if ($Td !=0){
_____        $num=array ($Ti*$Td*$kc , $kc*$Ti , $kc );
27 _____        $den=array ($Ti ,0);
_____    } else {
29 _____        // $num=array ($kc , $kc / $Ti );
_____        $num=array ($kc , $kc / $Ti );
31 _____        $den=array (1 ,0);
_____    }
33 _____} else {
_____    if ($Td !=0){
35 _____        // $num=array ($kc*$Td , $kc );
_____        $num=array ($Td , $kc );
37 _____        $den=array (1);
_____    } else {
39 _____        $num=array ($kc );
_____        $den=array (1);
41 _____    }
_____    }
43 _____} else {
_____    $num=array (1);
45 _____    $den=array (1);
_____    }
47 _____}

49
_____ $num=new polinomio ($num);
51 _____ $den=new polinomio ($den);
_____ return new tf ($num,$den , "G(s)");
53 _____}

55 _____public function LC($s1 , $s2){

```

```

___$a=$s1->num;
57 ___$b=$s1->den;
___$d=$s2->num;
59 ___$e=$s2->den;
___$da=$d->mulPor($a);
61 ___$eb=$e->mulpor($b);
___$den=$eb->sumPor($da);
63 ___return new tf($da,$den,"p($)" );
___}

```

C.3. Diagrama de Bode

Código 3: se encarga de hacer la gráfica de magnitud y fase, adicionalmente calcular las frecuencias en las que trabaja la función de transferencia; se usa como ayuda dentro del rlocus para ajustar la gráfica en sistemas de polos conjugados.

```

1 public static function plotFase($num,$den){
___   $polos=Raices::calculaRaices($den);
3 ___   $ceros=Raices::calculaRaices($num);
___   $frecs=self::frecuencias($polos);
5     $maxF = max($frecs);
     $ninF = min($frecs);
7     $a=$ninF/1000;
     $b=$maxF*20;
9 ___   if($ninF==0){
___     $a=1/$b;
11 ___   }
___   $a=abs($b-$a)/10000;
13     $bode='[';

```



```

15     $hcx=0;
16     $hcc=0;
17     $frec=$a/10000;
18     if (count($num)>1){
19         do{
20             $frec =. $frec . " | ";
21             $bode=$bode . ",";
22             $num = self::calculador1($ceros,$frec);
23             $den = self::calculador1($polos,$frec);
24             $Hs=-$num+$den;
25             $frec=$frec+$a;
26         } while ($frec <$b);
27     } else {
28         $mag=$num[0];
29         $numx=new Complejo($mag,0);
30         do{
31             $bode=$bode . ",";
32             $num = atan($numx->i/$numx->r);
33             $den = self::calculador1($polos,$frec);
34             $frec=$frec+$a;
35             $hc=- self::rad2deg($num) + self::rad2deg($den) ;
36             $bode=$bode . "{x:$frec,y:$hc,}";
37         } while ($frec <=$b);
38     }
39     ___$bode= $bode . "]"";
40     ___return $bode;
41 }
42
43 ___private static function frecuencias($polos) {
44     if($polos==null) $polos=$this->polos;
45     $z0=array();
46     $z0_c=array();

```

```

47     $real=array ();
        $x=0;
49     $frecuencias=array ();
        for($i=0; $i<count($polos); $i++){
51         if($polos[$i][1]>0){
                $z0[]=new Complejo($polos[$i][0], $polos[$i][1]);
53         } else if($polos[$i][1]<0){
                $z0_c[]=new Complejo($polos[$i][0], $polos[$i][1]);
55         } else if($polos[$i][1]==0) $real[]=$polos[$i][0];
        }

57

        if(count($z0)==count($z0_c)){
59             for($x=0;$x<count($z0);$x++){
                    $a=Complejo::add($z0[$x], new Complejo($z0[$x]->r, -$z0[$x]->i));
61                    $b=Complejo::mul($z0[$x], new Complejo($z0[$x]->r, -$z0[$x]->i));
                    $frecuencias[]=abs(sqrt($b->r));
63            }
        }

65     for($x=0;$x<count($real);$x++){
            $frecuencias[]=abs($real[$x]);
67     }
        return $frecuencias;
69 }

71 ____public static function plotBode($num,$den){
    ____ $polos=Raices::calculaRaices($den);
73 ____ $ceros=Raices::calculaRaices($num);
    ____ $frecs=self::frecuencias($polos);
75     $maxF = max($frecs);
        $ninF = min($frecs);
77     $a=$ninF/1000;
        $b=$maxF*20;
79 ____ if($ninF==0){

```

```
—      $a=1/$b;
81 —    }
—      $a=abs($b-$a)/5000;
83 —      $bode='[';
—      $hcx=0;
85      $hcc=0;
—      $frec=$a/10000;
87      if(count($num)>1){
—        do{
89 —          $bode=$bode.", "; $hcc=self::calculador2($ceros,$frec);
—          $hcx=self::Calculador2($polos,$frec,1);
91 —          $hc= $hcc+$hcx;
—          $bode=$bode."{x:$frec,y:$hc,}";
93 —        } while($frec <=($b));
—      } else {
95 —        $mag=$num[0];
—        $h=20*log10(abs($mag));
97 —        do{
—          $hcx=0;
99 —          $bode=$bode.", "; $hcx=self::Calculador2($polos,$frec,1);
—          $frec=$frec+$a;
101 —          $hc= $h+$hcx;
—          $bode=$bode."{x:$frec,y:$hc,}";
103 —        } while($frec <$b);
105 —      }
—      $bode= $bode."]";
107 —      return $bode;
—    }
```

C.4. Derivadas

Código 4: Este código se encarga de generar las derivadas de un polinomio y regresa otro polinomio, en caso de una función de transferencia, la secciona en dos polinomios numerador y denominador, y entrega los resultados en términos de arreglo de objetos de tipo polinomio,

```

2 public static function derivCoef($coef=array(), $grado=0){
   ___$coefm=$coef*$grado;
4 ___return $coefm;
   ___}
6
   ___// entrada tipo array
8 ___public static function derivPol($polinomio){
   ___$grado=count($polinomio);
10 ___$poldev=array();
   ___$gradox=$grado-1;
12 ___$gradod=0;
   ___for($i=0; $i<$grado; $i++){
14 ___   ___if($gradod <= ($grado-2))
       ___       $poldev[$gradod]=self::derivCoef($polinomio[$i], $gradox);
16 ___   ___$gradod +=1;
       ___   ___$gradox -=1;
18 ___   ___}
   ___return $poldev;
20 ___}

22 ___// entradas de objeto polinomio
   ___public static function derivfrac($polix, $poli2){
24 ___   $ap=new Polinomio(self::derivPol($polix->polinomio));
   ___   $bp=new Polinomio(self::derivPol($poli2->polinomio));

```

```

26  __ $a=$polix ;
    __ $b=$poli2 ;
28  __ $numpi=$ap->mulPor($b)->resPor($bp->mulpor($a));
    __ $denpi=$b;
30  __ $denpi=$denpi->mulPor($b);
    __ $fraccion=array($numpi->polinomio,$denpi->polinomio);
32  __ return $fraccion;
    __}

```

C.5. Limites

Código 5: Este código es importante debido a que de este depende que las librerías para hacer fracciones parciales funcionen, trabaja con números complejos y conjugados, dependiendo la creación del objeto en el kernel de la plataforma.

```

1  //entrar el polinomio como arreglo y entrar la tendencia como objeto complejo
3  __public static function limitePolCoplex($polinomio,$complejo){
    ____$grado=count($polinomio);
5  ____$gpol=$grado-1;
    ____$rescomp=new Complejo();
7  ____$real=new Complejo();
    ____$for($i=0;$i<$grado;$i++){
9  ____$rescomp=Complejo::muldo($polinomio[$i],Complejo::powc($complejo,$gpol));
    ____$real=Complejo::add($real,$rescomp);
11 ____$gpol--=1;
    __}
13
    __return $real;

```

```
15  __}

17  //entrar el complejo como arreglo y entrar la tendencia como objeto complejo
    __public static function limiteCCoplex($Cpolinomio, $complejo){
19  ____$grado=count($Cpolinomio);
    ____$gpol=$grado-1;
21  ____$rescomp=new Complejo();
    ____$real=new Complejo();
23  ____$for($i=0;$i<$grado;$i++){
    ______$rescomp=Complejo::mul($Cpolinomio[$i],Complejo::powc($complejo,$gpol));
25  ______$real=Complejo::add($real,$rescomp);
    ______$gpol-=1;
27  ____}
    __return new Complejo($real->r,$real->i);
29  __}

31  __public static function limitepol($polinomio,$tiende=0){
    ____$grado=count($polinomio);
33  ____$gpol=$grado-1;
    ____$vali=0;
35  ____$for($i=0;$i<$grado;$i++){
    ______$val=$polinomio[$i]*pow($tiende,$gpol);
37  ______$vali=$vali+$val;
    ______$gpol-=1;
39  ____}
    __return $vali;
41  __}
```

C.6. Fracciones Parciales

Código 4: este usa el método de heaviside, es parte de la librería auxiliar para convertir una función de transferencia en términos de la frecuencia a términos del tiempo.

```

1   public function __construct($numerador=0,$den=0){
      ___$this->num=$numerador;
3   ___$this->denfactor=$den;
      ___$this->roots=Raices::calculaRaices($this->denfactor);
5   ___}

7   public function corFrac($dato=array()){
      $num=$dato[$i][0];
9   $den=$dato[$i][1];
      for($i=0; $i<count($reentrega); $i++){
11          $data=count();
          }
13  }

___public function OutFracPar(){
15 ___$caso=1;
      ___$entrega=array(array());
17 ___$entregax=array(array());
      ___//debe estar en terminos de arreglo unidimensional
19 ___for($i=0; $i<count($this->roots); $i++){
      ___if($this->roots[$i][1]!=0){
21 ___$caso=2;
      ___break;
23 ___}
      ___}
25 ___switch ($caso) {
      ___case 1: //polos reales no repetidos

```

```

27 _____$entrega=$this->caso1();
           $reentrega=$entrega;
29 _____break;
           _____case 2: //polos complejos no repetidos y conjugados
31 _____$entrega=$this->metodo2();
           _____// actualizar entrega
33 _____break;
           _____default:
35 _____return null;
           _____break;
37 _____}
           _____return $entrega;
39 _____}

41 _____// caso 1 heaviside de polos reales no repetidos
           _____public function caso1($rootsex){
43 _____$coefnum=array();
           _____$limitedenominador=array();
45 _____$polipart=array(array());
           _____$limitepolipart=array();
47 _____$A_n=array();
           _____$entregar=array(array());
49 _____for($i=0;$i<count($rootsex);$i++){
           _____$limitedenominador[$i]=limiteP::limitepol($this->num,$rootsex[$i]);
51 _____$polipart[$i][0]=1;
           _____$polipart[$i][1]=- $rootsex[$i];
53 _____}
           _____$mulcal=1;
55 _____for($i=0;$i<count($rootsex);$i++){
           _____$limitepolipart=1;
57 _____for($ia=0;$ia<count($rootsex);$ia++){
           _____$mulcal=limiteP::limitepol($polipart[$ia],$rootsex[$i]);
59 _____if($mulcal!=0)

```



```

_____ $limitepolipart=$limitepolipart*$mulcal;
61 _____}
_____ $A_n=$limitedenominador[$i]/$limitepolipart;
63 _____ $entregar[0][$i]=array(round($A_n,4));
_____ $entregar[1][$i]=$polipart[$i];
65 _____}
_____ return $entregar;
67 _____}

69 _____ public function caso2(){
_____ $raices_del_denominador_noRepetidas=array();
71 _____ $raices_del_denominador_repetidas=array();
_____ $numerador=$this->num;
73 _____}

75 // caso 3 de heaviside polos complejos no repetidos
_____ public function caso3(){
77 _____ $coefnum=array();
_____ $limitedenominador=array();
79 _____ $polipart=array(array());
_____ $limitepolipart=array();
81 _____ $A_n=new Complejo();
_____ $entregar=array(array());
83 _____ for($i=0;$i<count($this->roots);$i++){
_____ $pcomplejo=new Complejo($this->roots[$i][0],$this->roots[$i][1]);
85 _____ $limitedenominador[$i]=limiteP::limitePolCoplex($this->num,$pcomplejo);
_____ $polipart[$i][0]=new Complejo(1,0);
87 _____ $polipart[$i][1]=
_____ new Complejo(-$this->roots[$i][0],-$this->roots[$i][1]);
89 _____}
_____ $mulcal=1;
91 _____ $limitepolipart=new Complejo(1,0);
_____ $npol=array();

```

```

93 ____for ($i=0;$i<count($this->roots);$i++){
    ____if ($polipart[$i][1]->i!=0){
95 _____$limitepolipart=new Complejo(1,0);
    _____for ($ia=0;$ia<count($this->roots);$ia++){
97 _____$mulcal=limiteP::limiteCCoplex(
    _____$polipart[$ia],
99 _____new Complejo($this->roots[$i][0],$this->roots[$i][1])
    _____);
101 _____if ($mulcal->i!=0){
    _____$limitepolipart=Complejo::mul($limitepolipart,$mulcal);
103 _____}
    _____}
105 _____$A_n=Complejo::div($limitedenominador[$i],$limitepolipart);
    _____$A_n=new Complejo(round($A_n->r,4),round($A_n->i,4));
107 _____} else {
    _____$limitepolipart=new Complejo(1,0);
109 _____$mulcalx=array();
    _____for ($ia=0;$ia<count($this->roots);$ia++){
111 _____$na[0]=$polipart[$ia][0]->r;
    _____$na[1]=$polipart[$ia][1]->r;
113 _____$mulcal=limiteP::limiteCCoplex(
    _____$polipart[$ia],
115 _____new Complejo($this->roots[$i][0],$this->roots[$i][1]));
    _____if ($mulcal->r==0 && $mulcal->i==0){
117 _____} else {
    _____$mulcalx[$ia]=$mulcal;
119 _____$limitepolipart=Complejo::mul($limitepolipart,$mulcal);
    _____}
121 _____}
    _____$A_nx=Complejo::div($limitedenominador[$i],$limitepolipart);
123 _____$A_n=new Complejo(round($A_nx->r,4),0);
    _____}
125 _____$entregar[$i][0]=$A_n;

```

```

_____ $entregar[ $i ][1]= $polipart[ $i ];
127 _____}
_____ return $entregar ;
129 _____}

```

C.7. Laguerre: Raíces de Polinomio

Código 7: Se usa el método matemático de Laguerre (ver anexo: B), se usa este método por ser el más rápido de implementar matemáticamente en el ordenador, sin contar que usando la librería de derivadas y límites se pueden calcular fácilmente las raíces de polinomios de orden superior al segundo haciendo aproximación gráfica usando los puntos de infección y las tendencias hacia los puntos de corte.

```

public static function zroots($a, $m, $polish){
2 _____ $roots=array ();
_____ $ad=array ();
4 _____ $contaobj=count ($a);
_____ // ok
6 _____ for ($i=0; $i < $contaobj; $i++){
_____ $roots[ $i ]=new Complejo ();
8 _____}
_____ // ok
10 _____ for ($j=0; $j < 100; $j++){
_____ $ad[ $j ]=new Complejo ();
12 _____ }
_____ // ok
14 _____ for ($j=0; $j <=$m; $j++){
_____ $ad[ $j ]= $a[ $j ];

```

```
16 ____}

18 ____//funciona sin laguer
    ____    for ($j=$m;$j >=1;$j--) {
20 ____    $x=new Complejo();
    ____    static :: laguer($ad,$j, $x);
22 ____    $rrata=2*2e-6*abs($x->r);
    ____    if (abs($x->i) <= $rrata) $x->i=0;
24 ____    $roots[$j]=$x;
    ____    $b=$ad[$j];
26 ____    //codigo correcto sin laguer()

28 ____    //for ok
    ____    for ($jj=$j-1;$jj >=0;$jj--) {
30 ____    $c=$ad[$jj];
    ____    $ad[$jj]=$b;
32 ____    $b=Complejo::add(Complejo::mul($x,$b),$c);
    ____    }
34 ____}

36 ____//condicion funciona sin laguer
    ____    if($polish==true){
38 ____    for($j=1;$j <=$m;$j++){
    ____    static :: laguer($a,$m,$roots[$j]);
40 ____    }
    ____}

42

    ____//ok
44 ____    for ($j=2;$j <=$m;$j++) {
    ____    $x=$roots[$j];
46 ____    for ($i=$j-1;$i >=1;$i--) {
    ____    if ($roots[$i]->r <= $x->r){
48 ____    break;
```

```

_____}
50 _____$roots[$i+1]=$roots[$i];
_____ }
52 _____ $roots[$i+1]=$x;
_____ }
54 _____ return $roots;
_____}
56
_____const EPSS= 1e-7;
58 _____const MR= 8;
_____const MI= 10;
60 _____const MAXIT = 80;
_____const EPS = 2.0e-6;
62 _____const MAXM = 100;

64 _____private static function laguer($a, $m, $x){
_____ $err=0;
66 _____$frac = array(0.0,0.5,0.25,0.75,0.13,0.38,0.62,0.88,1.0);
_____// bandera error en php $x
68 _____for ($iter=1;$iter <=80;$iter++) {
_____ $b=$a[$m];
70 _____$err=Complejo::abs($b);
_____ $d=new Complejo(0,0);
72 _____$f=new Complejo(0,0);
_____ $abx=Complejo::abs($x);
74 _____for ($j=$m-1;$j >=0;$j--) {
_____ $f=Complejo::add(Complejo::mul($x,$f),$d);
76 _____ $d=Complejo::add(Complejo::mul($x,$d),$b);
_____ $b=Complejo::add(Complejo::mul($x,$b),$a[$j]);
78 _____ $err=Complejo::abs($b)+$abx*$err;
_____}
80 _____ $err = $err*1e-7;
_____ if (Complejo::abs($b) <= $err){

```

```
82 _____return ;
_____}
84
_____ $g=Complejo :: div ($d , $b);
86 _____ $g2=Complejo :: mul ($g , $g );
_____
88 _____ $h=Complejo :: sub ($g2 , Complejo :: muldo (2 , Complejo :: div ($f , $b)));
_____
90 _____ $sq=Complejo :: sqrtz (
_____ Complejo :: muldo (($m-1),
92 _____ Complejo :: sub (
_____ Complejo :: muldo ($m, $h) , $g2)
94 _____ ));
_____
96 _____ $gp=Complejo :: add ($g , $sq);
_____ $gm=Complejo :: sub ($g , $sq);
98 _____ $abp=Complejo :: abs ($gp);
_____ $abm=Complejo :: abs ($gm);
100 _____ if ($abp < $abm) $gp=$gm;
_____
102 _____ if (self :: FMAX($abp , $abm) > 0.0){
_____ $rata=new Complejo ($m, 0.0);
104 _____ $dx=Complejo :: div ($rata , $gp);
_____ } else {
106 _____ $rata2x=new Complejo (cos ($iter) , sin ($iter));
_____ $dx=Complejo :: mul (1+$abx , $rata2x);
108 _____ }
_____
110 _____ $x1=Complejo :: sub ($x , $dx);
_____ if ($x->r == $x1->r && $x->i == $x1->i){
112 _____ return ;
_____ // break;
114 _____ }
```

```

_____if ($iter % 10!=0){
116 _____$x->r=$x1->r;
_____ $x->i=$x1->i;
118 _____} else {
_____ $temp=Complejo::sub($x, Complejo::mul($frac[$iter/10], $dx));
120 _____$x->r=$temp->r;
_____ $x->i=$temp->i;
122 _____}
_____}
124 _____echo "<br>demasiadas iteraciones en laguer";
_____return;
126 _____}

```

C.8. Discretizacion: Tustin

Código 8: Esta sección de la librería, se encarga de evaluar y convertir una función de transferencia que está en términos de la frecuencia, a términos discretos, con el fin de simular una representación gráfica en tiempo real sin importar su orden.

```

1 private $nums;
_____ private $dens;
3 _____ private $vdiv;
_____ private $t;
5 _____ public function __construct(){
_____ $this->nums=array();
7 _____ $this->dens=array();
_____ $this->vdiv=0;
9 _____ $this->t=0.001;
_____ }

```

```

11 public function muestra($muestra){
    $this->t=$muestra;
13 }
        public function Entrar_sistema(&$num,&$den){
15     $this->nums=$num;
        $this->dens=$den;
17     }

19 public function A($mul,$orden){
    $sis=count($this->Tpascalz($orden));
21     $potencia=pow(2,$orden);
    foreach ($this->Tpascalz($orden) as &$i) {
23         $poxlim[]=$mul*$potencia*$i;
    }
25     return new polinomio($poxlim);
    }

27

29 public function B($orden){
    $sis=count($this->Tpascalz($orden,0));
    $potencia=pow($this->t,$orden);
31     foreach ($this->Tpascalz($orden,0) as &$i) {
        $poxlim[]=$potencia*$i;
33     }
    return new polinomio($poxlim);
35     }

37     public static function latexmultipol($polinomios){

39     $contapoli=count($polinomios);
    $texto='';
41     for($i=0;$i<$contapoli;$i++){
        $polin=$polinomios[$i];
43     if(count($polin)!=0) $texto=$texto."(".self::latexpoli($polin).")";

```



```
    }
45    return $texto;
    }

47

49    public static function mostrarPolinomio($polinomio){
        return '$$ \dot p(x) ='.self::latexpoli($polinomio). '$$';
51    }

        public function Tpaschalz($orden, $ac=1){
53
            $pasc=array();
55            $n=$orden+1;
            $x=0; $i=0; $j=0;
57            $pascx=array();
            // valida el numero de lineas
59            for ($i=1; $i<=$n ; $i++)
                {
61                    //Construimos el triangulo de pascal
                    for ($j=$x; $j>=0; $j--)
63                        {
                            if($j==$x || $j==0)
65                                {
                                    $pasc[$j] = 1;
67                                }
                                else
69                                {
                                    $pasc[$j] = $pasc[$j] + $pasc[$j-1];
71                                }
                            }
                        }
73                    $x++;
                    $cams=$this->operador($x,$ac);
75                    for($j=0; $j<$x; $j++)
                        {
```

```

77         $pascx [ $j ]= $pasc [ $j ]*$cams [ $j ];
           }
79     }
       return $pascx;
81 }

83     public function mulden () {
       $gden = $this ->dens ->grado ();
85     $gradpol=0;
       for ( $i=$gden; $i >=1; $i -- ) {
87         if ( $i==$gden ) {
           $mulAP [ $i ]= $this ->A ( $this ->dens ->polinomio [ $gradpol ], $i );
89         } else {
           $mulAP [ $i ]= $this ->A (
91             $this ->dens ->polinomio [ $gradpol ],
           $i ) ->mulPor (
93             $this ->B ( $gradpol )
           );
95         }
           $gradpol++;
97     }
       $k=new polinomio ( array ( $this ->dens ->polinomio [ $gden ] ) );
99     $mulAP [ 0 ]= $k ->mulPor ( $this ->B ( $gden ) );
       return $mulAP ;
101 }

103     public function sumapolnum () {
       $conta=count ( $this ->mulden () ) - 1;
105     $p=$this ->mulden () [ 0 ];
       for ( $i=1; $i <= $conta; $i ++ ) {
107         $p=$p ->sumPor ( $this ->mulden () [ $i ] );
           }
109     $this ->vdiv=$p ->polinomio [ 0 ];

```

```
111     $conta=count($p->polinomio)-1;
112     for($i=0; $i<=$conta; $i++){
113         $polfin[$i]=round($p->polinomio[$i]/$this->vdiv,4);
114     }
115     return new polinomio($polfin);
116 }
```