



PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito final para obtener el Título
de INGENIERO DE SISTEMAS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA EL SIMULADOR DE EVENTOS DISCRETOS GALATEA.

Por

Br. Erik Velásquez

Tutor: Dr. Jacinto Dávila

Tutor: Prof. Kay Tucci

Octubre 2016

©2016 Universidad de Los Andes Mérida, Venezuela

Diseño e implementación de un Sistema Web para el Simulador de Eventos Discretos GALATEA.

Br. Erik Velásquez

Proyecto de Grado — Sistemas Computacionales, 61 páginas

Resumen: Se buscó implementar y desplegar como servicio, una aplicación utilizada para modelar y simular eventos discretos en sistemas distribuidos, utilizando como medio los servicios web. El objetivo principal de este proyecto consistió en buscar, mediante el uso de un sistema web, una solución al problema existente entre los usuarios y usuarias del Centro de Simulación y Modelado (CESIMO) de la Universidad de los Andes, que suelen tener muchas dificultades para ejecutar los modelos de simulación en el simulador GALATEA, especialmente al momento de configurar y activar todas sus funcionalidades. Mediante el patrón de diseño de software modelo vista controlador (MVC) y usando el framework de desarrollo web Django; se desarrolló el sistema web, que usando como motor principal a GALATEA, permite crear, controlar y simular proyectos de simulación. Al igual se creó una implementación de gestión de archivos, la cual permite compartir y controlar archivos entre los distintos usuarios y usuarias del CESIMO.

Palabras clave: Servicios Web, Sistemas Distribuidos, Web Semántica, Arquitectura Orientada a Servicios, Simulación.

Nadie.

Nadie.

Nadie.

Índice

Índice de Tablas	vi
Agradecimientos	vii
1 Introducción	1
1.1 Antecedentes	1
1.2 Planteamiento del problema	2
1.3 Justificación	3
1.4 Alcance	3
1.5 Objetivos	4
1.5.1 Objetivo general	4
1.5.2 Objetivos específicos	4
1.6 Metodología	4
1.7 Estructura del documento	5
2 Marco teórico	6
2.1 Simulación Distribuida	6
2.2 Simulación de Eventos Discretos	7
2.3 GALATEA	7
2.3.1 La arquitectura de la plataforma Galatea	8
2.4 Servicio Web	8
2.5 Arquitectura Orientada a Servicios	9
2.6 Herramientas usadas para el desarrollo del sistema	9
2.6.1 Modelo Vista Controlador (MVC)	9
2.6.2 Django	11
2.6.3 Bases de Datos	12
2.6.4 JSON	13
2.6.5 AJAX	13

3	Desarrollo del Sistema WEB	14
3.1	Diseño de la aplicación WEB	15
3.1.1	Arquitectura	15
3.1.2	Capa 1: Presentación	16
3.1.3	Capa 2: Servidor Web	16
3.1.4	Capa 3: Integración de Procesos	17
3.1.5	Capa 4: Datos	17
3.1.6	Diseño de la Base de Datos	17
3.1.7	Diseño de Pantallas	21
3.1.8	Diseño de URLs	23
3.2	Gestión de Archivos	24
3.2.1	Configuración	24
3.2.2	Control de Archivos	24
3.2.3	Control de Carpetas	24
3.2.4	Control de Permisos	24
4	Integración de GALATEA	25
5	Pruebas	26
5.1	Criterios y variables	26
5.1.1	Estudio preliminar de criterios y variables	26
5.1.2	Pruebas preliminares	39
5.2	Construcción del modelo	44
5.2.1	Definición del sistema de inferencia	44
5.2.2	Estructura del sistema difuso	45
6	Conclusiones	57
6.1	Recomendaciones	58
	Bibliografía	60

Índice de Tablas

3.1	Tabla de enrutamiento del servidor web.	16
3.2	Entidades relacionadas al manejo de usuarios.	20
3.3	Entidades relacionadas al manejo de archivos y carpetas.	20
3.4	Entidades relacionadas al manejo de archivos y carpetas compartidos.	20
3.5	Entidades relacionadas al manejo de logs.	21
5.1	Datos recogidos por los departamentos de la EISULA. Año 2014.	28
5.2	Comparación de diferencias basada en datos de la EISULA. Año 2014.	29
5.3	Datos de entrada. Porcentaje de actividades del departamento “x”.	33
5.4	Pesos de cada área.	33
5.5	Indicadores resultantes.	33
5.6	Algunas reglas sobre los indicadores.	34
5.7	Datos para los Departamentos A y B.	39
5.8	Resultados y comparación de los Departamentos A y B.	42
5.9	Datos de entrada. Segunda prueba.	43
5.10	Asignación de cargos.	54

Agradecimientos

Nadie.

Nadie.

Nadie.

Nadie.

Nadie.

Nadie.

Nadie.

Nadie.

Capítulo 1

Introducción

El siguiente extracto, introduce una descripción de cómo se ha desarrollado el servicio que prestará una aplicación basada en simulación de eventos discretos, que estará fundamentado en GALATEA (Uzcátegui et al., 2011) , el cual es un software para simulación de sistemas multi-agentes producto de dos líneas de investigación: lenguajes de simulación basados en la teoría de simulación de Zeigler y agentes basados en lógica. El propósito es que sirva para proveer un servicio de simulación de eventos discretos al usuario desde cualquier punto. Así también, se pretende analizar conceptos de seguridad mediante el cual, ante una petición al computador (request), ésta sea procesada sólo si ha sido previamente validada.

1.1 Antecedentes

Como trabajos similares a esta propuesta se puede mencionar los realizado por (Uzcátegui et al., 2011) el cual relata la fase de desarrollo, cómo surge y hacia donde se proyecta GALATEA. Implementada como una plataforma libre de código abierto para simulación de sistemas multi-agente que incorpora estrategias de simulación bien conocidas con la que cualquier modelista o simulista puede ensayar dichas estrategias en problemas de simulación de sistemas complejos. Por otra parte, otro precedente tomado en cuenta ha sido (Rengifo, 2011), el cual trata de una tesis de pre-grado que expone el desarrollo de un servicio web para la Modeloteca del Sistema Nacional de Simulación. La misma consistió en buscar, mediante el uso de una aplicación web, la solución al problema existente en el Centro de Simulación y Modelado (CESIMO) de la Universidad de Los Andes, en el cual había dificultades para mantener un registro referente a los proyectos que allí se desarrollaban, lo que como resultado provocaba que a menudo se perdiera información referente a los mismos, ó que incluso se llevara a cabo proyectos de forma innecesaria, ya que trabajos parecidos habrían sido realizado antes. Resultados que, debido a la falta de un repositorio institucional compartido, resultan imposibles de reusar o integrar. A partir de estos dos antecedentes lo que se pretende es, tomando como base

GALATEA, adaptarlo como un servicio web en el cual el usuario pueda interactuar con el simulador, sin tener las limitaciones tales como las que pueden surgir al usar sistemas operativos o navegadores diferentes, incluso ante arquitecturas distintas. A su vez, de (Rengifo, 2011), se buscó obtener la experiencia ganada al desarrollar un Servicio Web en sí. Al igual se toma en cuenta a (Marcano, 2015), buscando implementar y desplegar como servicio, una aplicación utilizada para modelar y simular eventos discretos en sistemas distribuidos, utilizando como medio los servicios web. A pesar de que ambas experiencias desarrollaron soluciones, aún no es posible contar con la experiencia de uso remoto de GALATEA como un servicio Web configurado por expertos administradores, pero con todas las facilidades y características del simulador al alcance de cualquier usuario registrado. El modelo vista controlador (MVC) es un patrón de diseño de software, que separa los datos y la lógica de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. Este patrón de diseño es muy popular en el marco de aplicaciones web ya que su abstracción permite escribir software altamente desacoplado y fácil de mantener y escalar. Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo-vista-controlador. La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos. Django permite construir aplicaciones web rápidamente, gracias a su filosofía de baterías incluidas, es decir, que incluye una inmensa gama de características comunes a la mayoría de las aplicaciones web como las validaciones, autenticación de usuarios, manejo de sesiones entre muchos otros (Foundation, 2015). En su proyecto de grado (Pérez, 2016), utilizando MVC y Django como herramientas fundamentales desarrolló un sistema de monitoreo distribuido de enlaces de redes a través de un servicio web centralizado. Este servicio además hace énfasis en la visualización de los datos recolectados a partir de pruebas periódicas.

1.2 Planteamiento del problema

La preocupación por los sistemas distribuidos y de cómo diferentes máquinas podían comunicarse entre sí surgió en la década de los 90. Hasta ese momento, era suficiente con que las aplicaciones de un mismo ordenador pudieran establecer una comunicación. Los servicios Web son muy prácticos, pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no debe afectar al otro. En la medida que se ha avanzado en términos tecnológicos, se percibe un crecimiento proporcional de la información que se

genera en todos los ámbitos, sea científico, humanístico, económico etc. Estos fenómenos no suceden como acontecimientos aislados, sino que son posibles gracias al conocimiento adquirido y al esfuerzo de muchos, de ponerlo a disposición de quien desee acceder a ellos. Teniendo en cuenta el acceso libre y continuo de la información, surge el planteamiento del problema, el cual tiene dos vertientes, la primera, consiste en que no se dispone de una plataforma web acondicionada para que, de manera fácil y rápida se pueda hacer uso del simulador GALATEA los usuarios y usuarias del Centro de Simulación y Modelado (CESIMO) de la Universidad de los Andes, suelen tener muchas dificultades para ejecutar los modelos de simulación en el simulador GALATEA, especialmente al momento de configurar y activar todas sus funcionalidades. Si bien se posee un servidor académico en CESIMO, es necesario desarrollar mecanismos que posibiliten el uso de las herramientas que éste ofrece, por ello la necesidad de servicios web que permitan que las funcionalidades que posee el simulador estén disponibles tanto a nivel local como a nivel externo (Sistema Distribuido). En cuanto al otro aspecto, consiste en verificar que la integridad de la información estén garantizados, en especial cuando el usuario desee enviar algún dato, o realizar alguna consulta, por lo cual el sistema debe validar a dicho usuario previamente.

1.3 Justificación

Dado que las organizaciones protegen sus redes mediante firewalls -que filtran y bloquean gran parte del tráfico de Internet-, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados. Es importante señalar que los servicios web se pueden utilizar sobre cualquier protocolo, sin embargo, TCP es el más común. La principal motivación para realizar este trabajo, es la de desarrollar un sistema web que sirva como base para la simulación de eventos discretos, esperando que un sistema web amplíe la base de usuarios y usuarias del simulador liberándolos del trabajo de configuración y permitiendo acceder a los modelos de simulación, al igual ponerlos en contacto con diferentes expertos que trabajan en el CESIMO y así validar sus modelos de simulación y compartir conocimientos.

1.4 Alcance

Se desea culminar en este proyecto el diseño de una arquitectura de software que permita desarrollar un sistema web para el uso de GALATEA. Se desarrollará un prototipo de esa arquitectura, implementando el sistema web para la ejecución de los modelos de simulación. El prototipo debe cubrir todo lo referente al modelado de los eventos discretos, para el funcionamiento adecuado del sistema. Adicionalmente, la arquitectura del mismo debe permitirles adaptarse al contexto del usuario, y reaccionar en base a la interacción del usuario con el sistema.

1.5 Objetivos

1.5.1 Objetivo general

Diseñar e implementar un sistema web para los usuarios y usuarias, modelistas y simulistas del simulador de eventos discretos GALATEA, que les permita realizar todas las tareas habituales de modelado, codificación y análisis en sus computadores y en la forma que prefieran, pero permitiéndoles realizar las tareas automáticas de compilación, gestión de archivos, simulación y gestión de salidas, en el espacio virtual y con los recursos compartidos de un servidor Web.

1.5.2 Objetivos específicos

1. Desarrollar un sistema web que permita el control de usuarios junto con los roles a ser utilizados en el sistema.
2. Diseñar e implementar una arquitectura de software que permita la comunicación entre el software de simulación y el sistema web.
3. Instalar y configurar en un servidor la arquitectura de software para el sistema de simulación.
4. Incorporar el simulador GALATEA como servicio para el sistema web.
5. Diseñar y desarrollar un cliente GUI/controlador para un modelo que se pueda gestionar archivos y simular con GALATEA a través del sistema web desarrollado.
6. Sistematizar la experiencia de uso del sistema web para simulación.
7. Analizar el sistema web desarrollado y establecer las conclusiones.

1.6 Metodología

Para el desarrollo del proyecto, se utilizó el método SCRUM de desarrollo de software. El mismo “es una metodología de desarrollo muy simple, que requiere trabajo duro, porque la gestión no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto”. Entre sus características se encuentra que:

1. Es un método de desarrollo de carácter adaptable.
2. Es orientado a las personas, antes que a los procesos.
3. Emplea desarrollo ágil, interactivo e incremental.

1.7 Estructura del documento

Capítulo I. Introducción.

En esa sección se describe brevemente las características del problema abordado en el proyecto, analizando los antecedentes, delimitando los objetivos y estableciendo la metodología con que se procedió al estudio y resolución del mismo.

Capítulo II. Marco teórico.

En esa sección se exponen algunos conceptos relacionados con el problema planteado en el proyecto. Se da una breve introducción sobre los términos y herramientas utilizados para desarrollar el mismo.

Capítulo III. Desarrollo del Sistema WEB.

Contiene el diseño general del sistema de simulación, así como la arquitectura de la aplicación web, sus sub-sistemas, componentes relevantes y casos de uso.

Capítulo IV. Integración de GALATEA.

Explicar aquí.

Capítulo V. Pruebas.

Explicar aquí.

Capítulo VI. Conclusiones.

Explicar aquí.

Capítulo 2

Marco teórico

La simulación por computadora se ha convertido en una parte útil del modelado de muchos sistemas naturales en física, química y biología, y sistemas humanos como la economía y las ciencias sociales (sociología computacional), así como en dirigir para ganar la penetración (profundidad) su comportamiento cambiará cada simulación según el conjunto de parámetros iniciales supuestos por el entorno. En éste capítulo se explican en detalle las herramientas, tecnologías y conceptos usados para la consecución del proyecto que se propone. Definimos la simulación de eventos discretos DEVS (Discrete Event Simulation), así mismo nos introducimos en GALATEA, con el fin de proporcionar una base inicial para la codificación del nuevo sistema. Explicamos las ventajas de un servicio web y las herramientas empleadas para el desarrollo del mismo, como lo serian el Patrón de diseño Modelo Vista Controlador (MVC) y Django.

2.1 Simulación Distribuida

Los sistemas de simulación distribuidos son herramientas muy útiles en las labores de investigación porque permiten reducir el tiempo necesario para ejecutar experimentos. Las simulaciones distribuidas pueden hacerse de forma que se acelere una simulación o que se acelere la ejecución de un conjunto de simulaciones. Los sistemas de simulación basados en agentes se pueden utilizar con éxito en ambos tipos de simulaciones distribuidas (Santana et al., 2004).

La Simulación Distribuida, es la que permite conducir modelos de simulación a través de múltiples hosts o computadoras separados por redes, se ocupa de cuestiones que surgen de la distribución de un programa de simulación de eventos discretos en varias computadoras (Banks et al., 2010). La simulación paralela de eventos discretos se refiere a la ejecución en plataformas multi-procesador que contienen múltiples unidades de procesamiento central que interaccionan entre sí con frecuencia, por ejemplo, miles de veces por segundo.

2.2 Simulación de Eventos Discretos

La simulación de eventos discretos es la "imitación" de un proceso de operación o de un sistema del mundo real construido sobre la base del tiempo (Banks et al., 2010). Esta codifica el comportamiento de sistemas complejos como una secuencia de eventos ordenados y bien definidos. En simulación de eventos discretos (DEVS por sus siglas en ingles), el funcionamiento de un sistema se representa como una secuencia cronológica de los acontecimientos. Cada evento tiene lugar en un instante de tiempo y marca un cambio de estado en el sistema. Frente a su homóloga, la simulación de tiempo continuo, esta se caracteriza por un control en la variable del tiempo que permite avanzar a éste a intervalos variables, en función de la planificación de ocurrencia de tales eventos a un tiempo futuro. Estos sistemas se caracterizan por mantener un estado interno global del sistema, que puede no obstante estar física o lógicamente distribuido, y que cambia parcialmente debido a la ocurrencia de un evento. La ejecución de un evento puede desencadenar la generación de nuevos eventos futuros. Cada uno está marcado por su tiempo, por lo que el orden de generación puede no coincidir con el orden de ejecución.

2.3 GALATEA

Galatea es una plataforma para simulación de eventos discretos, DEVS, con una semántica basada a una red de nodos como metáfora del sistema a simular, la misma semántica matriz del sistema Glider, el proyecto paterno local, que fuera formalizada como parte del nuevo proyecto y luego generalizada para luego re-acomodar la simulación continua, la simulación de sistemas multi-agentes y la simulación distribuida, simulación con autómatas celulares y simulación con modelos explícitos de espacios urbanos o arquitectónicos (Uzcátegui et al., 2011).

Podemos explicar de forma breve y concisa en que consiste la misma, segun (Uzcátegui et al., 2011):

Galatea es una plataforma libre de código abierto para simulación de sistemas multi-agente que incorpora estrategias de simulación bien conocidas con la que cualquier modelista o simulista puede ensayar esas estrategias en problemas de simulación de sistemas complejos. La historia de Galatea comienza mucho antes que se planteara formalmente el proyecto con ese nombre. En 1993, nuestro muy joven Centro de Simulación y Modelos, CeSiMo, propone un proyecto para explorar la re-implementación de la plataforma de simulación Glider sobre una plataforma orientada a objetos dando origen a un prototipo experimental. El problema del cambio estructural, inspirado por investigaciones en economía, se había convertido entonces en uno de los objetivos de investigación fundamentales del CeSiMo y vendría a dictar también la pauta para Galatea. La noción de agente hizo su aparición en algunos reportes internos en los que se enfatizaba su importancia para modelar sistemas complejos como una economía nacional. En 1998 se planteó la posibilidad de integrar Glider con herramientas de inteligencia artificial para modelar agentes. En el 2000, un

proyecto vendría a combinar aquel prototipo de 1993, con una teoría de agentes basada en lógica computacional que se planeaba integrar en una nueva teoría de simulación de sistemas multi-agentes. Allí nació Galatea. El logro fundamental para el proyecto, sin embargo, llegaría con las aplicaciones. En 2004, Galatea fue incorporada al banco de pruebas de un proyecto en biocomplejidad.

2.3.1 La arquitectura de la plataforma Galatea

La arquitectura de GALATEA está basada en objetos. Tanto los agentes como el simulador principal están desarrollados de acuerdo a dicho diseño (orientado a objetos OOD), para apoyar la distribución, la modularidad, la escalabilidad y la interactividad como lo exige la especificación HLA (High Level Architecture). El lenguaje base usado para su desarrollo es el Java, bien especificado en (Uzcátegui et al., 2011). Usando como lenguaje base Java, se especifico un nuevo lenguaje de simulación al que denominan lenguaje Galatea. La sintaxis del lenguaje Galatea es, de hecho una mezcla de la sintaxis Glider, las estructuras básicas de Java y las reglas de conductas de los agentes escritas en los lenguajes de programación lógica Actilog. La compleja semántica de Galatea establece que el código Java sea compilado y ejecutado por el motor de simulación de eventos discretos, mientras que las reglas de los agentes son interpretadas por un motor de inferencia implementado sobre una máquina Prolog (Uzcátegui et al., 2011). Es una plataforma mixta, que corre sobre máquinas virtuales Java y Prolog.

2.4 Servicio Web

Un servicio web (Web Service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La World Wide Web Consortium lo define como “...un sistema de software diseñado para soportar interacción interoperable máquina a máquina sobre una red. Este tiene una interface descrita en un formato procesable por una máquina (específicamente WSDL). Otros sistemas interactúan con el servicios web en una manera prescrita por su descripción usando mensajes SOAP, típicamente enviados usando HTTP con una serialización XML en relación con otros estándares relacionados con la web” (Machuca, 2011). Se puede definir de manera más sencilla como un conjunto de tecnologías estándares de software para el intercambio de datos entre aplicaciones tales como SOAP, WDSL y UDDI. Estos pueden ser desarrollados en una gran variedad de lenguajes para ser implementados sobre muchos tipos de redes de computadores. El éxito de la interoperabilidad se consigue gracias es la adopción de protocolos y estándares abiertos.

2.5 Arquitectura Orientada a Servicios

Una arquitectura orientada a servicios (SOA, Service Oriented Architecture), es una arquitectura de estilo técnico que proporciona los medios para integrar sistemas dispares y descubrir las funciones de negocio reutilizables. Es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos. Las soluciones SOA han sido creadas para satisfacer los objetivos de negocio las cuales incluyen facilidad y flexibilidad de integración con sistemas legados, alineación directa a los procesos de negocio reduciendo costos de implementación, innovación de servicios a clientes y una adaptación ágil ante cambios incluyendo reacción temprana ante la competitividad (Quiroga, 2011).

La definición de SOA por el W3C es: “Conjunto de componentes, los cuales pueden ser invocados y cuyas descripciones de interfaces pueden ser invocadas y descubiertas”.

Gartner: “SOA es una arquitectura de software que comienza con una definición de interfaz y construye toda la topología de la aplicación como una topología de interfaces, implementaciones y llamadas a interfaces. Sería más adecuado llamarla “arquitectura orientada a interfaces”. SOA es una relación de servicios y consumidores de servicios, ambos suficientemente amplios para representar una función de negocios completa”.

SOA no es sólo una arquitectura de servicios visto desde una perspectiva de la tecnología, si no las políticas, prácticas y frameworks con los que se garantiza la forma correcta de que los servicios sean provistos y consumidos. Es importante que si un servicio no va a ser usado por múltiples consumidores, la especificación sea generalizada; los servicios necesitan ser extraídos de la implementación y los desarrolladores de las aplicaciones del consumidor no deberían necesitar saber nada sobre los modelos de bajo nivel y sus reglas (Quiroga, 2011).

2.6 Herramientas usadas para el desarrollo del sistema

A continuación se describen las herramientas usadas para el desarrollo del sistema web de simulación de eventos discretos GALATEA:

2.6.1 Modelo Vista Controlador (MVC)

El modelo vista controlador (MVC) es un patrón de diseño de software, que separa los datos y la lógica de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. Este patrón de diseño es muy popular en el marco de aplicaciones web ya que su abstracción permite escribir

software altamente desacoplado y fácil de mantener y escalar. La figura 2.1 muestra la relación entre los módulos de MVC.

Los modelos son los componentes de la aplicación del sistema que realmente hacen el trabajo. Se mantienen muy distintos de las vistas, que muestran aspectos de los modelos. Los controladores se utilizan para enviar mensajes a los modelos, y proporcionar la interfaz entre el modelo con sus vistas asociadas y los dispositivos de interfaz de usuario interactivas (por ejemplo, teclado, ratón). Cada vista se puede pensar que está estrechamente vinculada con un controlador, que tienen cada uno exactamente un modelo, sino un modelo puede tener muchos pares vista / controlador (Krasner y Pope, 1988).

Modelo

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, así por ejemplo un sistema de administración de datos climatológicos tendrá un modelo que representará la temperatura, la humedad ambiental, el estado del tiempo esperado, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación (Pantoja, 2004).

Vista

Las vistas son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo; así por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital (Pantoja, 2004).

Controlador

El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador (Pantoja, 2004).

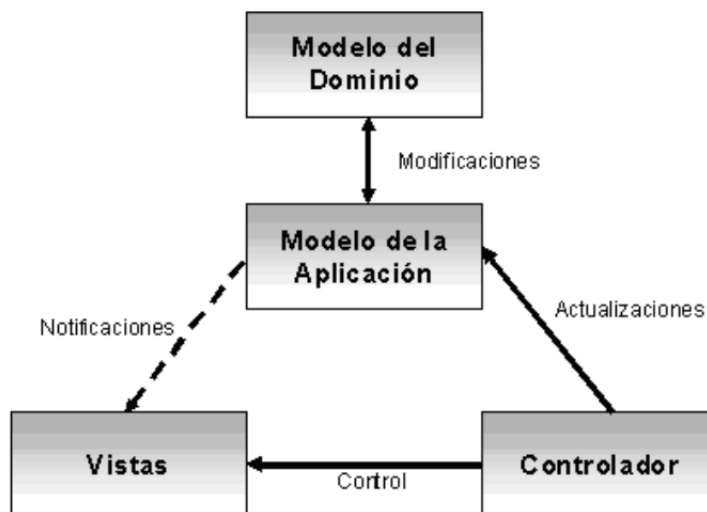


Figura 2.1: Relaciones entre los módulos del patrón MVC.

2.6.2 Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–controlador. La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos. Django permite construir aplicaciones web rápidamente, gracias a su filosofía de baterías incluidas, es decir, que incluye una inmensa gama de características comunes a la mayoría de las aplicaciones web como las validaciones, autenticación de usuarios, manejo de sesiones entre muchos otros (Foundation, 2015).

Cuando Django recibe una petición ésta pasa por un despachador de URLs (URL Dispatcher), cuya tarea es emparejar el URL con una vista y delegar a la vista el manejo de la petición. La vista contiene la lógica necesaria para atender la petición entrante. Generalmente esto consiste en retirar o actualizar algunos datos del modelo, que a su vez se comunica con el manejador de base de datos, finalmente la vista combina los datos retirados de la base de datos, la petición y la sesión activa con una plantilla (template) para generar la respuesta que será devuelta.

Django ha demostrado ser escalable y flexible, se sabe de instancias de Django atendiendo ráfagas de cincuenta mil peticiones por segundo, además es de código abierto, gratuito y cuenta con una extensa comunidad de colaboradores y amplia documentación.

2.6.3 Bases de Datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Según (Oracle, 2015), una base de datos es una colección de datos estructurados. Puede ser cualquier cosa desde una simple lista de compras. una galería de fotos o las bastas cantidades de información en una red corporativa. Para agregar, acceder y procesar los datos almacenados en una base de datos, se necesita un sistema manejador de base de datos. Ya que los computadores hacen un muy buen trabajo manejando grandes cantidades de datos, los sistemas manejadores de bases datos juegan un papel central en la computación como utilidades independientes o partes de otras aplicaciones.

MySQL

Mysql es un sistema manejador de bases de datos relacionales (RDBMS por sus siglas en ingles) de código abierto bajo la licencia GPL (GNU General Public License). Mysql se caracteriza por ser rápido, confiable escalable y fácil de usar, es posible instalar Mysql tanto en una maquina junto a otras aplicaciones como servidores web o también instalarlo en máquinas dedicadas para que use todo el poder de cómputo disponible. Mysql posee características para ejecutarse en clusters de máquinas junto con un motor de replicación para obtener una alta escalabilidad (Oracle, 2015).

PostgreSQL

PostgreSQL es un sistema de base de datos de gran alcance, de código abierto objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de datos y correctitud. Se ejecuta en todos los principales sistemas operativos, incluyendo Linux, UNIX, y Windows. Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, combinaciones, vistas, triggers y procedimientos almacenados (en varios idiomas). Incluye más SQL: 2008 tipos de datos, incluyendo entero, numérico, Boolean, CHAR, VARCHAR, DATE, INTERVALO y TIMESTAMP. También es compatible con el almacenamiento de objetos binarios grandes, como imágenes, sonidos, o de vídeo. Tiene interfaces de programación nativo de C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros, y la documentación excepcional (PostgreSQL, 2016).

Mapeo Objeto-Relacional

El Mapeo Objeto-Relacional (ORM por sus siglas en ingles) es un método para interactuar con bases de datos relacionales desde el paradigma de la programación orientada a objetos, de esta manera es posible aprovechar conceptos como herencia y polimorfismo. Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

El uso de un ORM simplifica enormemente el manejo de la estructura de datos subyacente ya que permite al programador manejar los datos a un mayor nivel de abstracción como si fueran objetos, sin necesidad de general manualmente las consultas SQL, además ésta capa de abstracción permite desacoplar el código de la aplicación de los detalles específicos de cada RDBMS (Pérez, 2016).

2.6.4 JSON

JSON (JavaScript Object Notation) es un formato textual de intercambio y almacenamiento de datos no estructurados, JSON posee un formato que es fácil de leer para humanos y fácil de interpretar para máquinas y más ligero que XML por lo que se ha popularizado para el desarrollo de APIs. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza las convenciones que son familiares para los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen JSON un lenguaje ideal de intercambio de datos.

2.6.5 AJAX

AJAX (Asynchronous JavaScript And XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor (Fuentes, 2009).

Capítulo 3

Desarrollo del Sistema WEB

El sistema de simulación de eventos discretos mediante la web consiste de tres elementos principales: (1) La aplicación Web que funciona como coordinador del sistema, control de usuarios y roles; (2) Administración de archivos que permite llevar el control de acceso, creación y modificación de los archivos dentro del sistema, y (3) El Sistema Integración con GALATEA, que ofrece las funcionalidades de simulación, conexión y control con el motor de simulación.

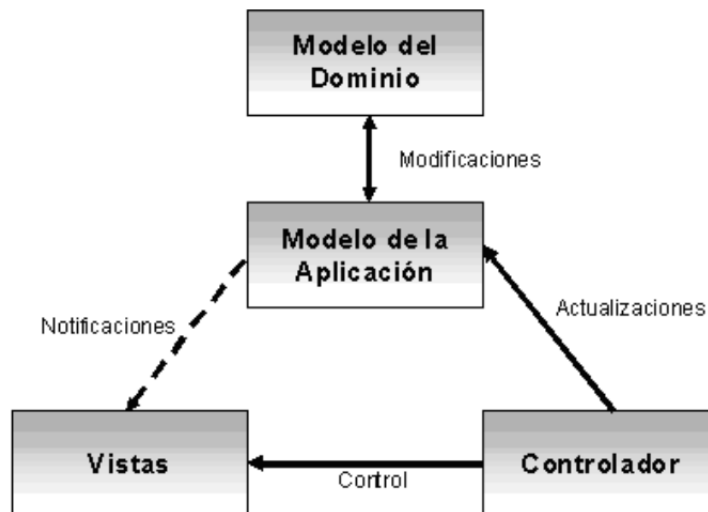


Figura 3.1: Sistema de Simulación Web GALATEA.

En la figura 3.1 se observa la relación que existe entre los componentes principales que conforman el sistema. El punto de entrada principal del sistema es la aplicación Web, que es la encargada de manejar el acceso de los usuarios a las distintas funcionalidades del sistema, esto lo realiza mediante la implementación de roles de usuarios y manejo de permisologías. La aplicación web controla la

visualización, acceso y modificación de los distintos archivos y carpetas, mediante peticiones a el módulo de administración de archivos. Otro aspecto fundamental de la aplicación Web es el control y visualización de las simulación, mediante el acceso al sistema de Integración con GALATEA.

El módulo de Administración de Archivos es una pequeña API que se desarrollo, la cual permite llevar el control de creación, modificación y eliminación de archivos dentro del sistema. Esta API está escrita en python y maneja el control de excepciones u errores que pueden aparecer mediante se realiza ese control, al igual permite verificar el espacio en disco usado por cada usuario. En secciones posteriores se especificara con mas detalles la funcionalidad de este módulo.

El sistema de Integración con GALATEA es una API de interconexión entre nuestra aplicación Web y el simulador GALATEA. Como ambos sistemas están desarrollados en ambientes distintos, con códigos distintos, se busco desarrollar un mecanismo de comunicación. La API Permite crear una instancia del Simulador GALATEA y ejecutarlo, controlar la compilación y los parámetros de ejecución de la simulación. Esto lo desarrollamos en python mediante la implementación de control de procesos y socket. En el capítulo siguiente especificaremos a detalle en funcionamiento de este sistema de integración.

3.1 Diseño de la aplicación WEB

Para el desarrollo de esta aplicación se usó el framework de desarrollo web Django que implementa un patrón de diseño MVC.

3.1.1 Arquitectura

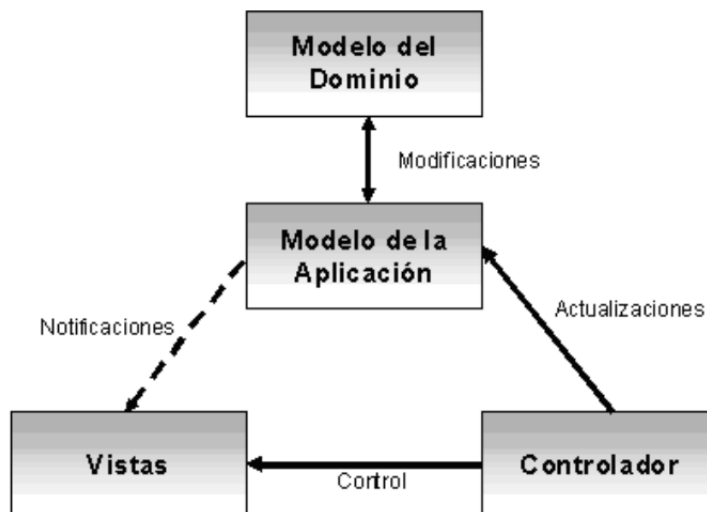


Figura 3.2: Arquitectura de la aplicación web.

La arquitectura del sistema web de simulación GALATEA consiste en cuatro capas, la capa superior o capa de presentación se ejecuta en el navegador del cliente, para ver los resultados obtenidos de la simulación, visualización de archivos, entre otras acciones que puede realizar el usuario; esta capa se comunica con la capa dos o capa de servicio, que es ejecutada por el servidor web y es la responsable de atender las peticiones de los usuarios del sistema; la capa tres es la encargada de las peticiones a los otros módulos que conforman el sistema, comunicación con el módulo de manejo de archivos e integración con el simulador GALATEA; la capa cuatro o capa de datos aloja los datos de la aplicación como usuarios, archivos, carpetas, simulaciones entre otros.

3.1.2 Capa 1: Presentación

La capa de presentación es la interfaz gráfica que permite la interacción del usuario con el sistema de simulación Web. Esta capa consiste en los despliegues de navegadores en cualquier dispositivo conectado a Internet y su propósito es ofrecer la usuario una interfaz gráfica para ingresar el sistema, configurar y manejar simulaciones, observar resultados, entre otros.

La capa de presentación puede ser extendida para la implementación de distintos front-ends como aplicaciones nativas para sistemas operativos móviles o de escritorio a través de APIs programáticas.

3.1.3 Capa 2: Servidor Web

El servidor web es el punto de entrada a la aplicación, éste responde a las peticiones realizadas desde el front-end a través del protocolo HTTP, el enfoque del servidor web es manejar tareas ligeras de la forma mas rápida posible y retornar respuestas para ofrecer al usuario baja latencia en su interacción con el sistema.

Ruta	Acción
/static/	Servicio de archivo estático
/media/	Servicio de archivo estático
/	re-dirección a la aplicación

Tabla 3.1: Tabla de enrutamiento del servidor web.

El servidor web debe manejar dos tipos de peticiones: aquellas que requieren respuestas dinámicas ajustadas a los datos específicos que pertenecen a cada usuario en particular y peticiones de archivos estáticos como CSS, JavaScript, imágenes o archivos HTML pre-definidos que cambien poco o rara vez. A pesar de que la aplicación web es capaz de manejar ambos tipos de peticiones, cada una de ellas tendrá que pasar por todo el pipeline de Django (middlewares, resolución de URLs, procesamiento de la vista, etc), esta complejidad es innecesaria cuando se trata de archivos estáticos y va a sobrecargar el servidor web en entornos de producción.

Para manejar eficientemente todo tipo de peticiones, debemos implementar un mecanismo que sea capaz de servir los archivos estáticos rápidamente y que sirva como proxy entre el front-end y la

aplicación web, hay una gran variedad de servidores como Apache¹, Nginx² o Lighttpd³ diseñados especialmente con este propósito, éstos tendrán una sencilla tabla de enrutamiento que decidirá cómo manejar las peticiones. La aplicación web podrá correr entonces en un proceso separado y sólo atenderá peticiones dinámicas, así se asegurará la eficiencia en el uso de los recursos computacionales disponibles.

3.1.4 Capa 3: Integración de Procesos

Definir

3.1.5 Capa 4: Datos

Incluye todos los sistemas encargados de almacenar o alojar los datos en forma persistente ya sea a través de bases de datos o disco duro.

Bases de datos

La base de datos es el almacén principal de los datos estructurados de la aplicación web, almacena todos tipo de datos de información como datos del usuario, referencias de los archivos, logs de información, entre otras cosas.

El diseño de la base de datos es una de los pilares fundamentales del diseño del sistema, el correcto diseño de las tablas de la base de datos tiene repercusiones importantes en la escalabilidad y tiempo de respuesta del sistema.

Archivos de usuario

Los archivos de usuario son de gran importancia a ser manejados en el sistema, estos pueden ser de configuración de perfiles como imágenes, y los de simulaciones que varían desde archivos Java, archivos GALATEA y archivos de texto con información de simulaciones, entre otros.

Todos estos archivos son almacenamos en un sistema de almacenamiento secundario, el cual es configurado dentro de la aplicación y controlado para su correcto funcionamiento y acceso. Cada usuario cuenta dentro del sistema con una porción de almacenamiento para su uso.

3.1.6 Diseño de la Base de Datos

Puesto que Django hace uso de un ORM para manejar la base de datos, desde le punto de vista de la aplicación web, cada tabla de la base de datos es una clase, por lo que es posible aprovechar todas las características de la programación orientada a objetos como: herencia de clases, clases abstractas, implementación de métodos específicos a un modelo y sobre carga de métodos.

¹Apache <https://www.apache.org/>

²Nginx <https://nginx.org/>

³Lighttpd <https://www.lighttpd.net/>

A continuación se describen los modelos que forman parte de la base de datos:

Cada usuario registrado esta representado por el modelo *user*, el cual contiene la información básica del usuario e información de ingreso al sistema.

El manejo de roles se define en el modelo *group*, en el cual manejamos dos roles principales: el de administrador y el de usuario.

Las relaciones de usuarios se maneja mediante *relation*

Los archivos *file*

Las carpetas *folder*

Los archivos compartidos mediante *sharefile*

Las carpetas compartidas en *sharefolder*

Hay una gran variedad de logs que se manejan los cuales se encuentran en distintas tablas como: *logfile* para logs de archivos, *logsfolder* para las modificaciones de carpetas, entre otros.

Diagrama de entidad-relación

La figura 3.3 muestra las entidades principales y sus relaciones dentro de la base de datos.

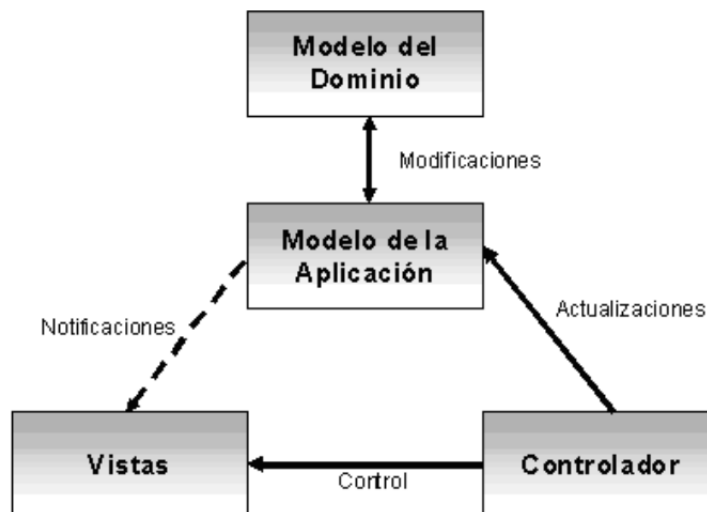


Figura 3.3: Entidades principales de la base de datos.

La figura 3.4 muestra las entidades relacionadas al control y manejo de usuarios, sus roles y perfiles

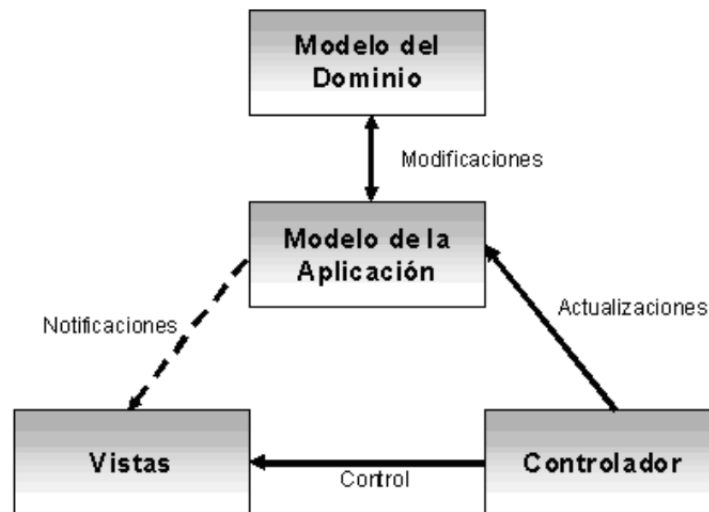


Figura 3.4: Entidades manejo de usuarios.

La figura 3.5 muestra la relación para el control de archivos y carpetas que puede manejar el usuario.

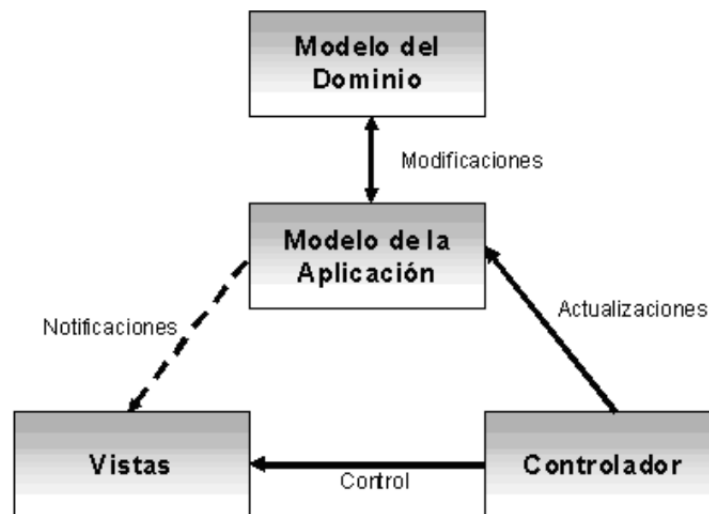


Figura 3.5: Entidades manejo archivos y carpetas.

La figura 3.6 muestra la relación para el control de archivos y carpetas compartidas entre otros usuarios.

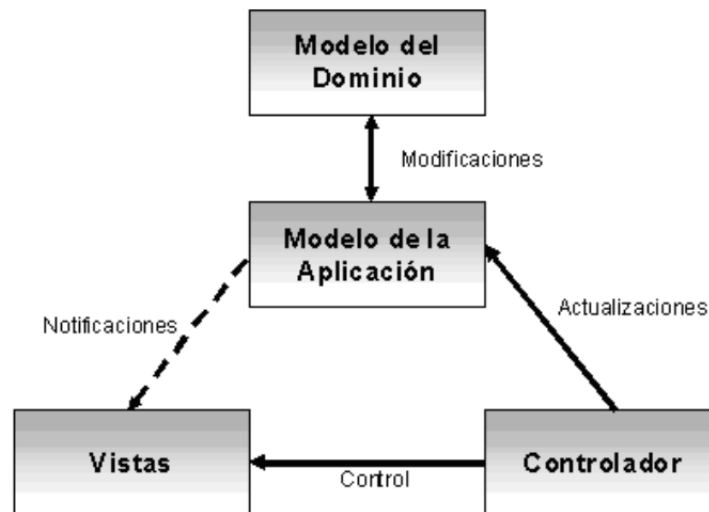


Figura 3.6: Entidades manejo archivos y carpetas compartidos.

Entidades

Las siguientes tablas resumen las entidades y la función de las mismas dentro del sistema web:

Nombre de la entidad	Función
<i>user</i>	explicar
<i>userimage</i>	explicar
<i>group</i>	explicar

Tabla 3.2: Entidades relacionadas al manejo de usuarios.

Nombre de la entidad	Función
<i>user</i>	explicar
<i>userimage</i>	explicar
<i>group</i>	explicar

Tabla 3.3: Entidades relacionadas al manejo de archivos y carpetas.

Nombre de la entidad	Función
<i>user</i>	explicar
<i>userimage</i>	explicar
<i>group</i>	explicar

Tabla 3.4: Entidades relacionadas al manejo de archivos y carpetas compartidos.

Nombre de la entidad	Función
<i>user</i>	explicar
<i>userimage</i>	explicar
<i>group</i>	explicar

Tabla 3.5: Entidades relacionadas al manejo de logs.

Relaciones

- **modelo1 - modelo2 (1-1):** Explicación.
- **modelo1 - modelo2 (1-1):** Explicación.
- **modelo1 - modelo2 (1-1):** Explicación.
- **modelo1 - modelo2 (1-1):** Explicación.
- **modelo1 - modelo2 (1-1):** Explicación.

3.1.7 Diseño de Pantallas

Todas las pantallas del sistema fueron creadas bajo el paradigma del diseño web adaptable, es decir, que pueden ajustarse a cualquier resolución de pantalla sin reducir la usabilidad o sacrificar la experiencia del usuario, el servidor no tiene que decidir entre un conjunto de plantillas para distintas resoluciones, sino que envía al cliente sólo documento HTML y este combina las reglas de presentación de archivos CSS y la lógica de archivos JavaScript para desplegar una página web adaptada al dispositivo del usuario.

Ya que una aplicación web está compuesta por decenas de vistas que comparten componentes como: barras de navegación, cabeceras y pie de páginas, Django incluye un micro-lenguaje de plantillas con funcionalidades de herencia e inclusión, estructuras de repetición y decisión, esto hace que sea posible tener una taxonomía de plantillas, de manera que sólo es necesario escribir los elementos comunes a un conjunto de plantillas una vez y crear nuevas pantallas en el tope de otras. Se le llama rendering al proceso de combinar los datos de la base de datos y la petición del usuario, junto con una plantilla predefinida en la aplicación para generar páginas web personalizadas.

Todas las plantillas heredan de una plantilla base, ésta incluye la declaración del archivo HTML, la cabecera y define bloques que pueden ser sobrescritos por las plantillas "hijas". Los bloques principales definidos en nuestro diseño son: **(1) Bloque de título** permite definir el título de la pagina, **(2) Otro bloque** permite explicación, **(2) Otro bloque** permite explicación.

A continuación se muestran las principales pantallas de la interfaz de usuario del servicio web:

- **Pantalla 1:** Explicación.

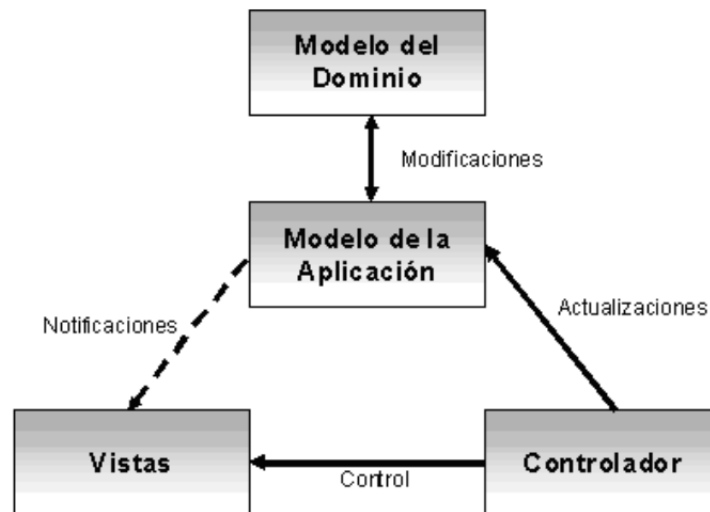


Figura 3.7: Pantalla1.

- **Pantalla 2:** Explicación.

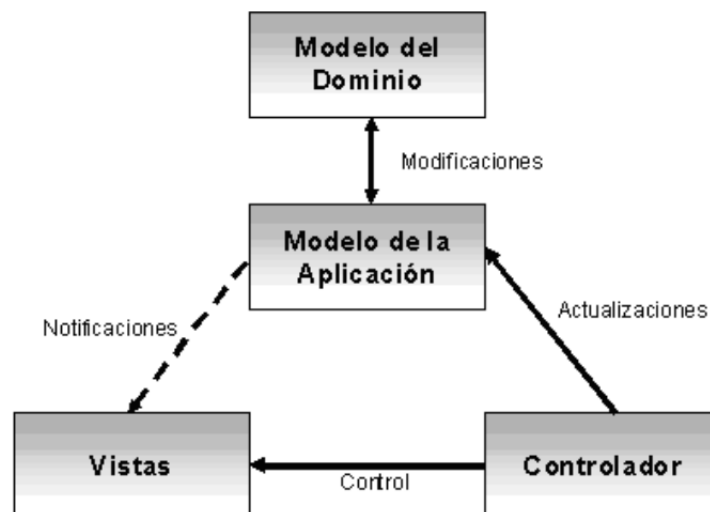


Figura 3.8: Pantalla2.

- **Pantalla 3:** Explicación.

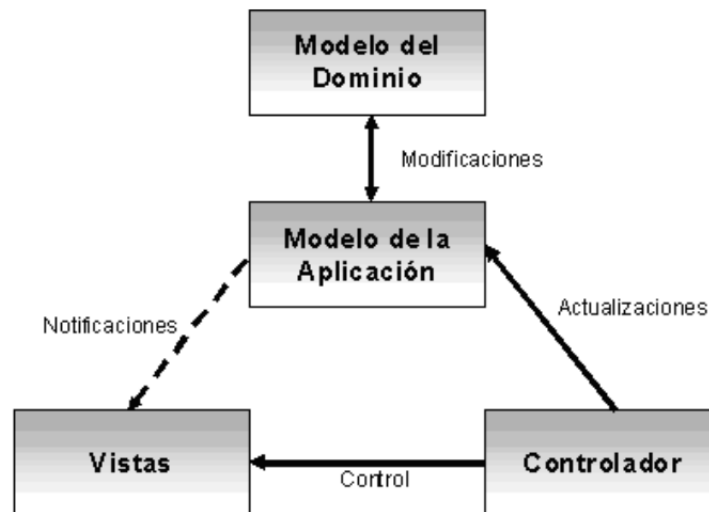


Figura 3.9: Pantalla3.

3.1.8 Diseño de URLs

Ya que la aplicación web consiste en un largo conjunto de vistas, debemos diseñar URLs que apunten a cada una de ellas de forma ordenada y lógica. Para esto, se ha usado una convención en la que la unidad principal de operación como archivos, carpetas, usuarios, a continuación se muestran ejemplos de las reglas usadas para diseñar las URLs:

- Operaciones sobre algo:
 - **GET** /admin/algo/ explicación.
 - **POST** /admin/algo/ explicación.
 - **GET** /user/algo/ explicación.
 - **POST** /user/algo/ explicación.
- Operaciones sobre algo:
 - **GET** /admin/algo/ explicación.
 - **POST** /admin/algo/ explicación.
 - **GET** /user/algo/ explicación.
 - **POST** /user/algo/ explicación.
- Operaciones sobre algo:
 - **GET** /admin/algo/ explicación.
 - **POST** /admin/algo/ explicación.

- **GET** /user/algo/ explicación.
- **POST** /user/algo/ explicación.

3.2 Gestión de Archivos

Definir

3.2.1 Configuración

Definir

3.2.2 Control de Archivos

Definir

3.2.3 Control de Carpetas

Definir

3.2.4 Control de Permisos

Definir

Capítulo 4

Integración de GALATEA

Integración de GALATEA aquí.

Capítulo 5

Pruebas

Los análisis realizados en distintas fuentes de información sobre los departamentos, ayudan a delimitar y especificar los datos y variables que influyen en sus necesidades, y a comprender el funcionamiento y el papel que juega cada uno en la situación docente y estudiantil. Anteriormente en la Universidad de Los Andes se han propuesto algunas herramientas que han sido de utilidad en el proceso de asignación de cargos, como son los baremos elaborados en la Facultad de Arquitectura y Diseño, y en la Escuela de Ingeniería Eléctrica.

5.1 Criterios y variables

La revisión y análisis de los informes de actividades cumplidas, cuya función es recoger información sobre las actividades de los profesores para evaluar su rendimiento, se detalla en el apéndice ???. Este estudio preliminar también contribuye en la construcción de variables de interés, por medio de un análisis interpretativo de los valores plasmados.

5.1.1 Estudio preliminar de criterios y variables

El baremo de la Escuela de Ingeniería Eléctrica fue desarrollado como herramienta para argumentar la solicitud de nuevos cargos docentes en la unidad académica (Viloria, 2015). Este baremo está basado en el desarrollado en la Facultad de Arquitectura y Diseño (Rodríguez, 2010). En la versión de la Escuela de Ingeniería Eléctrica se observan cálculos sobre la deficiencia de profesores para cada materia, el cual es extendido luego para cada departamento y para la escuela.

La necesidad de cargos de una escuela dependerá de la disponibilidad horaria de cada profesor. La disponibilidad horaria influye a su vez en la cantidad de secciones que se pueden ofertar de cada materia. La cantidad de secciones influye en el número de alumnos que tendrá cada sección. El número de alumnos inscritos es una variable externa dada por datos del semestre inmediato anterior o actual.

Los siguientes valores influyen en la disponibilidad horaria de un profesor: horas de carga asignada, que va ligada a la dedicación del profesor, que puede ser dedicación exclusiva (DE), tiempo completo (TC), medio tiempo (MT) o tiempo convencional (Tconv); horas dedicadas a las actividades de administración; profesores en estado de beca, plan de formación, año sabático, permiso remunerado o no remunerado, entre otros.

Los límites sugeridos en cantidad de alumnos por sección están establecidos en normas o reglamentos (??), los cuales dependen de la modalidad del curso. En la revisión al baremo construido para la Escuela de Eléctrica, se encontraron los siguientes puntos:

- Los cálculos se hacen en base a las materias de cada área de conocimiento.
- Cada materia puede ser dictada por un conjunto de profesores pertenecientes al área correspondiente.
- El cálculo de las secciones de la materia que pueden ser atendidas se basa en la capacidad horaria de los profesores a *DE* y *TC*. Se utiliza como horas disponibles para clases un límite de 30.
- El número de alumnos que debe tener cada sección, según sugerencias por modalidad del curso, varía entre 15 y 30.
- El número de alumnos que deben inscribirse se toma como el número de alumnos inscritos para cada materia en OREFI, es decir, la cantidad real del semestre actual o más cercano.
- El número de profesores requeridos para la asignatura se basa en un cálculo que utiliza factores ligados al tipo de materia (especial o estándar) y factores ligados a la dedicación del profesor.
- Se calcula el déficit de profesores como la diferencia de los profesores requeridos y los disponibles. Luego se anexa también el déficit de profesores debido al porcentaje reglamentario de profesores que puede estar de sabático, beca, permiso, etc. En este caso se ha usado el 20 % del total de profesores. Según el ?, art. 11, art. 24, es un 10 % de profesores para becas, un 10 % para años sabáticos, sólo en caso de que sean más de 10 profesores.
- Se anexa también el déficit por carga administrativa, para lo cual se toma en cuenta solo aquellos cargos que tengan más de 10 horas administrativas.
- Se anexa las horas sobrantes en la carga académica.
- Finalmente se combinan cada déficit para obtener el total por área académica y facultad.
- No se especifica realmente la cantidad de profesores en beca, sabático, etc. Sólo se asume el 20 % de déficit total.
- No se especifica la cantidad de profesores jubilables o jubilados activos.
- No se toman en cuenta explícitamente las áreas de investigación y extensión.

Por otro lado, se encuentran algunas restricciones en los informes de actividades que impiden realizar cálculos similares a los del baremo de la Escuela de Ingeniería Eléctrica:

- En la planilla debe estar recogido el número de estudiantes en cada sección para poder aplicar correctamente los factores de ponderación de horas teóricas. En caso de que sea usado debidamente, no se debe incluir nuevamente el estudio de la cantidad deseable de alumnos por sección y la cantidad real inscrita de alumnos por sección, ya que estaría incluido implícitamente los factores utilizados.
- En el baremo de la Escuela de Ingeniería Eléctrica, las horas que dedica un profesor a *DE* o *TC* a clases deben ser 30. Según esto, el Porcentaje de Actividades Docentes (*PAD*) calculado en las planillas de informes de actividades cumplidas debe reflejar como máximo, 75 % para *DE* y 85,7 % para *TC*. El resto debería ser repartido entre las otras actividades. Otro tema a recordar en este punto es el llenado de los informes sólo por personal a *DE*, como lo establece el ?, art.58, num 2.a, o discutir si este debe extenderse a todo el personal docente, como parecer suceder actualmente para el caso del personal a *TC*, *MT* y *Tconv*.

En resumen, los valores plasmados en los informes de actividades miden el desempeño del departamento según las horas dedicadas a las actividades. Por la naturaleza comprimida de los valores, se hace difícil la interpretación de los mismos para un análisis más detallado en cuanto a número de estudiantes, límites de alumnos y secciones, y estado de los profesores. Por estas razones se considera seleccionar algunas variables de interés basadas en las observaciones de los baremos de la Escuela de Ingeniería Eléctrica y la Facultad de Arquitectura y Diseño, además de observaciones realizadas en informes proporcionados por los departamentos en cuanto a situación docente y estudiantil (ver apéndice ??).

Cada departamento de la Escuela de Sistemas recoge datos de importancia para el proyecto, relacionados a la demanda estudiantil. Entre los datos que recogen se tiene información sobre la situación estudiantil y docente junto con los indicadores: número de estudiantes por profesor, número promedio de secciones por profesor, número de secciones atendidas por profesores jubilados y jubilables, porcentaje de profesores jubilables, entre otros. En la tabla 5.1 se muestra el resumen de estos datos por cada departamento para el año 2014.

D_3	44,1	1,5	8	25	20,1	20	99,5
-------	------	-----	---	----	------	----	------

Tabla 5.1: Datos recogidos por los departamentos de la EISULA. Año 2014.

Si se observan detenidamente los valores en la tabla 5.1, se puede inferir que el departamento con más necesidad es el “ D_3 ”, debido a que cuenta con valores mayores para los factores considerados como indicadores: mayor cantidad de estudiantes y secciones por profesor, mayor presencia de profesores jubilados y jubilables, y mayor número de secciones requeridas. Probablemente debido a esto no

alcanza su capacidad para cubrir el 100 % de secciones requeridas.

El límite de alumnos por sección utilizado es de 30, sin hacer distinción entre materias teóricas, laboratorios o prácticas. Cada departamento establece un “número de materias que deberían dictarse” y se compara con las que se han podido ofertar. La tabla 5.2 muestra los valores resumidos. En la fila “Promedios” se calcula la media de los valores para cada columna. Luego para cada departamento, se calcula la diferencia entre su valor y la media obtenido, de igual forma para todos los factores especificados en la tabla 5.1. Por ejemplo, para la columna “Estudiantes por profesor”, se obtiene el promedio como:

$$\text{Promedio} = \frac{20,6 + 40,2 + 44,1}{3} = 35,0$$

Mientras que las diferencias para cada departamento se calculan como:

$$D_1 = 20,6 - 35,0 = -14,6$$

$$D_2 = 40,2 - 35,0 = 5,2$$

$$D_3 = 44,1 - 35,0 = 9,1$$

Esto refleja la distancia que existe entre los valores medios y los valores correspondientes a cada uno, mostrando los excesos o faltas que tengan los departamentos, para cada factor.

Los totales de la última columna son calculados como la suma de las diferencias para cada factor. De esta manera se construye un valor que resume las diferencias entre las medias y los factores de cada departamento. Los valores altos positivos reflejan la posición del departamento por encima de la media total, mientras que los negativos indican que el departamento ocupa una posición por debajo del promedio, considerando así a los departamentos con mayor valor positivo como los de mayor necesidad.

Tabla 5.2: Comparación de diferencias basada en datos de la EISULA. Año 2014.

Al analizar esta tabla, según los valores reflejados se puede llegar a la conclusión de que el departamento con mayor necesidad de nuevos cargos es “ D_3 ”. Suponiendo que se compara cada valor con el promedio entre los 3 departamentos, el resultado indica que el departamento de “ D_3 ” (con la suma más alta) refleja mayor necesidad, como se había sugerido anteriormente.

Análisis de conceptos para la estructura del modelo

Se puede pensar en las áreas de investigación, extensión, mejoramiento, administración como áreas de apoyo y servicio a la docencia. De esta manera, cabe preguntarse cómo influye cada área en la disponibilidad de los profesores para cubrir las actividades de docencia urgentes y hasta qué límite se le debe dar importancia a cada área. Indudablemente la urgencia de atención de las actividades de docencia depende de la cantidad de alumnos que demandan cupos en las materias, combinado con la restricción de cantidad deseable de alumnos, que generalmente por el déficit de profesores disponibles termina siendo relegada, lo que puede afectar el rendimiento de los estudiantes. La figura 5.1 muestra algunos factores de interés sobre la necesidad del departamento.

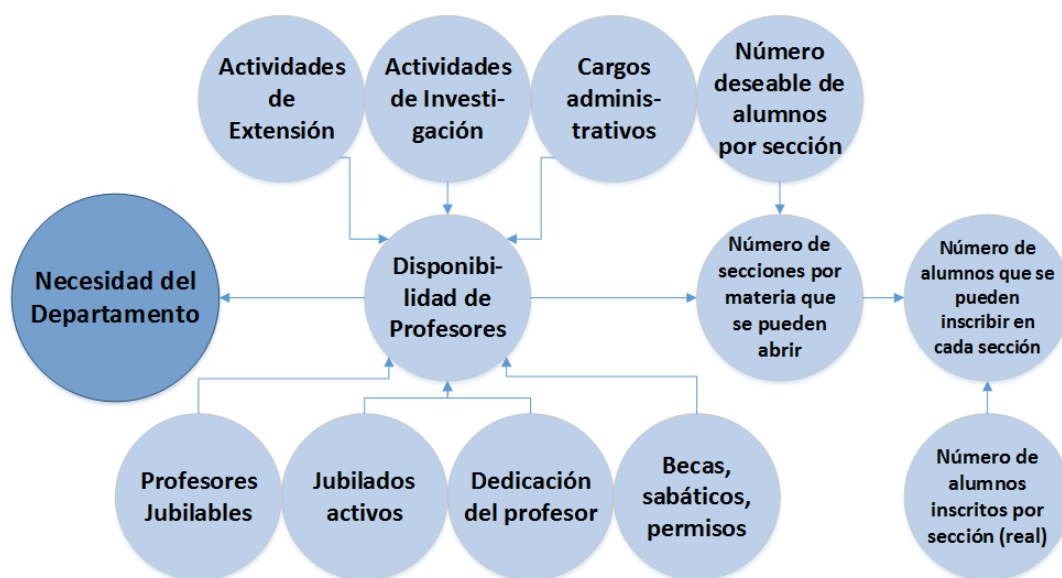


Figura 5.1: Factores de interés sobre la necesidad de un departamento.

¿Qué papel jugarían los valores recogidos en los informes de actividades? Si se utiliza un esquema similar al baremo de eléctrica, los valores PAD , $POAD$, PAI , PAM , $PAEX$, PAA , no permitirían estudiar individualmente variables como el número de alumnos por sección, debido a que está implícita en el valor PAD , o $POAD$. De la misma manera, el número de cargos administrativos estaría implícito en PAA , y el número de profesores en beca, sabático, etc., estaría implícito en PAM . Por otro lado, la interpretación de estos valores y sus promedios, por su naturaleza resumida, se hace difícil. Determinar los límites aceptables de estos valores puede presentar contradicciones. Por ejemplo, si un profesor (o departamento) tiene un rendimiento PAI de 150 %, ¿es malo o bueno? ¿En qué medida se incurre en exceso? ¿En qué medida se considera insuficiente? Que un departamento tenga un PR total por encima de 150 %, puede indicar que los profesores están saturados de actividades, por lo que tendrían urgencia de nuevos cargos, o puede indicar que incurren en excesos en el resto de las áreas, no necesariamente en docencia.

Estudiar y analizar cada porcentaje sin desligarse de los otros podría ayudar a entender cómo se distribuyen las horas del departamento en las actividades, y tomar decisiones sobre los ajustes a realizar. ¿Implica esto que no se debería incentivar la investigación, extensión, y mejoramiento del personal? Dependiendo del enfoque que se tenga respecto a las prioridades de la facultad o escuela, la importancia de cada área debe ser establecida. Un enfoque equitativo para cada área quizás no sea conveniente, teniendo en cuenta el esfuerzo que debe hacerse siempre en el área de docencia por la creciente demanda de los estudiantes.

Dado que se pretenden incluir como base de comparación para la asignación de cargos las distintas áreas que se llevan a cabo en la Facultad, se consideran los valores plasmados en los informes de actividades cumplidas de la Facultad de Ingeniería, además del análisis realizado sobre los baremos mencionados, como preliminares para elaborar una estructura que responda al objetivo del proyecto. De modo que se estudiaron los valores de *PAD*, *POAD*, *PAI*, *PAA*, *PAM*, *PAEX*, además de considerar algunas otras variables de interés y distintas formas de agrupación de variables, con el fin de realizar pruebas sobre la utilidad, influencia y relevancia de cada valor dentro de la toma de decisiones, y su adaptación en modelos de lógica difusa. Para tomar en cuenta las distintas áreas de actividades se especifican variables que consideren la Docencia, Investigación, Extensión, Administración y Mejoramiento. Algunas ideas sobre la agrupación de las variables se muestran a continuación:

- Opción 1: *DIEO*

- Docencia: se propone combinar en una sola variable los factores: *PAD*, *POAD* (tomadas de los informes de actividades), proporción de profesores jubilables, promedio de alumnos por profesor, promedio de secciones por profesor, entre otras.
- Investigación: se propone tomar *PAI*.
- Extensión: se propone tomar *PAEX*.
- Otras: se propone combinar en una sola variable los factores *PAA* y *PAM*.

Desventajas: Las variables *PAD* y *POAD* incluyen implícitamente información sobre alumnos por profesor, secciones, etc, por lo que incluir estos factores podría resultar en redundancia. No hay seguridad de que sea conveniente combinar los factores *PAA* y *PAM* debido a la naturaleza distinta de las áreas.

- Opción 2: *DIEMA*

- Las áreas de Docencia, Investigación y Extensión se toman de la misma manera que la opción 1.
- Las variables sobre Administración y Mejoramiento se toman separadas.

Este caso presenta dificultades similares a la opción 1, pero toma las actividades *PAA* y *PAM* como individuales.

- Opción 3: *DIEMA* más otras.
 - Docencia: se toma la combinación de *PAD* y *POAD*.
 - Investigación: se toma *PAI*.
 - Extensión: se toma *PAEX*.
 - Mejoramiento: se toma *PAM*.
 - Administración: se toma *PAA*.

Luego podría incluirse el resto de variables y se evalúa con el sistema difuso: Proporción de profesores jubilables, promedio de alumnos por profesor, promedio de secciones por profesor, proporción de profesores en beca-sabático, proporción de profesores con cargos administrativos.

Desventajas: Los valores resumidos del rendimiento en cada área no permiten analizar de forma más detallada la situación estudiantil y docente del departamento, como el déficit de profesores, promedios, etc.

Una estructura posible se puede desarrollar con la opción 1 de la manera que se muestra en la figura 5.2, junto con el flujo del proceso para modelos de lógica difusa. La entrada al sistema difuso estaría compuesta por indicadores contruidos a partir de las variables anteriores, por medio de pesos de importancia para cada área. En esta estructura, *PAP* es el promedio de alumnos por profesor, *PS* es el promedio de secciones por profesor y *PJ* es la proporción de profesores jubilables.

Con el fin de manejar los distintos porcentajes de rendimiento en cada actividad, los pesos que se le asignarían a cada área podrían ser de ayuda en este caso. Las áreas que se consideren de mayor importancia en el momento de la asignación de cargos, tendrán un peso acorde con los requerimientos. En este diagrama se observa que se construyen indicadores para cada área luego del cálculo de cada variable. Se obtiene la prioridad para cada departamento y se comparan durante ciclos para realizar las asignaciones. Durante el ciclo se actualizan los datos de departamento (se modifican los datos de informes de actividades y se repite el ciclo hasta que se acaben los cargos). Este enfoque puede resultar en un procesamiento lento y necesita obligatoriamente la revisión de los informes de actividades actualizados en cada ejecución.

Sobre indicadores

En el siguiente apartado se muestran algunos análisis y pruebas sobre el uso de pesos dentro de la construcción de indicadores de interés para el sistema difuso.

Se asume que las variables tomadas se referirán a cada área y a cada una se le asocia un α_i tal que:
 α_i es el grado de importancia o peso para cada área.

X_i es el porcentaje de cada actividad cubierta por el departamento.

Por lo tanto, $\frac{X_i}{\alpha_i}$ indica la medida en que un departamento cubre los requerimientos del momento para la asignación. A continuación se muestra un ejemplo con la aplicación de indicadores. La tabla 5.3, recoge los porcentajes de actividades y de rendimiento de un departamento

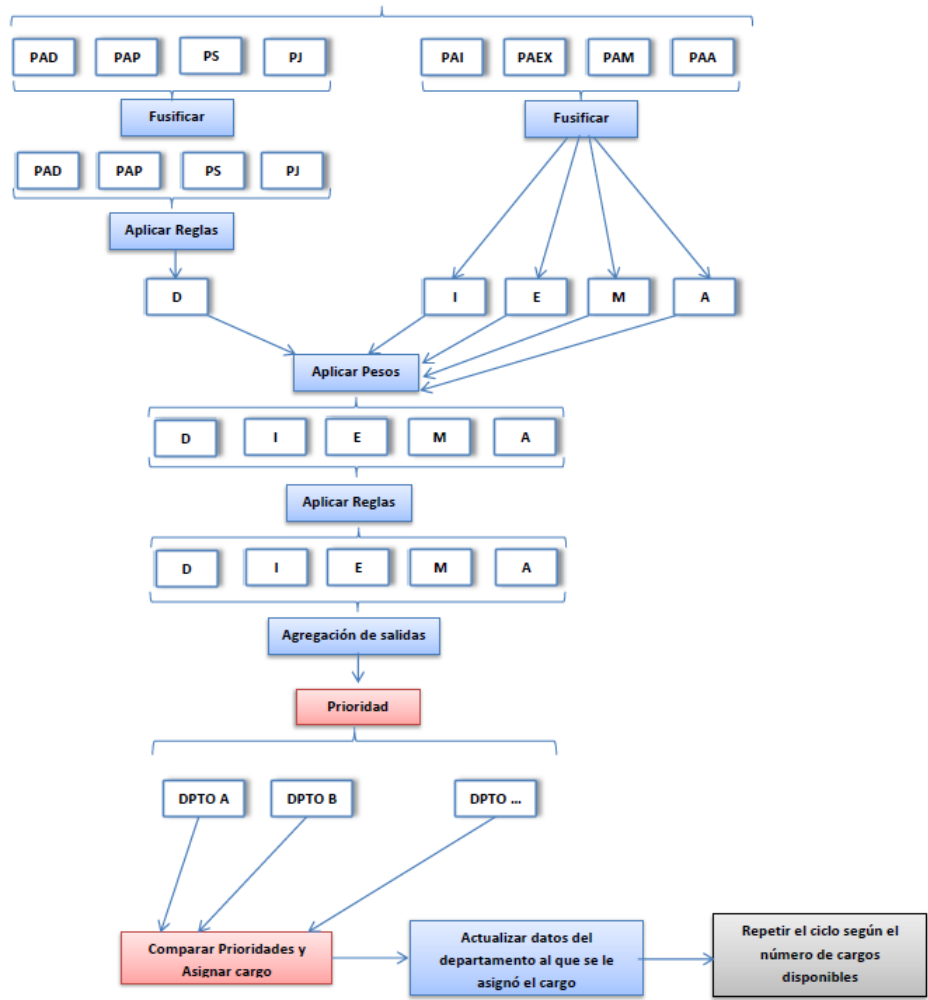


Figura 5.2: Una estructura de procesamiento preliminar.

PAD_x	PAI_x	$PAEX_x$	PAM_x	PAA_x
63,43	34,63	1,40	11,34	22,14

Tabla 5.3: Datos de entrada. Porcentaje de actividades del departamento “x”.

α_1	α_2	α_3	α_4	α_5
45 %	20 %	5 %	20 %	5 %

Tabla 5.4: Pesos de cada área.

IAD_x	IAI_x	$IAEX_x$	IAM_x	IAA_x
1,41	1,73	0,28	0,57	4,48

Tabla 5.5: Indicadores resultantes.

- IAD: Indicador de Actividades de Docencia.
- IAI: Indicador de Actividades de Investigación.
- IAEX: Indicador de Actividades de Extensión.
- IAM: Indicador de Actividades de Mejoramiento.
- IAA: Indicador de Actividades de Administración.

En este caso, para las actividades de Administración se estableció una importancia o peso de 5 % sobre todas las actividades. IAA indica que el porcentaje de actividad está por encima de lo que se requeriría. Estos indicadores pueden ser usados para construir conjuntos que puedan utilizarse en un modelo de Lógica Difusa. Se pueden construir rangos o conjuntos para etiquetar cada indicador, como por ejemplo: Bajo, Regular, Aceptable, Alto, como se muestra en la figura 5.3.

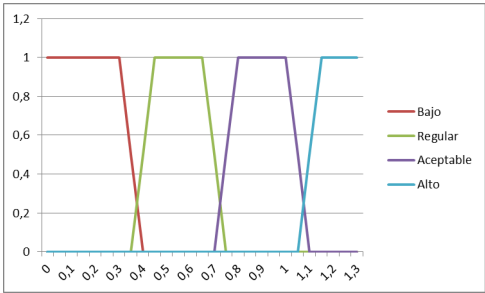


Figura 5.3: Funciones de pertenencia para los indicadores.

En la tabla 5.6 se muestra un ejemplo de clasificación para las combinaciones de conjuntos de los indicadores anteriores. Por ejemplo:

- Si hay 5 actividades en Bajo entonces Prioridad es Baja.
- Si hay 5 actividades en Alto entonces Prioridad es Alta.
- Si hay 2 en Bajo y 3 en Aceptable entonces Prioridad es Normal.

BAJO	REGULAR	ACEPT	ALTO	PRIORIDAD	BAJO	REGULAR	ACEPT	ALTO	PRIORIDAD
5	0	0	0	BAJA	2	2	1	0	MEDIA
0	5	0	0	MEDIA	2	2	0	1	NORMAL
0	0	5	0	NORMAL	2	1	2	0	NORMAL
0	0	0	5	ALTA	2	1	0	2	NORMAL
4	1	0	0	BAJA	1	2	2	0	NORMAL
4	0	1	0	BAJA	1	2	0	2	ALTA
4	0	0	1	BAJA	3	1	1	0	BAJA
1	4	0	0	MEDIA	3	1	0	1	BAJA
1	0	4	0	NORMAL	1	3	1	0	MEDIA
1	0	0	4	ALTA	1	3	0	1	MEDIA
2	3	0	0	MEDIA	1	1	3	0	ALTA
2	0	3	0	MEDIA	1	1	0	3	ALTA
2	0	0	3	NORMAL	2	1	1	1	MEDIA
3	2	0	0	BAJA	1	2	1	1	MEDIA
3	0	2	0	BAJA	1	1	2	1	NORMAL
3	0	0	2	BAJA	1	1	1	2	NORMAL

Tabla 5.6: Algunas reglas sobre los indicadores.

Estos estudios son tomados como base para la comprensión del manejo de las variables en dirección

a la construcción de una estructura eficiente para la asignación de cargos. De este modo se continúa con la búsqueda de esta estructura, como lo muestra la siguiente sección, con la especificación de datos y el uso de los mismos dentro de la construcción de variables, así como formas de asignación de cargos.

Definición y manejo de variables. Otro enfoque

En esta sección continúa el estudio sobre las distintas configuraciones que pueden tomarse en cuenta para el procesamiento de datos, tratamiento de variables y ejecución del sistema difuso.

- Área de docencia: Como se ha mencionado anteriormente, la necesidad de un departamento en el área de Docencia puede ser determinada en función de variables como: proporción de profesores jubilables, proporción de profesores jubilados activos, promedio de alumnos por sección, promedio de alumnos por profesor, porcentaje de rendimiento en docencia, entre otras que sean consideradas influyentes para los asuntos docentes. De acuerdo a esto, se proponen las siguientes definiciones para variables que puedan ser tomadas para el sistema difuso.

Descripción de variables:

- *PAD*: es el porcentaje de actividades de docencia (registrado en los informes de actividades cumplidas).
- Proporción de profesores jubilables (*PJ*): Proporción de profesores del departamento que han cumplido con la antigüedad para jubilarse. La presencia de profesores jubilables en un departamento puede afectar la capacidad para cubrir la demanda de secciones y alumnos por atender. Si la proporción de profesores jubilables es *alta*, el departamento podría tener una necesidad de cargos importante para períodos próximos, principalmente para cubrir la docencia. Se debe considerar un aumento en la prioridad del departamento (a través de un factor de ajuste). La definición de la proporción como *alta* debe ser establecida de acuerdo a los datos observados. Si la proporción de profesores jubilables es *baja*, la prioridad del departamento no debe modificarse.

$$PJ = \frac{\text{Número de profesores jubilables}}{\text{Número total de profesores del departamento}}$$

Ejemplo: Un departamento tiene 10 profesores en total, y entre ellos hay 3 profesores jubilables, por lo tanto $PJ = \frac{3}{10} = 0,33$. Este valor puede servir como factor para modificar el valor de la necesidad del departamento, como se verá más adelante.

- *PJA*: Proporción de profesores Jubilados activos. La presencia de profesores jubilados activos representa un riesgo latente de disminución de la capacidad del departamento. Mientras aumente esta proporción, aumenta la necesidad de cargos del departamento.

De forma análoga a la definida para la proporción de profesores jubilables (PJ), se calcula el factor PJA como sigue:

$$PJA = \frac{\text{Número de profesores jubilados activos}}{\text{Número total de profesores del departamento}}$$

La medida de la necesidad de un departamento en el área de Docencia puede ser definida como D (Docencia), de la siguiente manera:

$$D = PAD * (1 + PJ + PJA)$$

De esta manera, PAD aumenta de acuerdo a la proporción de los factores PJ y PJA . Esto significa que mientras más riesgo de perder cargos activos tenga el departamento, mayor será el valor de D . Este valor será utilizado en el sistema difuso, el cual evalúa las prioridades del departamento en cada área. Si el indicador de necesidad en cada área es categorizado como muy *alto* (mucho más del 100 % requerido), se asume que el departamento tiene una sobrecarga de actividades en el área, por lo que se concluye que asignándole un nuevo cargo, disminuirían las cargas excesivas para los profesores.

Ejemplo: Un departamento con 50 % de actividades docentes cumplidas (de la carga total), $PJ = 0,3$ y $PJA = 0,02$, tendría un valor $D = 50 * (1 + 0,3 + 0,2) = 75$. Si las proporciones PJ y PJA son 0, entonces se toma como indicador el PAD de 50 %.

Otra variable que se puede tomar en consideración para modificar D , es el promedio de alumnos por sección, o el promedio de alumnos por profesor. En base a una cantidad de alumnos por sección recomendada, se puede determinar si un departamento tiene sobrecarga en ese sentido. El enfoque mostrado sobre esta variable, y la manera como se aplican los factores sobre el rendimiento de docencia, debe ser estudiado y analizado a fondo para determinar si es conveniente este tratamiento.

- Promedio de alumnos por profesor, alumnos por sección (PAP , PAS):

El cálculo de promedios de alumnos por sección o alumnos por profesor implica la consideración de algunos detalles tales como la cantidad de secciones que atiende un profesor y la cantidad de alumnos que están inscritos en varias secciones (asignaturas distintas). A partir de este valor y con base en la cantidad de alumnos recomendados para cada sección, denotado por K , se puede calcular la sobrecarga de alumnos por sección del departamento. Ejemplo: Si el promedio de alumnos por sección (PAS) es 25,72 y la cantidad recomendada es $K = 20$, entonces $\frac{PAS}{K} = 1,28$. Este valor puede ser tomado como un indicador a considerar para el sistema difuso. Suponiendo que entre 0 y 1,5 se considera un valor

dentro de lo normal, y si es mayor que 1,5 el valor es excesivo, entonces el indicador de necesidad de la Docencia puede ser modificado como sigue:

$$\begin{cases} D = PAD * (1 + PJ + PJA + PAS/K) & PAS/K > 1,5 \\ D = PAD * (1 + PJ + PJA) & 0 < PAS/K < 1,5 \end{cases}$$

Es decir, en el caso en que el promedio de alumnos por sección sobrepase en más de 50 % a la cantidad recomendada, se aumenta la necesidad del departamento en el área de docencia en proporción al exceso de alumnos por sección. En caso contrario, la necesidad del departamento permanece igual.

- Para las áreas de Investigación y extensión también se toman, como en opciones anteriores, los valores de rendimiento de los informes de actividades, es decir:

$$I = PAI.$$

$$E = PAEX.$$

Uso de los indicadores en el caso actual

Para determinar la necesidad del departamento en cuanto a la asignación de cargos, se consideran las opiniones de los profesores sobre la importancia de cada área de actividades al momento de la asignación. Esta importancia es interpretada como un peso para cada área, el cuál será usado para construir los indicadores que permitan estudiar la necesidad de cada departamento. Suponiendo que los pesos para cada área son α_D , α_I , y α_E s, los indicadores, de manera similar a enfoques anteriores, son:

$$ID = \frac{D}{\alpha_D} \quad II = \frac{I}{\alpha_I} \quad IE = \frac{E}{\alpha_E}$$

Estos indicadores serían utilizados como entradas al sistema difuso que determine las prioridades de cada departamento, para apoyar la toma de decisiones sobre la asignación de cargos.

Algunas consideraciones adicionales

A continuación se exponen algunas consideraciones sobre los conceptos y variables tratados para el proyecto.

- Las variables para las áreas de Administración y Mejoramiento pueden ser tomadas de la misma manera que Investigación y Extensión, o si se requiere anexar alguna otra variable que influya en una de ellas, proceder como en el caso de la Docencia.
- En el caso de las actividades administrativas: En el informe de actividades cumplidas se recogen las horas dedicadas a estas actividades. Como variable alternativa se pudiese usar la proporción

de profesores con cargos administrativos. En el baremo de la Escuela de Eléctrica, se toman en cuenta los cargos administrativos con más de 10 horas (Decanato, jefes de departamento, jefes de laboratorios, entre otros), y estos se anexan al cálculo del déficit de profesores disponibles (para la docencia).

- En el caso de las actividades de mejoramiento: En el informe de actividades cumplidas se recogen las horas dedicadas a estas actividades. Como variable alternativa se pudiese usar la proporción de profesores en becas, sabáticos y demás actividades de mejoramiento que impliquen la ausencia del profesor en las aulas de clases. En el baremo de la Escuela de Eléctrica, se toma en cuenta el número de profesores que pueden tener este estado y se anexan al déficit de profesores disponibles (para la docencia).

Rangos, conjuntos y funciones de pertenencia

Cada conjunto difuso constaría de 2 o 3 funciones de pertenencia, dependiendo de qué tan específica se quiere la clasificación, reduciendo la complejidad al manejar el número de reglas resultantes. Por ejemplo: para el indicador de Docencia, se podrían definir 2 funciones de pertenencia: Bajo, Alto, o 3 funciones de pertenencia: Bajo, Normal, Alto. Para el conjunto de salida (que indique la Prioridad o nivel de Necesidad del departamento), se construirían funciones similares: Baja, Alta o Baja, Normal, Alta.

Asignación de cargos

Suponiendo que las salidas del sistema difuso (prioridad) para cada departamento son: $P_1, P_2, P_2, \dots, P_n$, donde n es el número de departamentos en estudio, la cantidad de cargos a asignar se puede determinar como:

- Opción 1: Asignar un cargo al departamento con mayor prioridad. Repetir el ciclo de cálculos con valores actualizados hasta que se acaben los cargos.
- Opción 2: Calcular la proporción de cargos B correspondiente a cada departamento. Para el departamento x se tiene:

$$B_x = \frac{P_x}{\sum_{x=1}^n P_x} \quad (5.1)$$

Si C es el número de cargos disponibles para la asignación, entonces los cargos a asignar al departamento x se denota por A_x :

$$A_x = B_x * C \quad \forall x \in [1 \dots n] \quad (5.2)$$

Teniendo en cuenta las consideraciones realizadas en las distintas estructuras mostradas, se procede a realizar algunas pruebas que permitan delimitar el modelo final. Las mismas se muestran en la

sección siguiente.

5.1.2 Pruebas preliminares

Se realizaron algunas pruebas en el módulo de Lógica Difusa de Matlab[®] para observar el comportamiento del sistema. El siguiente ejemplo hace uso de una estructura de variables que puede ser extendida al resto de las estructuras propuestas, y se ejecuta con el fin de avanzar en la elaboración de un modelo preliminar de Lógica Difusa. Los datos utilizados son ficticios.

Primera prueba

El proceso se da en las siguientes fases:

Fase 1: Procesamiento preliminar de datos.

Se tienen los porcentajes de Actividades de Investigación de dos Departamentos *A* y *B*.

Se tienen las proporciones de profesores jubilables de cada departamento.

Para el procesamiento preliminar, debe establecerse el peso o importancia para cada área de actividades. En este, caso para simplificar el problema, se tiene sólo un área, por lo que el peso (determinado como α) será de 100 %. Dado α y el Porcentaje de Actividad de Investigación (*PAI*) de cada departamento, se calcula el siguiente indicador, el cual refleja el rendimiento del departamento en dicha actividad:

$$I = \frac{PAI}{\alpha}$$

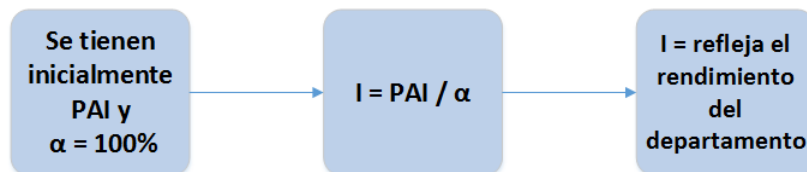


Figura 5.4: Flujo del proceso.

Luego, el indicador *I*, será una de las entradas al Sistema de Inferencia Difuso, junto con la variable Proporción de Profesores Jubilables (*PJ*).

Departamento	PAI	α	I	PJ
A	53,9	100 %	0,54	0,273
B	28,4	100 %	0,28	0,286

Tabla 5.7: Datos para los Departamentos A y B.

El sistema de inferencia difuso tendrá como salida el conjunto difuso Prioridad, el cual reflejará el nivel de prioridad para asignación de cargos que tenga el Departamento, como se muestra en

la figura 5.5.

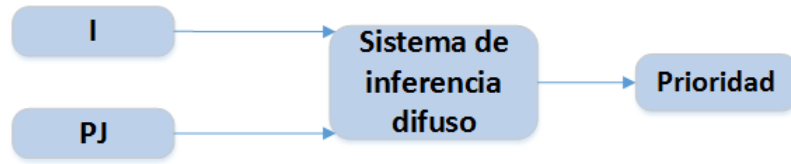


Figura 5.5: Estructura del Sistema de Inferencia Difuso.

El sistema de inferencia difuso contendrá las definiciones de las funciones de membresía para cada entrada, además de las reglas Si-Entonces que definan las relaciones entre las variables.

Fase 2: Se definen las funciones de membresía para cada variable. Utilizando la herramienta para Lógica Difusa de Matlab[®], se construyen las funciones tal como se muestra en las figuras 5.6, 5.7 y 5.8:

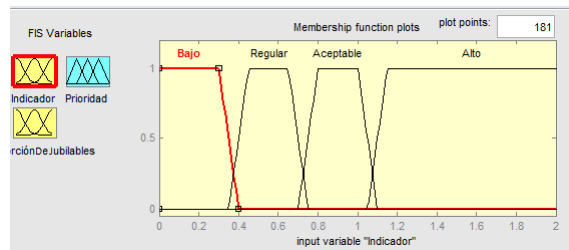


Figura 5.6: Funciones de Membresía para la variable I Indicador.

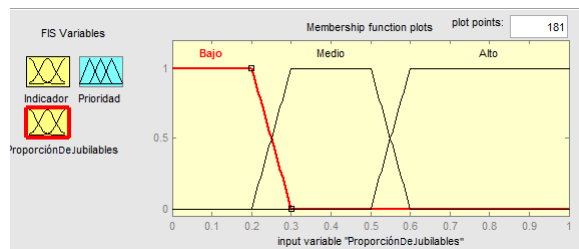


Figura 5.7: Funciones de Membresía para la variable PJ (Proporción de profesores jubilables).

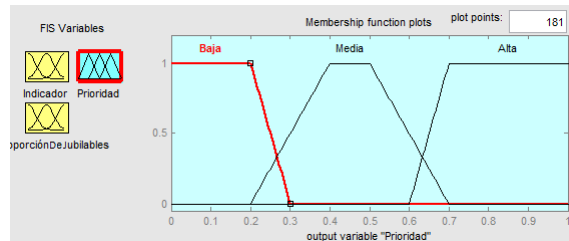


Figura 5.8: Funciones de membresía para la salida Prioridad.

El sistema de inferencia de tipo Mamdani, consta entonces de dos variables de entrada y una variable de salida (ver figura 5.9).

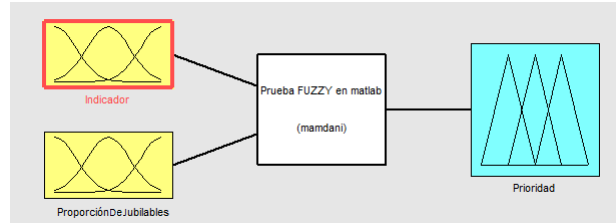


Figura 5.9: Sistema de inferencia difuso tipo Mamdani.

Las reglas Si-Entonces se establecen con ayuda de la herramienta (ver figura 5.10). Considerando que la variable I puede pertenecer a 4 conjuntos distintos, la variable PJ puede pertenecer a 3, la salida Prioridad puede resultar en 3 conjuntos diferentes, y asumiendo reglas con el operador AND , se construyeron 12 reglas distintas para evaluar.

Si I es Bajo y PJ es Bajo entonces Prioridad es Baja
 Si I es Bajo y PJ es Medio entonces Prioridad es Baja
 Si I es Bajo y PJ es Alto entonces Prioridad es Media
 Si I es Regular y PJ es Bajo entonces Prioridad es Baja
 Si I es Regular y PJ es Medio entonces Prioridad es Media
 Si I es Regular y PJ es Alto entonces Prioridad es Media
 Si I es Aceptable y PJ es Bajo entonces Prioridad es Baja
 Si I es Aceptable y PJ es Medio entonces Prioridad es Media
 Si I es Aceptable y PJ es Alto entonces Prioridad es Alta
 Si I es Alto y PJ es Bajo entonces Prioridad es Media
 Si I es Alto y PJ es Medio entonces Prioridad es Alta
 Si I es Alto y PJ es Alto entonces Prioridad es Alta

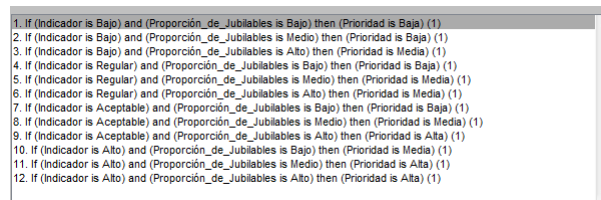


Figura 5.10: Conjunto de reglas en Matlab®.

Fase 3: Introduciendo variables de entrada para dos Departamentos A y B , y utilizando los datos recogidos en la Fase 1, se obtienen los resultados mostrados en las figuras 5.11 y 5.12.

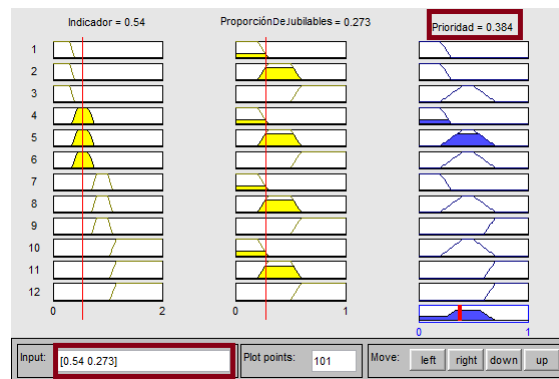


Figura 5.11: Resultados para el Departamento A.

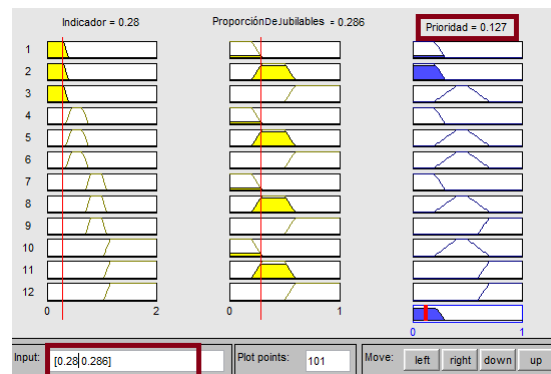


Figura 5.12: Resultados para el Departamento B.

Como se observa en las imágenes, el departamento con mayor Prioridad al momento de asignar nuevos cargos es el Departamento A.

Departamento	Salida (Conjunto)	Salida (Valor)
A	Media	0,384
B	Baja	0,127

Tabla 5.8: Resultados y comparación de los Departamentos A y B.

Segunda prueba

Otra prueba ejecutada con un enfoque similar en Matlab[®] incluye variables para las áreas de docencia, investigación y extensión, de tres departamentos. Los datos utilizados no pertenecen a ningún departamento real de la facultad. Los datos de entrada y la salida del sistema se especifican en la tabla 5.9.

Dpto	D	I	E	Prioridad
A	0,2	0,3	0,4	0,2508
B	0,3	0,5	0,1	0,2732
C	0,2	0,4	0,3	0,2508

Tabla 5.9: Datos de entrada. Segunda prueba.

Regla de ejemplo:

Si Indicador de Docencia es Alto e Indicador de Investigación es Alto e Indicador de Extensión es Alto entonces Prioridad es Alta.

Luego de obtenidas las prioridades para cada departamento, la repartición de cargos puede hacerse con enfoques distintos:

- Opción 1: Asignar cargo al departamento con mayor prioridad y repetir el proceso según el número de cargos disponibles.

Se genera la lista de departamentos ordenados por prioridad, en orden descendente. Se asigna entonces un cargo al departamento de mayor prioridad y seguidamente se repite el procesamiento de datos tal como en la figura 5.2, es decir, actualizando los datos para el departamento al que se le asignó el cargo, y reiniciando el ciclo de asignación según los cargos disponibles.

Salida:

PA 0,2508

PB 0,2732

PC 0,2508

- Opción 2: Calcular las proporciones de cargos correspondientes a cada departamento, tal como en la ecuación (5.1) especificada anteriormente en la asignación de cargos.

Y siendo C el número de cargos disponibles para la asignación, y n el número de departamentos, se asignan los cargos para cada uno como en la ecuación 5.2 establecida anteriormente.

Si el número de cargos es: 6, y se tienen 3 departamento a, b, c , entonces se calculan las proporciones de cargos B correspondientes a cada uno:

$$B_a = \frac{0.2508}{0.77428} = 0.32 \quad B_b = \frac{0.2732}{0.7748} = 0.35 \quad B_c = \frac{0.2508}{0.77428} = 0.32$$

Finalmente, los cargos a asignar (A) a cada departamento serían:

$$A_a = 1,92 \text{ cargos}$$

$$A_b = 2,1 \text{ cargos}$$

$$A_c = 1,92 \text{ cargos}$$

5.2 Construcción del modelo

En base a las investigaciones y pruebas realizadas, comparando las distintas opciones sobre agrupación de variables, modos de asignación de cargos, datos necesarios e influyentes en el rendimiento de los departamentos, se determinan aspectos relevantes que ayuden a elaborar la estructura final que recoja en mejor medida la necesidad de cargos de un departamento. Con esto en mente, se sigue el camino para determinar qué valores usar, de qué manera usarlos, qué enfoque tomar, cómo manejar valores implícitos, etc. En esta sección se especifican los puntos escogidos según la evaluación y pruebas y se desarrolla la estructura final para las variables y el sistema difuso.

5.2.1 Definición del sistema de inferencia

Luego de las pruebas realizadas, con base en los datos de los informes de actividades cumplidas, se reconstruye una nueva estructura considerando los análisis, resultado de las observaciones de las planillas y los baremos estudiados, y finalmente combinando los puntos que se destacan con mayor relevancia para la asignación de cargos. Para definir la estructura del modelo, en primer lugar se especifican los datos necesarios para el proceso, recordando que se utilizará un sistema de inferencia difusa para obtener el resultado que apoye la toma de decisiones.

Datos de entrada

Los datos requeridos para el cálculo de las distintas variables que son de relevancia como entrada al sistema difuso se especifican a continuación. Estos datos deben ser facilitados por el departamento o la escuela en conjunto.

- Datos de profesores (por departamento o escuela): nombres y apellidos, cédula, fecha de ingreso (fecha en la que el profesor inició el ejercicio de actividades en la universidad), escalafón (Titular, Asociado, Agregado, Asistente, Instructor), condición (Ordinario, Contratado, Jubilado activo, Interino, Visitante, Colaborador), dedicación, horas asignadas para las actividades del profesor según su Dedicación, estado (beca, sabático, permiso remunerado o no remunerado, jubilado activo, autorizado, plan de formación, activo, suspendido), carga administrativa (Cadm), escuela, departamento, área de conocimiento.
- Datos de materias (por departamento o escuela): código, nombre, escuela, departamento, área de conocimiento, horas de clases teóricas (HCT), horas de laboratorio o taller (HL), horas de práctica (HP), horas de campo (Hcamp), horas de seminario (Hsem), horas de régimen especial (HRE), límite de alumnos por sección.
- Datos sobre actividades cumplidas (por departamento, de los resúmenes de actividades cumplidas): departamento, AÑO, PAD, POAD, PAI, PAEX, PAM, PAA.
- De los inscritos (por materias del departamento): Código de materia, nombre de materia, número de inscritos, sección, profesor.

Los siguientes son constantes que serán utilizadas en el procesamiento de datos como se muestra en la sección 5.2.2.

- Factores de dedicación y factores de ponderación de horas.

Dedicación	Ponderación de horas
de = 1	FPHCT = 3
tc = 1	FPHL = 2
mt = 1/2	FPHP = 2
tconv = 1/3	FPHcamp = 2
jubact = 2/5	FPHsem = 2
	FPHRE = 1

FPHT = Factor de ponderación de horas teóricas.

FPHP = Factor de ponderación de horas prácticas.

FPHL = Factor de ponderación de horas de laboratorio.

FPHcamp = Factor de ponderación de horas de trabajo de campo.

FPHsem = Factor de ponderación de horas de seminario.

FPHRE = Factor de ponderación de horas de régimen especial.

El factor de ponderación HCT , para horas teóricas, se muestra aquí como el promedio aritmético de los factores para horas teóricas de la tabla (??). Alternativamente pueden usarse estos factores especificándolos para cada materia según los factores de mayor frecuencia, o según el conocimiento de experiencias anteriores. Para esto se manejaría una tabla alternativa con la información deseada, la cual sería usada posteriormente en el procesamiento de las variables. Según la tabla de ponderación de horas del Reglamento de Evaluación de Actividades de la Facultad de Ingeniería, los FPH dependerán de la modalidad de evaluación del profesor en la materia y del número de estudiantes que se hayan inscrito. Esto fue establecido para usarse en las planillas de informes de actividades de la facultad que debe presentar cada profesor de TC o DE.

- Datos constantes requeridos: Secciones por profesor (Ssp), porcentaje de actividades de investigación (ReqI), porcentaje de actividades de extensión. (ReqE), porcentaje de otras actividades de docencia (ReqOA), horas ponderadas de materia optativa, cargos u horas disponibles para la asignación.

5.2.2 Estructura del sistema difuso

Una vez definidos los datos necesarios para la construcción del modelo y por medio de los análisis, pruebas y observaciones realizadas anteriormente, se establecen las variables que se usarán como entrada al sistema de inferencia. Éstos cálculos se basan en los baremos de la Escuela de Eléctrica y

de la Facultad de Arquitectura mencionados anteriormente.

Cálculo de variables

En la siguiente sección se muestra el procesamiento de los datos para la construcción de variables intermedias y finales, de interés para el sistema difuso. Para cada materia del departamento, se tiene:

- HPT_i : Horas ponderadas totales semanales para cada materia.

$$\begin{aligned} HPT_i = & HCT_i * FPHT + HP_i * FPHP + HL_i * FPHL \\ & + Hcamp_i * FPHcamp + Hsem_i * FPHsem + HRE_i * FPHRE \end{aligned} \quad (5.3)$$

- $Pdem_i$: Se define como el número de profesores requeridos para la materia i .

$$Pdem_i = \frac{Sdem_i}{Spos_i} \quad (5.4)$$

$Sdem_i$: Es el número de secciones que deben ofertarse de la materia i , según la demanda de estudiantes.

$Spos_i$: Es el número de secciones de la materia i que pueden ser atendidas por profesores TC o DE (con un límite de horas para clases de 30 horas semanales).

Para calcular $Pdem_i$, se debe conocer primero el valor de cada una de estas variables intermedias. $Sdem_i$ viene dado por:

$$Sdem_i = \left\lceil \frac{Nalum_i}{Lalum_i} \right\rceil \in \mathbb{N} \quad (5.5)$$

$Nalum_i$: Es el número de alumnos que deben inscribir la asignatura i , según los inscritos en OREFI, o según datos de inscripción del departamento.

$Lalum_i$: Es el número de alumnos sugeridos por sección para la materia i .

De la misma forma, $Spos_i$ es un valor entero determinado por:

$$Spos_i = \left\lfloor \frac{HC}{HPT_i} \right\rfloor \in \mathbb{N} \quad (5.6)$$

Donde HC es una constante que representa las horas de clase disponibles de un profesor TC o DE (30 horas máximas para clases).

- $Pdem_a$: Se define como el número de profesores requeridos para el área a . Se calcula según la ecuación (5.7):

$$Pdem_a = \sum_{i=1}^n (Pdem_i) \quad \forall i \in a \quad (5.7)$$

Donde n es el número de materias pertenecientes al área de conocimiento a .

- $Pdisp_a$: Profesores disponibles de un departamento pertenecientes al área de conocimiento a .

$$Pdisp_a = \sum_{j=1}^m (fd_j) \quad (5.8)$$

Donde m es el número de profesores del área de conocimiento correspondiente a , y fd es el factor de dedicación del profesor.

- $Pdisp_d$: Profesores disponibles adscritos al departamento d .

$$Pdisp_d = \sum_{k=1}^r (fd_k) \quad (5.9)$$

Donde r es el número de profesores pertenecientes al departamento d .

- $Preq_a$: Profesores requeridos por área. Se refiere a la cantidad de profesores que hacen falta para cubrir totalmente la carga de secciones de materias pertenecientes al área a . Si un área de conocimiento cuenta con un total de 10 profesores, y en esa misma área se requieren 12 profesores (denotado por $Pdem_a$), entonces los profesores requeridos son 2. Se calcula como en la ecuación (5.10):

$$Preq_a = Pdisp_a - Pdem_a \quad (5.10)$$

Si $Preq_a$ es negativo, entonces el área de conocimiento a requiere de profesores para cubrir la demanda. Si $Preq_a$ tiene un valor positivo, los requerimientos de profesores del área a se consideran cubiertos. Un área cuya $Ptotal_a$ sea 10 y $Pdem_a$ sea 9, puede cubrir la demanda de profesores ($Preq_a = 1$). En el código que procesa estas variables, se reemplazan con 0 los valores de $Preq_a$ que sean positivos, lo cual significa que no hay profesores requeridos, de esta forma $Preq_a$ será siempre menor o igual que cero (≤ 0).

- $Pmej_a$: Profesores en mejoramiento por área de conocimiento. Se refiere al total de profesores que está en estado de mejoramiento, lo cual incluye actividades como becas, sabáticos, permisos, etc. En los datos de entrada se diferencian de los profesores que tienen estado “activo”. Se calcula según la ecuación (5.11):

$$Pmej_a = \sum_{j=1}^m (fd_j) \quad \forall j \mid \text{estado} \neq \text{“activo”} \quad (5.11)$$

Donde m es el número de profesores del área de conocimiento a .

- $Pmej_d$: Profesores en mejoramiento por departamento. De forma similar a la anterior, se contabilizan los profesores en estado de mejoramiento para cada departamento según la ecuación (5.12).

$$Pmej_d = \sum_{k=1}^r (fd_k) \quad \forall k \in d \quad (5.12)$$

Donde r es el número de profesores pertenecientes al departamento d .

- $Padm_d$: Se refiere a la carga administrativa del departamento d expresada en número de profesores. Se toman en cuenta los cargos con más de 10 horas administrativas.

$$Padm_d = \sum_{i=1}^n \frac{Cadm_i}{HC} \quad \forall i : 1 \text{ hasta } n \mid \text{carga administrativa} \geq 10 \quad (5.13)$$

- $Hsob_i$: Es el número de horas sobrantes de la carga académica. Si un profesor perteneciente al área a , es asignado a 2 secciones ($Spos_i$), de la materia i , con carga de 12 horas ($Hpond_i$), entonces las horas sobrantes son 6, suponiendo que la constante $HC = 30$. Así se muestra en la ecuación (5.14). Se denotan como “horas sobrantes” debido a que no son suficientes para cubrir una nueva sección completa con carga de 12 horas, en este caso.

$$Hsob_i = HC - Hpond_i * Spos_i \quad \forall i : i\text{-ésima materia} \quad (5.14)$$

- $Opos_i$: Electivas posibles según horas sobrantes de la carga académica. Esta son las materias que pueden ofertarse según las horas sobrantes calculadas anteriormente por cada materia (ver ecuación 5.14), y las horas ponderadas de la materia electiva (Hop). Se utiliza un valor constante promedio general para el valor Hop , tal como se muestra en la ecuación (5.15).

$$Opos_i = \frac{Hsob_i}{Hop} \quad (5.15)$$

- Op_a : Electivas posibles por área. Es la sumatoria de las electivas que pueden ofrecerse según las materias pertenecientes al área de conocimiento a .

$$Op_a = \sum_{i=1}^n Opos_i \quad \forall i \in a \quad (5.16)$$

Donde n es el número de materias pertenecientes al área a .

- Op_d : Electivas posibles por departamento. Es la sumatoria de las electivas que pueden ofrecerse según las áreas de conocimiento pertenecientes al departamento d .

$$Op_d = \sum_x^p Op_x \quad \forall x : a, b, c, \dots, p \quad | \quad x \in d \quad (5.17)$$

- $Psob_a$: Es el número de profesores disponibles según horas sobrantes de la carga académica, por área. Este valor representa la cantidad de profesores en base a las horas sobrantes de la carga académica, tal como se expone en la ecuación (5.14). Es decir, a un profesor se le asignan una o mas secciones de una materia del área a de acuerdo a su carga horaria, según su dedicación, es posible que sobren algunas horas por utilizar. Por ejemplo, si una materia obligatoria tiene 2 secciones de 12 horas ponderadas, y las horas máximas de clases de un profesor son 30 (HC), entonces las horas sobrantes son 6. Si Hop es menor o igual que 6, entonces se considera que hay horas disponibles para dictarla. Estas horas restantes se contabilizan para el área a . La ecuación (5.18) simplifica el cálculo. De esta manera se procede con cada materia del área a y según ellas se representa la fracción de profesores disponibles según las horas sobrantes, que pueden impartir materias electivas, según Hop .

$$Psob_a = \frac{(Pdisp_a * Hop)}{HC} \quad (5.18)$$

- $Psob_d$: Es el número de profesores disponibles según horas sobrantes de la carga académica, por departamento.

$$Psob_d = \sum_x^p Psob_x \quad \forall x : \{a, b, c \dots p\} \quad | \quad x \in d \quad (5.19)$$

- PAI y $PAEX$ y $POAD$ son los valores tomados de las planillas resúmenes de informes de actividades del departamento (Porcentajes).
- $Pjub_d$: Profesores jubilables del departamento d . Son los profesores que cumplen con el tiempo necesario para iniciar el proceso de jubilación. Se toma en cuenta su fecha de ingreso, con un límite de 25 años para jubilación, tal como muestra la ecuación (5.20).

$$Pjub_d = \sum_{k=1}^r (fd_k) \quad \forall k \quad | \quad fecha \quad actual - (fecha \quad ingreso)_k \geq 25 \quad (5.20)$$

Donde r es el número de profesores pertenecientes al departamento d .

Indicadores y conjuntos difusos

En este apartado se desarrolla el enfoque para el modelo difuso, utilizando las variables procesadas anteriormente. Se inicia construyendo los siguientes Indicadores:

- I1: Concerniente al rendimiento en actividades de Investigación del departamento.
- I2: Concerniente al rendimiento en actividades de Extensión del departamento.
- I3: Concerniente al rendimiento en Otras actividades de docencia del departamento.
- I4: Concerniente a la proporción de profesores jubilables del departamento.
- R: Concerniente al déficit de profesores del departamento.

I1: Actividades de Investigación.

Para este indicador se toma el porcentaje de rendimiento PAI de los resúmenes de informes de actividades cumplidas del departamento. El indicador se construye como sigue (5.21):

$$I1 = \frac{PAI}{ReqI} \quad (5.21)$$

donde $ReqI$ es un valor fijado en consenso como el valor deseado del porcentaje de actividades de investigación.

Las funciones de pertenencia para este indicador, mostradas en la figura 5.13, se dividen como:

1. Muy Bajo: El rendimiento del departamento en las áreas es muy bajo. No se considera prioritario para la asignación de cargos.
2. Bajo: El rendimiento del departamento en las áreas es bajo. La prioridad es escasa para la asignación de cargos.
3. Moderado: El rendimiento aún se mantiene por debajo de lo deseado. Se considera medianamente la asignación de cargos.
4. Alto: El rendimiento cubre los valores deseados.
5. Muy alto: El rendimiento excede los valores propuestos, lo que se considera como desequilibrio del departamento respecto a los requerimientos. No se considera prioritario asignar cargos.

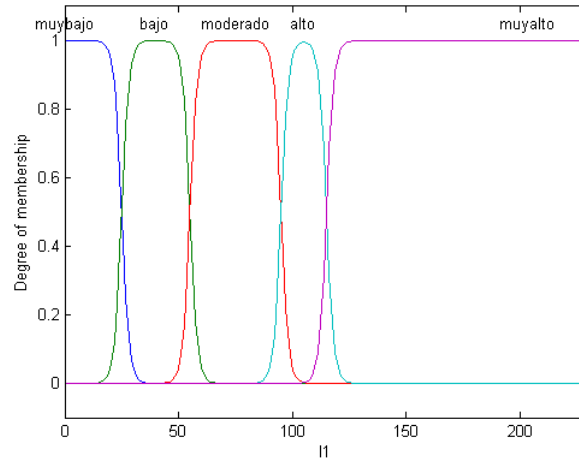


Figura 5.13: Funciones de pertenencia para I1.

I2: Actividades de Extensión.

Se construye análogamente al indicador I1 con el uso de la variable *PAEX* del informe de actividades, y *ReqEx* como requerimiento para el área (5.22). Las funciones de pertenencia son equivalentes.

$$I2 = \frac{PAEX}{ReqEx} \quad (5.22)$$

I3: Otras actividades de docencia.

De manera similar a I1 e I2, utilizando *POAD* de los informes de actividades y *ReqOA* como requerimiento de entrada se construye I3 (5.23). Las funciones de pertenencia son equivalentes a las dos anteriores:

$$I3 = \frac{POAD}{ReqOA} \quad (5.23)$$

I4: Proporción de profesores jubilables.

Este indicador se construye con base en la variable *Pjub* y el total de profesores del departamento, de la siguiente manera (5.24):

$$I4 = \frac{Pjub_d}{Ptotal_d} \quad (5.24)$$

Las funciones de pertenencia para este conjunto son:

1. Bajo: Se considera una proporción tolerable de profesores jubilables del departamento que no afecta en gran medida las necesidades.
2. Moderado: La proporción requiere de atención por encontrarse en un nivel que puede afectar la disponibilidad de profesores en los períodos próximos.

3. Alto: Se considera de alta prioridad la asignación de profesores para cubrir el aumento de la demanda próxima.

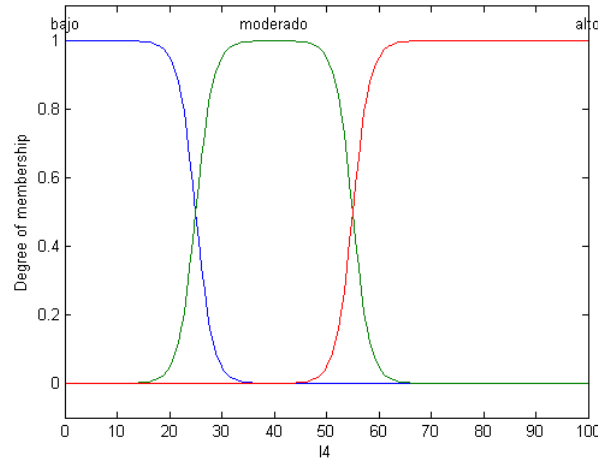


Figura 5.14: Funciones de pertenencia para I4.

R: La cantidad de profesores requeridos en el departamento estaría determinado por (5.25):

$$R = Preq_d - Pmej_d - Padm_d + Psob_d \quad (5.25)$$

Siendo el valor de $Preq_d$ siempre menor o igual a cero, se adicionan al déficit los profesores que están en mejoramiento, y los profesores no disponibles debido a las cargas administrativas. Además, se suman los profesores que sí están disponibles según las horas sobrantes en la carga académica ($Psob_d$), de ésta manera se obtiene el valor total para el déficit de profesores R que presenta el departamento d . R será un valor menor o igual que cero (≤ 0).

Los primeros 4 indicadores serán procesados bajo el sistema difuso a través de un conjunto de reglas, para obtener como resultado la prioridad de cada departamento en la asignación de cargos. Este conjunto de salida cuenta con 6 funciones de pertenencia (ver figura 5.15):

1. Mínimo: No tiene prioridad alguna.
2. Muy baja: Prioridad es de muy poca relevancia para optar a cargos nuevos.
3. Baja: Poca urgencia para la asignación de cargos.
4. Media: Prioridad media, con oportunidad a asignaciones en última instancia.
5. Alta: Prioridad relevante para la asignación de nuevos cargos en primera instancia.
6. Muy Alta: Prioridad suficiente para la asignación de cargos urgente en primera instancia.

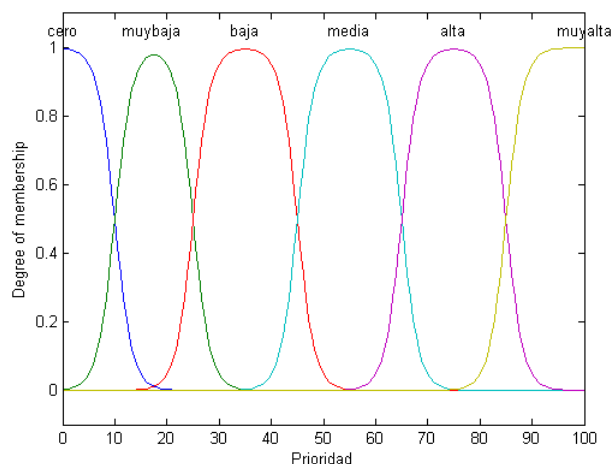


Figura 5.15: Funciones de pertenencia para Prioridad.

Reglas del sistema difuso

La cantidad de reglas totales, según el número de conjuntos y funciones de pertenencia, son 375. Para definir la salida de cada regla, se procedió a evaluarlas cuidadosamente considerando analogías y equivalencias en reglas similares. El conjunto de reglas se muestran en los apéndices ??, ??, ??, ??.

En la fase final de asignación, los cargos se asignan según el valor defusificado de la salida obtenida. El conjunto salida da como resultado un valor entre 0 y 100, el cual será tomado como el porcentaje de cargos que deben ser asignados al departamento. Se genera una lista ordenada de los departamentos según su prioridad.

Una vez obtenida la prioridad para cada departamento, se aplica este porcentaje al número de cargos requeridos. De esta forma se obtiene la cantidad de cargos que deberían asignarse, ajustada al rendimiento que tiene el departamento en las actividades. El número de cargos requeridos refleja el déficit de profesores que demuestra tener el departamento según demanda y oferta de profesores. Los cálculos correspondientes fueron especificados anteriormente. Esta lista de información estará ordenada según la prioridad obtenida.

Finalmente, en el momento de la asignación de cargos, se debe tener conocimiento de cuántos cargos están disponibles para la repartición. Teniendo la lista ordenada de prioridades y cargos que deberían asignarse, se realiza la asignación según los cargos disponibles hasta agotarse.

En este caso existe la posibilidad de que los cargos no sean suficientes para cubrir la necesidad de todos los departamentos involucrados según el orden de la lista. El orden de la lista justifica la prioridad de cada departamento, en este sentido los que están en posición menos privilegiada pueden no percibir asignación alguna según la disponibilidad. El escenario ideal sería aquel en el que los cargos disponibles son suficientes para cubrir las asignaciones sugeridas por el sistema, pero el hecho de la escasez de cargos precisamente es la base para el ordenamiento por prioridades de la lista resultado

según el rendimiento.

En la tabla 5.10 se muestra un ejemplo de asignación, con la lista ordenada según las prioridades, suponiendo que se tienen 5 cargos disponibles para la asignación. Nótese que en este caso se pudo cubrir la demanda del departamento con menos prioridad sólo parcialmente.

Departamento	Prioridad	R	Cargos a asignar	Cargos asignados
A	60	5	3	3
B	45	3	1,35	1,35
C	30	4	1,2	0,65

Tabla 5.10: Asignación de cargos.

*Se recuerda que se interpreta la cantidad asignada en base a cargos de *DE* (ver tabla ??); es decir, de 40 horas:

1 cargo DE: 1

1 cargo TC: 0,87

1 cargo MT: 0,45

1 cargo Tconv: < 0,3

Con esta nueva estructura se espera abarcar la información de influencia en la necesidad de los departamentos, ya que toma en cuenta un amplio número de factores referidos a la situación estudiantil y docente, administrativa, de investigación y extensión simultáneamente. En la figura 5.16, se ilustra el proceso y las variables involucradas.

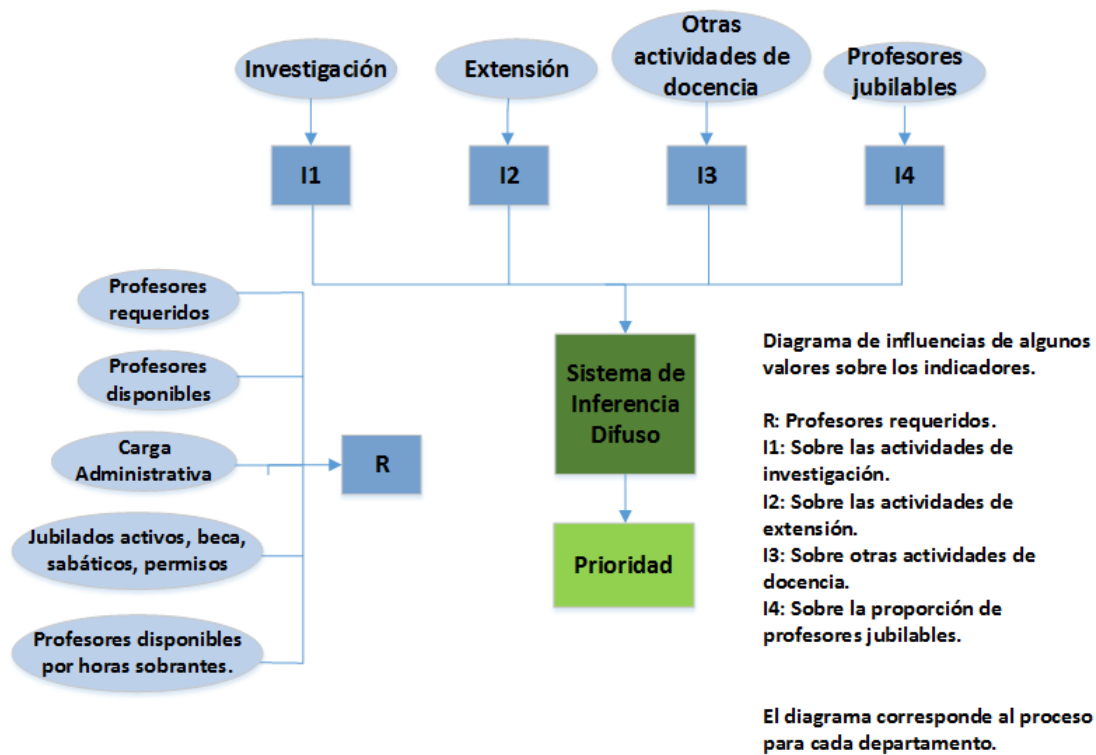


Figura 5.16: Estructura definitiva para el sistema difuso.

En las figuras 5.17, 5.18, 5.19, se muestran con mayor detalle los datos que son de influencia para cada variable calculada.

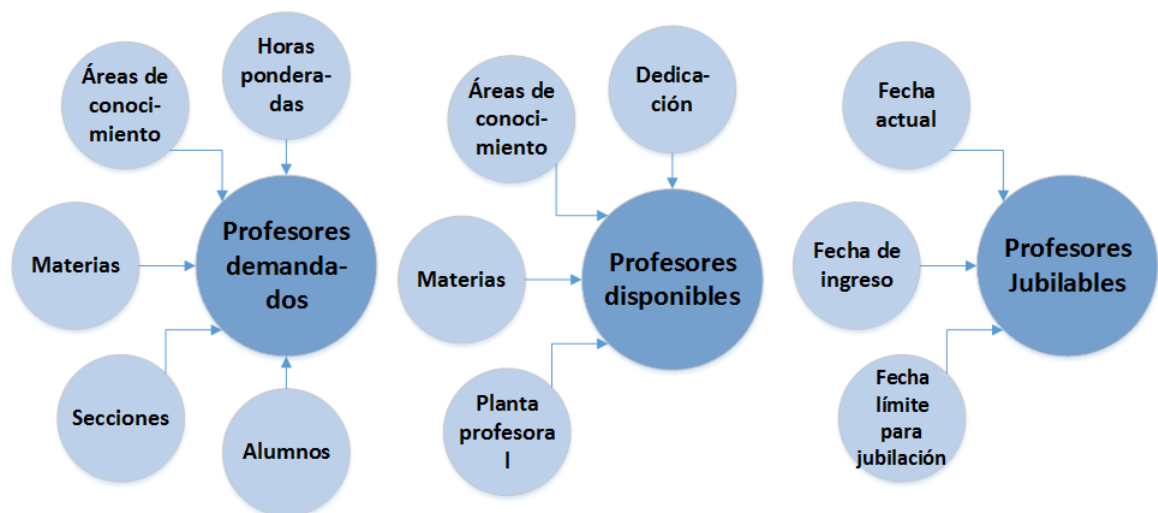


Figura 5.17: Diagrama de influencias. Profesores demandados, disponibles y jubilables.

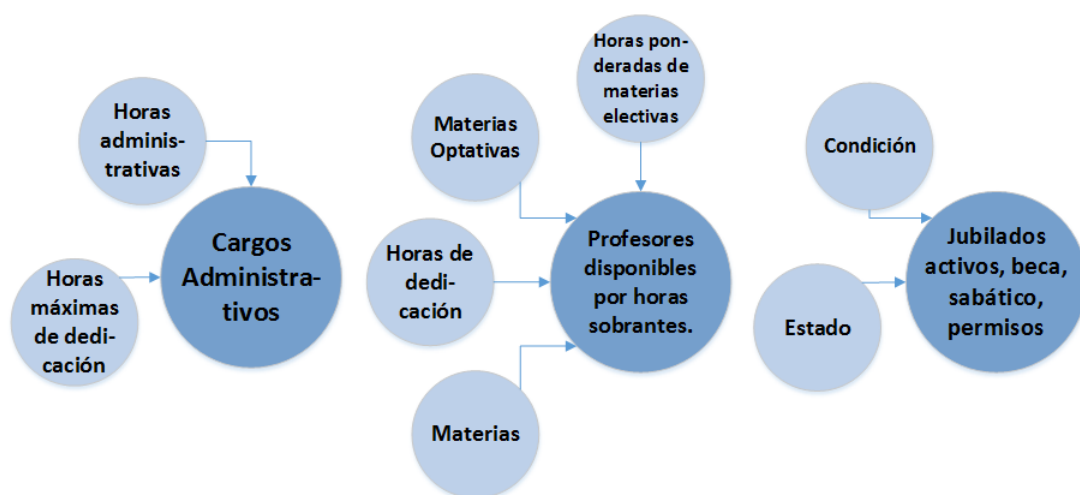


Figura 5.18: Diagrama de influencias. Cargos administrativos, horas sobrantes, mejoramiento.

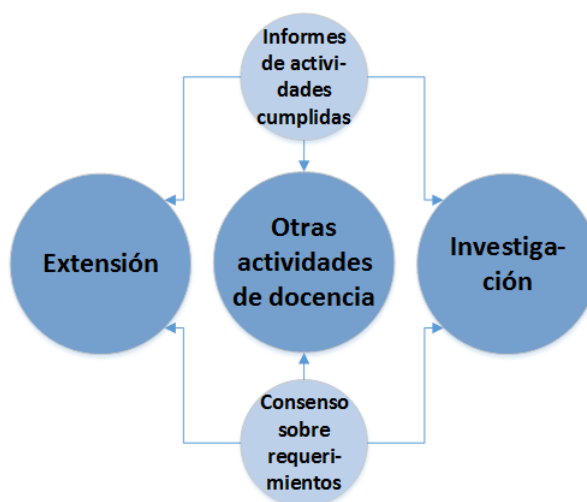


Figura 5.19: Diagrama de influencias. Investigación, extensión, otras actividades de docencia.

Capítulo 6

Conclusiones

Se pudo corroborar que el uso de la Lógica Difusa resulta ser un buen enfoque para el tratamiento del problema de asignación, permitió de manera flexible la inclusión de variables y su manejo e interpretación para los resultados. La naturaleza del problema, que involucra información imprecisa y subjetiva, puede suponer un riesgo en la construcción de modelos, por lo que debe definirse y ajustarse correctamente para el problema en cuestión, para que arroje resultados veraces. La Lógica Difusa se muestra como una herramienta potente que ataca este tipo de obstáculos.

Durante la realización de este proyecto se pudo observar cómo se evalúan las actividades de los profesores de la Facultad de Ingeniería, y los factores principales involucrados en el proceso. Los análisis realizados como base para el inicio de este proyecto, sobre este sistema de evaluación, permitieron detallar algunos puntos importantes relacionados con su aplicación, tal como especifican los reglamentos correspondientes. A partir de dichos análisis, como se expone en el informe anexo a este documento, se pretende aportar ideas y recomendaciones para mejorar la evaluación de los profesores por esta vía. Se facilita el uso de nuevos formatos dirigidos a este fin. Para mejorar la utilidad de los valores recogidos por los informes y resúmenes de actividades se indican algunos puntos que fueron observados durante el proyecto.

El modelo elaborado abarca las áreas de actividades que tienen lugar en la Facultad de Ingeniería: docencia, investigación, extensión, mejoramiento y administración, tal como se propuso estudiar en este proyecto; además de los aspectos relacionados con las demandas a nivel estudiantil, y la disponibilidad y rendimiento de los profesores. Durante las pruebas realizadas se pudo comprobar el funcionamiento adecuado del modelo, verificando los cambios en los resultados, mediante variaciones sobre los datos utilizados. Las reglas que conforman el Sistema de Inferencia Difuso pueden ser modificadas cuidadosamente de acuerdo con las demandas del momento, igualmente pueden ser modificadas las constantes requeridas, establecidas mediante consenso, de forma que el modelo

cuenta con una capacidad de adaptación deseable para la evaluación de los conceptos subjetivos inherentes al problema de asignación de cargos. La extrapolación del modelo a otras unidades académicas es posible, en la medida que se adapten los registros necesarios consistentes en los datos de entrada requeridos. Debido a estas razones se puede concluir que el modelo logró cubrir los objetivos planteados en el proyecto, y además cuenta con el potencial necesario para apoyar la toma de decisiones en la Facultad de Ingeniería de la ULA.

Las herramientas utilizadas para la implementación del modelo se adaptaron con facilidad a las necesidades que se presentaron en cuanto al manejo de grandes cantidades de datos y cálculos estadísticos, por medio del lenguaje de programación *R*, y la fase correspondiente a los sistemas de inferencia difusa se completó de manera satisfactoria, a través del software matemático Matlab[®].

6.1 Recomendaciones

Sobre los datos y su recolección, se recomienda que su recopilación, orden y archivado correspondientes a los profesores de los departamentos, sean realizados bajo pautas estándares de tal manera que sea relativamente fácil su ubicación, revisión y consulta. El registro digital de estos datos, adicional al registro físico, permitiría un acceso más rápido de la información disponible, así como también la elaboración fiable de los resúmenes de actividades para cada departamento.

Se recomienda una cuidadosa revisión y verificación de los informes entregados, así como también su agrupamiento en los resúmenes, tanto por parte del profesor remitente como por parte del personal encargado de su archivo. De esta forma se asegura que los valores sean confiables y útiles para el departamento.

Los profesores y demás miembros involucrados en su evaluación, deben estar suficientemente informados sobre los reglamentos, sus contenidos, sus deberes y derechos. De esta forma mejorar el registro de informes de actividades cumplidas y propuestas, y la comprensión e interpretación de los valores que ellos contienen. Con un adecuado conocimiento de los reglamentos y deberes se disminuyen las dudas y los errores causados debido a la confusión al momento del registro de informes, como el llenado de tareas compartidas, las tareas que pueden ser incluidas, los porcentajes y horas tabuladas, entre otros. Se espera que los nuevos formatos elaborados sean utilizados correctamente con este propósito.

La automatización de la recolección de los registros de evaluación y demás información de los profesores y actividades que sean de importancia para los informes, se perfila como un posterior proyecto que pudiese ponerse en marcha para la mejora de los procesos de asignación de cargos, y en general para el análisis del rendimiento de los profesores. Esto puede estar dirigido a la construcción de

un sistema para reportes y evaluación profesoral. Debido a las incidencias y confusiones encontradas durante la revisión de informes y resúmenes, cabe preguntarse si la estructura del reglamento para la evaluación es la adecuada o debe adaptarse a las necesidades actuales del personal docente.

Para el mejoramiento del modelo planteado se sugiere recoger mayor información y estudiar más a fondo el problema, dando mayor importancia a la opinión de los miembros involucrados en la asignación de cargos, consultando las visiones sobre la importancia de las áreas estudiadas y la interpretación de las soluciones, lo cual sirve como conocimiento experto que imprima mayor utilidad a los resultados.

Bibliografía

- Banks, J., Carson, J. S., Nelson, B. L., y Nicol, D. M. (2010). *Discrete-Event System Simulation*. Pearson, 5 edition.
- Fawcett, J., Quin, L., y Ayers, D. (2012). *Beginning XML*. Wrox, 5 edition.
- Fuentes, J. M. (2009). *Manual de AJAX*. 2 edition.
- Fundation, D. S. (2015). *The web framework for perfectionists with deadlines*. 5 edition.
- Krasner, G. E. y Pope, S. T. (1988). A description of the model-view-controller user interface paradigm in the smalltalk-80 system. Technical report, ParcPlace Systems, Inc., Mountain View.
- Machuca, C. A. M. (2011). Estado del arte: Servicios web. Technical report, Universidad Nacional de Colombia, Bogotá, Colombia.
- Marcano, G. (2015). Desarrollo de un servicio web para el simulador de eventos discretos galatea. Technical report, Universidad de Los Andes, Merida, Venezuela.
- Oracle (2015). Mysql 5.7 :: Reference manual 1.3.1 what is mysql?
- Pantoja, E. B. (2004). El patrón de diseño modelo-vista-controlador (mvc) y su implementación en java swing. Technical report, ParcPlace Systems, Inc.
- PostgreSQL (2016). Postgresql about.
- Pérez, J. A. G. (2016). Diseño e implementación de un sistema de monitoreo de redes orientado a la recolección masiva de datos. Technical report, Universidad de los Andes, Mérida, Venezuela.
- Quiroga, M. G. (2011). Estado del arte: Servicios web. Technical report, Universidad Politecnica de Catalunya, Barcelona, España.
- Rengifo, B. (2011). Desarrollo de un servicio web para la modeloteca del sistema nacional de simulación. Technical report, Universidad de Los Andes, Merida, Venezuela.
- Rodríguez, J. (2010). Cálculo de capacidad de matrícula y demanda de profesores.

- Santana, Y. C., Ángel Mateo Pla, M., y Terol, J. P. (2004). Herramienta de simulacion distribuida mediante agentes moviles jade. Technical report, Conferência IADIS Ibero-Americana.
- Uzcátegui, M., Dávila, J., y Tucci, K. (2011). Galatea: una historia de modelado y simulación. *Revista Ciencia e Ingeniería*, (1316-7081):85–94.
- Viloria, F. J. (2013-2015). Cálculo de demanda de profesores.