



Universidad de las Ciencias Informáticas
Facultad 2

Sistema informático para la conformación automática de
tribunales de defensa de trabajos de diploma
en la Facultad 2. v1.0

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Autores:

Reydel Capote Coipel
Yerandy Fernández García

Tutores:

Ing. Bárbara Triana Morales
Ing. Suany Leyva Hernández

Co-Tutores:

Lic. Carlos Luis Milian del Valle
MSc. Susel García Cedeño

La Habana, 2015
“Año 57 de la Revolución”



"El hombre nunca sabe de lo que es capaz hasta que lo intenta."

Charles Dickens

Declaración de Autoría:

Declaramos ser autores de la presente tesis Sistema informático para la conformación automática de tribunales de trabajos de diploma en la Facultad 2 v1.0 y autorizamos a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio. Para que así conste, firmamos la presente a los ____ días del mes de _____ del año _____.

Reydel Capote Coipel

Yerandy Fernández García

Ing. Bárbara Triana Morales

Ing. Suany Leyva Hernández

MSc. Susel García Cedeño

Lic. Carlos Luis Milian del Valle

Datos de Contacto:

Bárbara Triana Morales

Correo Electrónico: bt triana@uci.cu

Experiencia como Docente: 7 años.

Categoría docente: Asistente.

Profesora de Subsistemas de Organizaciones (Pregrado UCI).

Jefa de Departamento de IGSW facultad 2.

Suany Leyva Hernández

Correo Electrónico: slhernandez@uci.cu

Asesora de Mercadotecnia, Centro TLM.

Susel García Cedeño

Correo Electrónico: suselgc@uci.cu

Experiencia como Docente: 6 años.

Categoría docente: Asistente.

Título Académico: Master.

Licenciada en Psicología.

Profesora de Metodología de la Investigación Científica (Pregrado UCI).

Jefa de Departamento de Gestión Extensionista.

Carlos Luis Milian del Valle

Correo Electrónico: cluismilian@uci.cu

Experiencia como Docente: 12 años.

Categoría docente: Asistente.

Profesor de Inteligencia Artificial (Pregrado UCI).

Jefe de asignatura de Inteligencia Artificial 1 y 2.

A mi mamá por ser la persona más importante de mi vida, por ser mi orgullo e inspiración.

A mi tata y mi bebé bello que significan la fuerza que me ayudó a lograr este triunfo.

A mis abuelos que no están físicamente conmigo pero sé que estarían muy orgullosos de ver a uno de sus nietos formarse como profesional, siempre los llevaré conmigo.

A toda mi familia en especial mi tía tati, Yailén, Andrés, mi ahijada y mi mami por todo su apoyo incondicional.

Reydel

A mi mamá que no me alcanzan las palabras para agradecerle y decirle que la quiero con todo mi corazón, que esta tesis y todo lo que he logrado es para ella y por ella.

A mi papá por ser mi amigo, mi hermano y mi padre en los momentos que más lo he necesitado, por darme el aliento que me hacía falta para levantarme y luchar por lo que quiero y creo que es correcto, mi carrera va para ti.

Para mi pequeña Eimily, ahora no lo sabes, pero tú eres el amor de mi vida, ya me encargaré de demostrártelo con cada abrazo, beso, castigo y lágrima que te pueda dar. Tú eres la razón que tengo para vivir y seguir luchando por ver en tu carita linda esa sonrisa que hace que me olvide de todo y me hace luchar porque te encuentres siempre segura al lado de tu mamá y tu papá. A ti te dedico mi tesis, mi carrera y mi vida.

Yerandy

Reydel Capote Coipel:

Agradecer: A mi madre que es la única persona que siempre está ahí en mi vida para lo que necesite, por todos los momentos en los que yo pensé que no podría seguir adelante ella supo aconsejarme y decir lo que tenía que escuchar, no solo como madre, ni como amiga que es, sino también con el rol de padre que le ha tocado asumir...Gracias, todo esto es por ti y para ti.

A mi hermana Maylín que crecí junto a ti, somos inseparables y lo seremos siempre...Gracias por existir para mí.

A mi bebé bello Ismelito por poder ser un ejemplo para ti en el futuro y por no dejarme estudiar la tesis porque querías jugar...Te adoro mi niño.

A mi padrastro por ayudarme a llegar hasta aquí y apoyar a mi mamá.

A Ismaelito por ser un cuñado un poco refunfuñón pero para mí nunca hay un “no”...Gracias.

A toda mi familia en especial a mi tía tati que es incondicional, mi primita Yailen, mi tía Yainet y mi tío José, Ramón y Miriala, mis primos Arial, Ana María, Deynys no sabes la ayuda tan grande que me has dado, Ricardo, Zarita y Evita por no dejarme estudiar, a Tony gracias por tu ayuda, mis hermanos Taiché y Raudel, mi Mami y mi ahijada...Los quiero a todos.

A mis tutores, Baby, por haber confiado en mí desde que nos conocimos nos llevamos muy bien y nunca dudaste que yo podría llegar hasta aquí. Suany, esa sonrisa con que nos recibiste cuando te enteraste que nos tutorabas fue lo que más seguridad nos dio. Susel, reamente aprendimos algo nuevo cada vez que nos reunimos. Carlos agradecer que cada crítica siempre fue constructiva...Gracias sobre todo porque jamás existió un después.

A los profesores que más contribuyeron a mi formación Aliennis, Iris, Imirys, Gendor, Bárbara, Julio, José Norberto, Lionid, Yusniel, Yuset, Miguel...Gracias.

A Aylín que el tiempo se ha encargado de demostrarnos que no hay que permanecer juntos si la amistad es sólida...Dayron, Muchas gracias por todo.

A mis amigos del IPI que hoy siguen demostrándome que están y estarán siempre para lo que necesite, Anisbel eres mi confidente y lo serás siempre, Migue mi papá...recuerdas, Heidi mi negrona, el Lachy, Ively, La Yuri...ridy, Lisette, Orisbel, Yaimara, Jasiel...Siempre los recuerdo.

A mi compañero de tesis y hermano Yerandy, realmente es un honor haber transitado este camino contigo no solo la tesis sino también toda la universidad siempre fuimos inseparables...Gracias. A tu familia, que es como la mía Gladyta, Gilbe, Lisbet y mi sobri Eimily.

A mis compañeros de aula que algunos son y serán más que eso, grandes amigos, en especial Cary...mi vida, Darío...el pelu, Yoel, Meli, Yassel, Jose, Sol, Clara, Raúl, Yise, Diana, en fin todos.

A las amistades conocidas aquí que al igual perduraran, Yeny, Carlos, Boza, Azneny, pichu y más... Llegué a la meta...pero no es el final del camino...

Yerandy Fernández García:

A las personas que me han ayudado y permitido crecer como persona y profesional, empezando por mis padres que han sido mi guía y soporte a los largo de mi vida y carrera. Agradecerle a Lisbet por estar ahí siempre para mí, por su consejo, apoyo, cariño y amor incondicional te amo, gracias a ti por darme lo que más quiero en esta vida, ahora entiendo a mi mamá, lo que significa un hijo en la vida de una persona, te amo mi niña.

A mi familia perfecta, lo quiero a todos y gracias por educarme como una persona de bien y darme todo el cariño del mundo. A mi hermanita grande Yeraynis, mi primo y hermano que no tuve Reidys, a mis abuelos Arcadio y Rosalina (por malcriarme tanto), a mi mimma linda, las adoro; a mis tíos Roly y Panchy, a Omaida, Yani, Yare, Ale, que desde siempre han estado conmigo para darme chucho y quererme, los quiero y gracias por ser siempre así.

A Gilbertico y Lorena, a Gilbe porque has estado al lado de mi mamá todo este tiempo y apoyándome, muchas gracias por ser parte de mi vida también. A mi familia Yunio y Daylín, ustedes son mis hermanos para siempre, siempre han estado ahí para mí, gracias por tantos tintes y estas libritas de más que no salieron solas. A mi sangre Samuel, que desde siempre estás, con los problemas, las pesadeces, compartiendo los buenos y sobre todo los malos momentos en la vida, te quiero mi hermano.

A mi suegra Dianabel que más que suegra me ha acogido y querido como un hijo, esas lágrimas que has soltado por mí nunca se me van a olvidar. A Pepe, que más que abuelo eres como mi padre, gracias por las cosas que he aprendido y todos los buenos momentos que paso contigo. Agradecer a mi compañero de tesis y hermano Reydel que sin él no hubiese tesis, por estar ahí para correr conmigo cuando y donde hiciera falta y por soportarme, A mi otra madre Ofe por siempre y a Maylín, también son familia para mí. A mis amigos del IPI que se mantienen y seguirán, Carlos, Daniel, Aníbal, Jiubel, Pepe. A toda la gente del aula, en especial a los más cercanos, Yassel, José, Darío y Cariño. A las amistades que fueron surgiendo en la universidad y que se quedan para toda la vida, Yei, Iris, Anaylis, Yamila, Alexei y chuchú. A los profesores que conocí y les agradezco, Joan, Yuneiry, Yuset, Adrián, Dieter y Leonid. A mis Tutores, gracias baby por pelearme tanto por los aretes y las chancletas, Suany por siempre estar al tanto con sus críticas constructivas y su linda sonrisa, a Susel por su carisma, su tiempo dedicado y permitírnos aprender siempre cosas nuevas y al profe Carlos por hacernos correr con los detalles que siempre nos ayudaron a mejorar. Tengo el placer de agradecerles a ustedes por la dedicación, el tiempo y la amistad que siempre quedará. A la gente del Call por esos kill regalados, Lachy, Lucifer, Crank (mi padre a veces hijo), aoki, yto, Armo, Red Lady y Red Baron (muy fácil). Muchas gracias a todos los mencionados, estos recuerdos son los que me llevo de todos y de la UCI.

Resumen

Las exposiciones de los trabajos de diploma se incluyen en las actividades docentes de mayor envergadura y como forma de evaluar la culminación de estudios en la Universidad de las Ciencias Informáticas (UCI). Los tribunales de defensa de trabajos de diploma son los encargados de que dicha evaluación cuente con la calidad requerida. El proceso de identificación de los miembros de los tribunales cuenta con una aplicación de Escritorio, la cual no puede ser accedida por varios usuarios al mismo tiempo, si esta no se tiene instalada en cada computadora. Además, no tiene en cuenta la calidad de conformación de los tribunales y los oponentes de las tesis. El objetivo que se persigue con la presente investigación es: Desarrollar un sistema informático, que permita conformar automáticamente los tribunales de defensa de trabajos de diploma en la Facultad 2. Para lograrlo, se hace uso de técnicas de inteligencia artificial, específicamente, métodos de optimización.

Se implementó una aplicación Web aplicando la técnica de Metaheurística Algoritmo Genético la cual genera una propuesta de tribunales, identificando los miembros asignadas a los mismos, atendiendo al Área de Conocimiento, nivel de experticia en el tema, Categoría Docente y Categoría Científica de los profesores. Para la implementación de la solución se utilizó el lenguaje de programación Python 2.7.6, Django 1.7.1 como framework de desarrollo, seleccionando como IDE PyCharm 4.0.4, metodología de desarrollo RUP, con UML como lenguaje de modelado y la notación de modelado de procesos del negocio BPMN.

PALABRAS CLAVE: *Algoritmos genéticos, asignación de recursos, Conformación automática de tribunales, inteligencia artificial, Metaheurística, optimización combinatoria, trabajos de diploma, Web.*

Tabla de Contenido

Introducción	1
Capítulo 1. Fundamentación Teórica del Sistema informático para la conformación automática de tribunales de defensa de trabajos de diploma en la Facultad 2.	5
1.1 Sistema Categorial Empleado	5
1.2 Problema de Optimización	6
1.3 Peculiaridades de las técnicas metaheurísticas	8
1.4 Composición y funcionamiento de los Algoritmos Genético	10
1.5 Análisis de las soluciones existentes	11
1.5.1 Sistema informático para la conformación automática de tribunales de tesis en la Facultad 2	12
1.5.2 Propuesta de técnicas evolutivas para la confección automática de tribunales de defensa de trabajos de diploma	13
1.6 Propuesta de Tecnologías, Herramientas y Metodologías de desarrollo de software	13
1.6.1 Lenguaje de modelado UML 2.0	14
1.6.2 Lenguajes de Programación	14
1.6.3 Entornos de Desarrollo Integrado	15
1.6.4 Framework y Librerías	15
1.6.5 Servidor Web	16
1.6.6 Metodología de Desarrollo RUP	16
1.6.7 Herramienta de Modelado Visual Paradigm 5.0	19
1.6.8 Notación de Modelado de Procesos de Negocio (BPMN)	19
Capítulo 2. Características del Sistema informático para la conformación automática de tribunales de defensa de trabajos de diploma en la Facultad 2.	20
2.1 Modelo de Negocio	20
2.1.1 Descripción del negocio	22
2.2 Requisitos Funcionales	22
2.3 Requisitos No Funcionales	25
2.4 Propuesta de Solución	26
2.5 Descripción del Sistema	27
2.6 Priorización de los Casos de Uso	30
2.7 Descripción de Casos de Uso Crítico	32
Capítulo 3. Diseño del Sistema informático para la conformación automática de tribunales de defensa de trabajos de diploma en la Facultad 2.	38

3.1	Patrón de Arquitectura	38
3.2	Patrones de Diseño	41
3.3	Modelo de Datos	42
3.4	Modelo de Optimización a utilizar	43
3.5	Algoritmo a Desarrollar	46
Capítulo 4. Implementación y Prueba del Sistema informático para la conformación automática de tribunales de defensa de tribunales de defensa de trabajos de diploma en la Facultad 2.		48
4.1	Diagrama de Despliegue del Sistema informático para la conformación automática de los tribunales de defensa de tribunales de defensa de trabajos de diploma de la Facultad 2.	48
4.2	Validación del sistema	49
4.2.1	Prueba del sistema propuesto	50
4.3	Diseño de pruebas	51
4.3.1	Resultado del modelo de prueba de caja negra	52
4.4	Prueba de aceptación del cliente	53
4.4.1	Resultados de las pruebas de aceptación del cliente	54
4.4.2	Resultados de las pruebas de carga	54
Conclusiones		56
Recomendaciones		57
Referencias Bibliográficas		58
Bibliografía		60
Anexos		63

Índice de Figuras

<i>Figura 1 Taxonomía de las Metaheurísticas</i>	<i>9</i>
<i>Figura 2 Representación de un Algoritmo Genético</i>	<i>11</i>
<i>Figura 3 Método de Boehm y Turner</i>	<i>17</i>
<i>Figura 4 Modelo de Negocio</i>	<i>21</i>
<i>Figura 5 Propuesta de Solución</i>	<i>26</i>
<i>Figura 6 Diagrama de Paquetes de Caso de Uso del Sistema</i>	<i>28</i>
<i>Figura 7 DCUS del Paquete Autenticación</i>	<i>28</i>
<i>Figura 8 DCUS del Paquete Exportar</i>	<i>29</i>
<i>Figura 9 DCUS del Paquete Gestionar Tesis</i>	<i>29</i>
<i>Figura 10 DCUS del Paquete Gestionar Profesores</i>	<i>29</i>
<i>Figura 11 DCUS del Paquete Gestionar Tribunales</i>	<i>30</i>
<i>Figura 12 DCUS del Paquete Gestionar Nomencladores</i>	<i>30</i>
<i>Figura 13 Modelo-Vista-Controlador</i>	<i>39</i>
<i>Figura 14 Representación del MVC en el sistema</i>	<i>39</i>
<i>Figura 15 Diagrama de Clases del Sistema del CU Generar Tribunales</i>	<i>40</i>
<i>Figura 16 Diagrama de Clases del Sistema del CU Gestionar Profesores</i>	<i>40</i>
<i>Figura 17 Modelo de Datos</i>	<i>42</i>
<i>Figura 18 Diagrama de Despliegue</i>	<i>48</i>
<i>Figura 19 Prueba de Caja Negra</i>	<i>50</i>
<i>Figura 20 Resultados del método de prueba de caja negra</i>	<i>53</i>

Índice de Tablas

<i>Tabla 1 Conceptos fundamentales de los Algoritmos Genéticos</i>	<i>10</i>
<i>Tabla 2 Personas que intervienen en el proceso de negocio</i>	<i>22</i>
<i>Tabla 3 Requisitos Funcionales.....</i>	<i>22</i>
<i>Tabla 4 Priorización de los Casos de Uso</i>	<i>31</i>
<i>Tabla 5 Descripción del Caso de Uso Generar Tribunales</i>	<i>32</i>
<i>Tabla 6 Descripción del Caso de Uso Gestionar Roles</i>	<i>33</i>
<i>Tabla 7 Descripción de los nodos y enlaces de comunicación</i>	<i>49</i>
<i>Tabla 8 Caso de Prueba Generar Tribunales</i>	<i>52</i>
<i>Tabla 9 Pruebas de Carga del sistema</i>	<i>54</i>
<i>Tabla 10 Descripción del Caso de Uso Autenticar Usuario</i>	<i>63</i>
<i>Tabla 11 Descripción del Caso de Uso Exportar a Excel</i>	<i>64</i>
<i>Tabla 12 Descripción del Caso de Uso Exportar a XML</i>	<i>65</i>

Introducción

Las universidades enfrentan en la actualidad el gran reto de ampliar su capacidad de respuesta a las exigencias de la sociedad, fundamentalmente a las crecientes demandas que afrontan los profesionales en formación. Como instituciones de creación, preservación y promoción de la cultura general, propician el debate acerca de la calidad de los procesos a través de los cuales se forman los profesionales. En busca de la eficiencia de sus resultados y eficacia en la consecución de los fines se proyectan, planifican, diseñan y estructuran los planes y programas de estudio en la Educación Superior.

La formación profesional constituye un espacio de construcción de valores y principios, entre los estudiantes y profesores, que implica el desarrollo humano progresivo. Es además, un proceso en el que estos desarrollan el compromiso social, profesional y la trascendencia en su contexto, toda vez que elevan su capacidad para la reflexión divergente y creativa, para la evaluación crítica y autocrítica, para la solución de problemas, la toma de decisiones y la adaptación flexible a un mundo cambiante. Alcanzar una integralidad en la formación profesional a nivel universitario implica, ante todo, formar a un individuo competente y comprometido con su labor y la sociedad en que se inserta; independientemente de la especificidad que impone cada profesión y sus contextos.

El Ministerio de Educación Superior (MES) plantea que existen dos vías para evaluar a los futuros profesionales, la defensa de trabajo de diploma y el examen estatal. En el Reglamento de Trabajo docente metodológico del MES se plantea que: *Así, el ejercicio de culminación de estudios se convierte en la actividad final del Proceso Docente Educativo (PDE), de acuerdo al sistema de preparación de pregrado en la Universidad de las Ciencias Informáticas. La defensa del trabajo de diploma, comprende un momento importante, no solo en el orden instructivo, sino también educativo. De la conformación y rigurosidad de los tribunales, depende el éxito de esta actividad.* (1)

La Universidad de las Ciencias Informáticas (UCI) tiene la misión de formar profesionales altamente calificados y comprometidos con la patria, se alcanza un elevado nivel en el desarrollo de aplicaciones y servicios informáticos contribuyendo a la informatización de la sociedad. En la UCI se emplea la defensa del trabajo de diploma como forma de evaluar la culminación de estudios.

El acto de defensa de tesis comprende que exista un tutor, autor o autores, un oponente, una aprobación de un perfil de tesis, una tarea de investigación y un tribunal que evaluará a los estudiantes en el acto de defensa de trabajo de diploma, mejorando con ello la calidad y exigencia que debe acompañar a este acto evaluativo de culminación de estudios. Las condiciones mínimas definidas, a partir de lo comprendido en el acto de defensa de tesis, que debe tener un tribunal son: el presidente y oponente deben tener el mayor conocimiento del tema de la tesis, en un tribunal no deben coincidir los oponentes y tutores con alguno de sus miembros y dicho tribunal consta de cuatro miembros: Presidente, Vocal, Secretario y Oponente.

Hasta la fecha, son asignados los temas de tesis que han sido previamente definidos por miembros del claustro. Atendiendo a que en el PDE de la UCI se contempla a los estudiantes como sujetos activos de su propio proceso de formación, estos últimos pueden proponer un tema de tesis, que debe ser evaluado en cuanto a su alcance y relación con las líneas de investigación establecidas por la Facultad. Después se redacta el perfil de tesis, que debe ser aprobado para poder comenzar el proceso de elaboración del trabajo de diploma. De ser aprobado, se le asigna un tutor al estudiante, en caso de que no se hubiese propuesto en el momento de redacción del perfil de tesis. Posteriormente se conforman los tribunales que aprobarán los perfiles y evaluarán el acto de defensa de trabajo de diploma.

El hecho de que la identificación y conformación de los miembros de los tribunales de defensa de trabajos de diploma se realice manualmente, carga de implicaciones negativas el proceso, introduciendo: complejidad teniendo en cuenta que es necesario verificar que los miembros del tribunal no coincidan con los oponentes y tutores, elevado consumo de tiempo al efectuar la tarea, generación involuntaria de errores e imprecisiones; por lo que los directivos del PDE deben prestar la máxima atención.

Desde el 2012 existe la investigación: Sistema informático para la conformación automática de tribunales de tesis en la Facultad 2, la cual propone una solución a dicha problemática, sin embargo, no se realiza la asignación automática de los oponentes a los trabajos de diploma, por lo que en ocasiones, uno de los miembros del tribunal puede coincidir con el oponente de una de las tesis asignadas. La aplicación está desarrollada para ambientes de Escritorio, lo que requiere de su instalación y actualización en cada cliente y solamente funciona para el sistema operativo WindowsXP o superior. La solución se modela como un problema de optimización combinatoria monobjetivo, minimizando la diferencia entre las evaluaciones de

integralidad de los tribunales sin tener en cuenta la calidad de los mismos y no permite exportar en los formatos de Excel y XML.

Atendiendo a todo lo anteriormente especificado se identifica como **problema a resolver**: ¿Cómo mejorar el proceso de conformación de los tribunales de defensa de trabajos de diploma en la Facultad 2? En este sentido el **objetivo general** que se propone alcanzar es: Desarrollar un sistema informático que permita conformar automáticamente los tribunales de defensa de trabajos de diploma en la Facultad 2, cumpliendo con el conjunto de requisitos mínimos exigidos. Se define como **objeto de estudio**: Los sistemas informáticos para la asignación de recursos; atendiendo al objeto declarado se ha definido como **campo de acción**: La conformación automática de tribunales de defensa de trabajos de diploma.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

- Determinación de los referentes teóricos acerca de los sistemas de distribución automática de recursos.
- Selección de las herramientas y técnicas de inteligencia artificial a utilizar para el desarrollo del sistema.
- Modelación del problema de optimización combinatoria.
- Captura de los requisitos funcionales y no funcionales para identificar las características que tendrá el sistema.
- Identificación de los elementos del diseño y la implementación para diseñar e implementar la solución.
- Elaboración y documentación del proceso de pruebas para lograr que el sistema quede libre de errores.

Como **métodos a emplear**, se pueden citar los métodos del nivel teórico: Analítico-Sintético; Inductivo-Deductivo y el Análisis Documental, sustentados en el estudio de disímiles fuentes de información y en el procesamiento de los fundamentos científicos y de las apreciaciones de numerosos autores y colectivos de autores. Además se utilizó el método del nivel empírico: Observación Científica, que permite constatar la situación problemática existentes, así como validar el correcto funcionamiento del sistema propuesto.

La investigación se encuentra estructurada en cuatro capítulos, en el **Capítulo 1 Fundamentación Teórica** donde se realiza un estudio relacionado con los principales conceptos y características del proceso de conformación de tribunales de defensa de

trabajos de diploma. Además se realiza un análisis de las principales herramientas utilizadas con dicho fin. Se explican las herramientas y metodologías seleccionadas para realizar la implementación del sistema. Luego el **Capítulo 2 Análisis de la solución** en el que se describen las características de la solución propuesta, definidas a partir de la modelación de los procesos de negocio y la especificación de los requisitos de software. Concluyendo el **Capítulo 3 Diseño del Sistema** donde se especifican los elementos del diseño de la arquitectura del sistema y los patrones de diseño utilizados. Además se describen los elementos del modelo de optimización y del algoritmo utilizado para la solución. Como resultado el **Capítulo 4 Implementación y Prueba** en el cual se detalla el diagrama de despliegue y las pruebas realizadas al sistema.

Capítulo 1. Fundamentación Teórica del Sistema informático para la conformación automática de tribunales de defensa de trabajos de diploma en la Facultad 2.

En el presente capítulo se analizan los aspectos relacionados con el sistema categorial empleado, el problema de optimización y los sistemas que existen de asignación automática de recursos. Además de las características fundamentales de las herramientas y tecnologías de software utilizadas para el desarrollo del sistema.

1.1 Sistema Categorial Empleado

Al tomar como punto de partida el enfoque holístico de la enseñanza resulta válido referirnos al Proceso Docente Educativo como algo más que la integración de la enseñanza y el aprendizaje, más que cada uno de sus componentes, es la integración holística y sistémica de todos ellos junto con las cualidades, niveles de asimilación, de profundidad y estructural, en sus tres dimensiones: educativa, instructiva y desarrolladora. (2)

El PDE parte de una institución docente y se proyecta en la sociedad, con el encargo de educar al hombre para la vida a partir de compromisos sociales, debiendo ser capaz de enfrentarse a nuevas situaciones y problemas que se le presenten y resolverlos en pos de transformar la sociedad. (2)

El Trabajo de Diploma no es más que el tipo de trabajo investigativo de los estudiantes que les permite adquirir un mayor dominio y actualización de los métodos científicos y técnicas característicos de la profesión. La defensa del trabajo de diploma es un tipo de evaluación de la culminación de los estudios cuyo objetivo es comprobar el grado de dominio de los estudiantes de los objetivos generales de la carrera, mediante la solución, con independencia y creatividad de un problema propio de la profesión, utilizando la metodología de la investigación científica. (1)

El tribunal evalúa los trabajos de diploma, el cual, para emitir una calificación tendrá en cuenta los siguientes elementos:

- Calidad del trabajo (uso de la metodología de la investigación científica, actualización científico técnica, uso de las estrategias curriculares de acuerdo con el contenido del trabajo, entre otros).
- Capacidad creadora, originalidad e independencia en el desarrollo del trabajo.

- Calidad de la exposición durante la defensa, respuestas a las preguntas y dominio del tema.
- Opiniones del profesor designado para la orientación científica del estudiante, del oponente y de la entidad laboral para la cual se realizó el trabajo. (1)

1.2 Problema de Optimización

Los problemas de optimización permiten seleccionar el mejor elemento (con respecto a algún criterio) de un conjunto de elementos disponibles. Tienen como objetivo maximizar o minimizar una función real, eligiendo valores de entrada y computando el valor de la función descubriendo los mejores valores de una función objetivo. (3)

Un problema de optimización se puede describir matemáticamente como:

Dado un dominio X y una función objetivo

$$f(x):x \in X \rightarrow \mathbb{R}$$

El objetivo es encontrar x' que verifique (en caso de maximizar)

$$x' \in X: f(x') \geq f(x), \forall x \in X$$

Atendiendo al dominio de las variables, los problemas de optimización pueden dividirse en dos categorías: de Optimización Continua y Optimización Discreta. En el caso específico del problema a resolver en esta investigación, este debe tratarse como de Optimización Discreta o Combinatoria, pues la naturaleza de sus variables así lo exige.

Un problema de optimización combinatoria tiene como objetivo encontrar el máximo (o el mínimo) de una determinada función sobre un conjunto finito de soluciones (CS). Cabe destacar que dada la finitud de CS, las variables han de ser discretas, restringiendo su dominio a una serie finita de valores. Habitualmente, el número de elementos de CS es muy elevado, haciendo impracticable la evaluación de todas sus soluciones para determinar el óptimo. (4)

La mayoría de los problemas de Optimización Combinatoria pueden ser modelados como problemas de programación lineal entera. La programación entera trata los problemas de maximizar o minimizar una función de diversas variables, sujeta a condiciones de igualdad y/o desigualdad, restringiéndose todas o alguna de esas variables a tomar valores enteros. En general, se usa en todas aquellas áreas donde se trata de resolver el problema de asignar recursos de cualquier tipo, solo disponibles en cantidades discretas. (5)

Nótese como el argumento esgrimido anteriormente se relaciona con la temática de esta investigación, consistente en un problema de asignación de recursos, en este caso, de recursos humanos, en interés de un Proceso Docente Educativo de excelencia.

Los problemas de optimización combinatoria son una rama de la optimización en matemáticas aplicadas y en ciencias de la computación, relacionada con la inteligencia artificial. La Inteligencia Artificial implica el uso de métodos basados en el comportamiento inteligente de los seres humanos y de otros animales, para resolver problemas complejos. (6)

Algunos ejemplos de problemas de optimización combinatoria son:

- El viajante de comercio: Un viajante de comercio ha de visitar n ciudades, comenzando y finalizando en su propia ciudad. Conociendo el coste de ir de cada ciudad a otra, determinar el recorrido de coste mínimo.
- La Mochila: Dados n elementos e_1, e_2, \dots, e_n , con pesos p_1, p_2, \dots, p_n y beneficios b_1, b_2, \dots, b_n , y dada una mochila capaz de albergar hasta un máximo de peso M (capacidad de la mochila), se desea encontrar las proporciones de los n elementos x_1, x_2, \dots, x_n ($0 \leq x_i \leq 1$) que se tienen que introducir en la mochila de forma que la suma de los beneficios de los elementos escogidos sea máxima.
- Asignación de Recursos (Tareas): Se supone que se dispone de n trabajadores y n tareas. Sea $b_{ij} > 0$ el coste de asignarle el trabajo j al trabajador i . Una asignación de tareas puede ser expresada como una asignación de los valores 0 o 1 a las variables x_{ij} , donde $x_{ij} = 0$ significa que al trabajador i no le han asignado la tarea j , y $x_{ij} = 1$ indica que sí. Una asignación válida es aquella en la que a cada trabajador solo le corresponde una tarea y cada tarea está asignada a un trabajador. Dada una asignación válida, se define el coste de dicha asignación como:

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} b_{ij}$$

Se dice que una asignación es óptima si es de mínimo coste.

Para la conformación de un tribunal de tesis, se debe asignar un profesor para cada rol, de ahí que el problema a resolver sea similar al de asignación de recursos enunciado anteriormente, en este caso serían recursos humanos.

Para resolver los problemas de optimización combinatoria es posible el empleo de métodos exactos y de métodos aproximados. Si el problema de optimización tiene un espacio de soluciones muy grande, el coste de ejecución de los métodos exactos puede aumentar de forma exponencial, convirtiendo la resolución en prácticamente inviable por lo que no se utiliza este método para dar respuesta al problema. Debido a lo anterior se utilizarán métodos aproximados mediante el uso de técnicas de inteligencia artificial, como son las heurísticas y las metaheurísticas.

Las técnicas heurísticas son procedimientos para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución. (7)

Las técnicas Metaheurísticas “son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Los Metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos”. (8)

Conociendo entonces que las técnicas heurísticas son más específicas, o sea, dependen del problema que se desea resolver, mientras que las metaheurísticas son más generales y flexibles, se decide optar por el uso de estas últimas para darle solución al problema de optimización combinatoria, cuyo modelado representa el problema a resolver de la investigación.

1.3 Peculiaridades de las técnicas metaheurísticas

La Figura 1 muestra una clasificación de las metaheurísticas

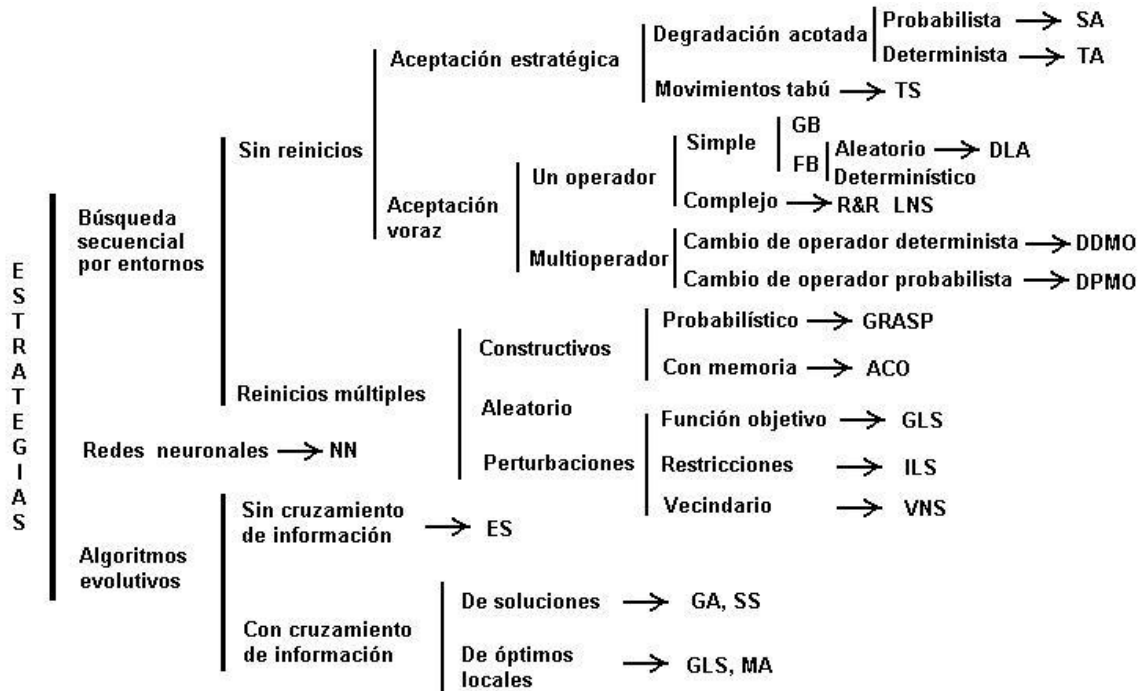


Figura 1 Taxonomía de las Metaheurísticas

Las metaheurísticas de búsqueda guían los procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explotar las estructuras de entornos asociadas.

Los algoritmos evolutivos o genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos (AG) son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas. (9)

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Si el Algoritmo Genético ha sido bien diseñado, la población convergerá hacia una solución óptima del problema. Además, la conformación de tribunales está sujeta a un conocimiento local, en el cual un tribunal específico debe cumplir con ciertas condiciones, y a la vez está asociada con un conocimiento global que implica que debe

existir cierto equilibrio entre la calidad de los tribunales, por lo que resulta necesario un proceso de búsqueda de posibles mejoras.

Se selecciona un algoritmo genético debido a que algoritmos intrínsecamente paralelos, realizan la búsqueda de la solución de forma simultánea con varios individuos. Son los algoritmos que hacen una barrida mayor del subespacio de posibles soluciones válidas y son los más exploratorios disponibles para los problemas de optimización. (10)

1.4 Composición y funcionamiento de los Algoritmos Genético

Los algoritmos genéticos emplean de la biología varios términos para ejecutar las acciones que realizan. (11) En la siguiente tabla se resumen los conceptos utilizados, su respectiva interpretación en los algoritmos genéticos y cómo está asociado cada término de la solución planteada en el Capítulo 3.

Tabla 1 Conceptos fundamentales de los Algoritmos Genéticos

Término	Descripción
Población	Conjunto de individuos o cromosomas. Equivale a un conjunto de soluciones alternativas (Conjunto de soluciones).
Cromosoma	Codificación de un individuo solución en una estructura manipulable computacionalmente (Una solución del conjunto de soluciones).
Gen	Subestructura de un cromosoma (Tribunal).
Alelos	Conjunto de posibles valores que puede tener la subestructura gen (Roles).
Desempeño	Valor resultante del proceso de evaluación de un cromosoma que incide en el proceso de selección (Costo).
Selección	Mecanismo mediante el cual se eligen determinados cromosomas que formarán parte de la población que se cruzará (ver en Capítulo 3).
Cruzamiento	Intercambio de genes entre dos estructuras cromosoma llamadas padres, que originan una estructura nueva llamada hijo (ver en Capítulo 3).
Mutación	Cambio aleatorio de alelos dentro de una subestructura gen (ver en Capítulo 3).

Los genes son la unidad fundamental de los algoritmos genéticos y la composición de varios genes es conocida como cromosoma, que representan las posibles soluciones de un problema. A cada miembro de la población se le aplica una función de aptitud, con el propósito de saber qué tan buena es la solución que se está codificando.

Posteriormente, se procede a hacer la selección de los cromosomas que van a cruzarse en la próxima generación, eligiendo una técnica de selección. El cruzamiento es el principal operador genético y para su realización es necesario seleccionar el operador de cruzamiento que va a operar sobre los cromosomas. En el proceso de cruzamiento se combinan las características de los cromosomas padres. También existe un operador de mutación que cambiará de manera aleatoria los alelos de un cromosoma. Para realizar el proceso, es necesario conocer la probabilidad de mutación, la cual indica los individuos que pueden mutar. El proceso de selección para determinar que alelo del gen será cambiado se realiza aleatoriamente. (11)

La figura 2 muestra de forma detallada el funcionamiento de un algoritmo genético. Los parámetros a seleccionar en los algoritmos genéticos se ajustan según las características del problema a resolver.

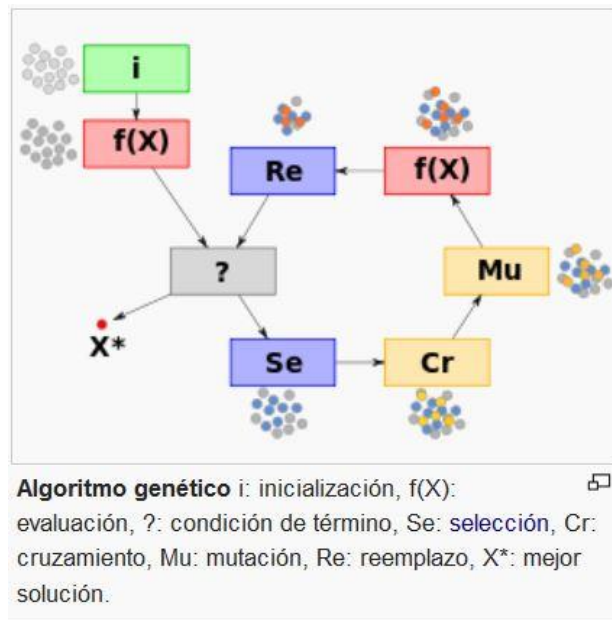


Figura 2 Representación de un Algoritmo Genético

1.5 Análisis de las soluciones existentes

La esencia de la problemática planteada en el diseño de esta investigación trata sobre un problema de asignación de recursos, con una cantidad elevada de variables dependientes, imposibles de ser tenidas en cuenta en su totalidad por los directivos de un proceso tan complejo como lo es el Proceso Docente Educativo.

Existen varios sistemas que utilizan el problema de asignación de recursos para su solución. Al realizar una búsqueda en distintas fuentes bibliográficas sobre aplicaciones

que asignaran recursos automáticamente, se encontraron algunas soluciones que se enfocan en problemas semejantes, pero que no llegan a tratar concretamente una solución al problema que se ha planteado en este diseño de investigación. A continuación se describen los sistemas encontrados:

1.5.1 Sistema informático para la conformación automática de tribunales de tesis en la Facultad 2

Sistema informático que permite conformar automáticamente los tribunales de defensa de trabajos de diploma en la Facultad 2. Para lograrlo, se hace uso de técnicas de inteligencia artificial, específicamente, de la metaheurística GRASP (del inglés Greedy Randomized Adaptative Search Procedure). La aplicación en cuestión está desarrollada para entornos de Escritorio, debido a esto requiere su instalación y actualización en cada cliente y solamente funciona para el sistema operativo WindowsXP o superior. Además se modela como un problema de optimización monobjetivo, minimizando la diferencia entre las evaluaciones de integralidad de los tribunales sin tener en cuenta la calidad de los mismos.

No permite generar automáticamente los oponentes de los tribunales. No permite exportar los tribunales obtenidos. Una vez obtenidos los tribunales, los mismos no se pueden modificar, lo que provoca que no se puedan realizar cambios en los mismos en caso de una situación excepcional. La solución de optimización de este problema es la base que se utilizará en la solución del problema planteado, pues permite generar soluciones aleatorias teniendo en cuenta los mismos criterios. Además, propició la información que brinda la evaluación de los tribunales para calcular los grados del conocimiento y los científicos. También aportó un soporte al levantamiento de requisitos.

Uno de los problemas encontrados en la solución: Sistema informático para la conformación automática de tribunales de tesis en la Facultad 2 es que modelan el problema de optimización combinatoria como una solución monobjetivo teniendo solamente en cuenta minimizar las evaluaciones de integralidad de los tribunales. Esto provoca que no se tenga en cuenta otros criterios como la disponibilidad de los profesores y los grados científicos de los mismos.

A continuación se detallará el estudio de una solución encontrada que permite la modelación al problema de optimización con varias variables.

1.5.2 Propuesta de técnicas evolutivas para la confección automática de tribunales de defensa de trabajos de diploma

Este sistema informático desarrollado en la universidad de Holguín se modela como un problema de optimización combinatoria multicriterio que puede ser resuelto por métodos clásicos. El trabajo propone un algoritmo evolutivo simple que resuelve de manera eficiente este problema. Permite confeccionar tribunales de trabajo de diploma de forma automática. Puede ser modelada como un problema de optimización combinatorio multicriterio, el cual puede ser definido informalmente de manera siguiente: encontrar las mejores asignaciones (tribunales) de cuatro profesionales (presidente, secretario, vocal, y oponente) para t tribunales de defensa de trabajos de diploma, a partir n profesionales, cumpliendo con los siguientes criterios: el uso del profesional debe estar en correspondencia con su disponibilidad, el conocimiento del tribunal en su conjunto tiene que ser el máximo posible sobre el tema del trabajo de diploma, y por último la distribución de profesionales con grado científico debe ser lo más uniforme posible, en particular de los doctores. (12)

Utiliza un algoritmo genético para obtener la solución. El sistema resuelve el problema planteado pero no tiene en cuenta la calidad de los tribunales, debido a que trabajan con construcción de equipos. Además no se tiene acceso al código fuente. Este sistema facilitó la selección de los Algoritmos Genéticos como técnica de metaheurística empleada y aportó la función definida para la disponibilidad de los profesores en la Función Objetivo.

Además se estudiaron otras dos soluciones: el sistema automático para asignación de aulas y distribución de y la solución a un problema de asignación de horarios de mantenimiento a máquinas que los dos aportan solución con un problema multiobjetivo, pero no brindan nuevos elementos a la solución.

1.6 Propuesta de Tecnologías, Herramientas y Metodologías de desarrollo de software

Las metodologías de desarrollo de *software* abarcan todo el ciclo de vida del *software*. (13) En la construcción del sistema es necesaria la selección de una metodología de desarrollo que documente detalladamente el proceso de desarrollo. La selección de las tecnologías y herramientas es fundamental en la modelación e implementación del sistema propuesto que a continuación se detallan:

1.6.1 Lenguaje de modelado UML 2.0

Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML por sus siglas en inglés), es un lenguaje de modelado para visualizar, especificar, construir y documentar un sistema. Ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Los lenguajes de modelado son usados en combinación con las metodologías de desarrollo de *software* para avanzar de una descripción inicial a un plan de implementación y comunicar dicho plan a los desarrolladores. (14)

1.6.2 Lenguajes de Programación

Se selecciona Python en su versión 2.7.6 como lenguaje de programación para la construcción del software. A continuación se especifican las características que permitieron su selección.

Python 2.7.6

Es administrado por la *Python Software Foundation* con una licencia de código abierto. Python fue diseñado para ser leído con facilidad y una de sus características es el uso de palabras donde otros lenguajes utilizarían símbolos. Es un lenguaje preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- Es un lenguaje multiplataforma.
- Es gratuito, incluso para propósitos empresariales. (15)
- Permite la incorporación de librerías que facilitan la implementación del algoritmo genético propuesto, las cuales se identifican más adelante.

1.6.3 Entornos de Desarrollo Integrado

Se empleará como IDE (Integrated Development Enviroment): PyCharm, sobre el cual a continuación se hace referencia para el desarrollo de la aplicación.

PyCharm 4.0.4

Es un IDE de Python con un completo juego de herramientas para el desarrollo productivo con el lenguaje de programación Python. Adicionalmente, el IDE provee capacidades de alto rango para desarrolladores profesionales de Web con el framework Django. (16)

Es multiplataforma, proporciona análisis de código y depuración gráfica. PyCharm es desarrollado por la empresa JetBrains y debido a la naturaleza de sus licencias tiene dos versiones, la Community que es gratuita y orientada a la educación y al desarrollo puro en Python y la Professional. (17)

Dentro de las ventajas de PyCharm encontramos:

- Autocompletado, resaltador de sintaxis, herramienta de análisis y refactorización.
- Integración con frameworks web como: Django, Flask, Pyramid y Web2Py.
- Frameworks javascripts: jQuery y AngularJS.
- Debugger avanzado de Python y Javascript.
- Integración con lenguajes de plantillas: Django Template.
- Soporta entornos virtuales e intérpretes de Python 2.x.
- Compatibilidad con SQLAlchemy (ORM), Google App Engine y Cython.
- Sistemas de control de versiones: Git, CVS y Mercurial. (18)

1.6.4 Framework y Librerías

Se empleará como framework de desarrollo Django en su versión 1.7.1 y librerías para facilitar y agilizar el trabajo.

Django 1.7.1

Es un framework web de código abierto. Fue inicialmente desarrollado para gestionar aplicaciones web de páginas orientadas a noticias de World Online, más tarde se liberó bajo licencia BSD. Proporciona una serie de herramientas para facilitar la creación de páginas, siguiendo los principios DRY (Don't Repeat Yourself - No Te Repitas) para evitar duplicidad en las líneas de código e invertir el menor esfuerzo posible. Por ejemplo, levantar un panel de administración básico sólo requiere un par de líneas de

Python. También se adscribe al diseño MVC (Modelo-Vista-Controlador), por lo que las diferentes partes del sitio están claramente separadas. Por ejemplo, el código de acceso a los datos es completamente independiente al que gobierna el aspecto externo de la página. (19)

Para agilizar el diseño y la obtención de información se utilizaron las siguientes librerías:

- Python-ldap: permite a las aplicaciones que usan este lenguaje acceder a los servicios LDAP, en este caso para obtener información de los usuarios que utilicen el sistema como usuario, nombre, verificar o actualizar la contraseña de los mismos, etc.
- Django-auth-ldap: es la librería que le permite a Django usar las funcionalidades de python-ldap.
- Django-crispy-forms: permite aplicarle clases css a los elementos de los formularios buscando un aspecto más agradable a la vista del usuario.

1.6.5 Servidor Web

Se empleará como servidor web Apache en su versión 2.2.22-1ubuntu1.8 para ejecutar la aplicación y se pueda ser accedida por los clientes.

Apache 2.2.22-1ubuntu1.8

El servidor Apache HTTP, también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento. Para entender mejor lo que es Apache, primeramente se debe precisar lo que es un servidor web. La definición más sencilla de servidor web, que es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc.). Estos servidores web utilizan el protocolo http. Los servidores web están alojados en un ordenador que cuenta con conexión a Internet. El servidor web, se encuentra a la espera de que algún navegador le haga alguna petición, como por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red. (20)

1.6.6 Metodología de Desarrollo RUP

En el presente trabajo, se utiliza el método de Boehm y Turner a partir del análisis de sus cinco criterios, se realiza la selección más acertada del enfoque, la metodología de

desarrollo RUP y las prácticas a utilizar en el proceso de desarrollo de software, quedando de la siguiente forma:

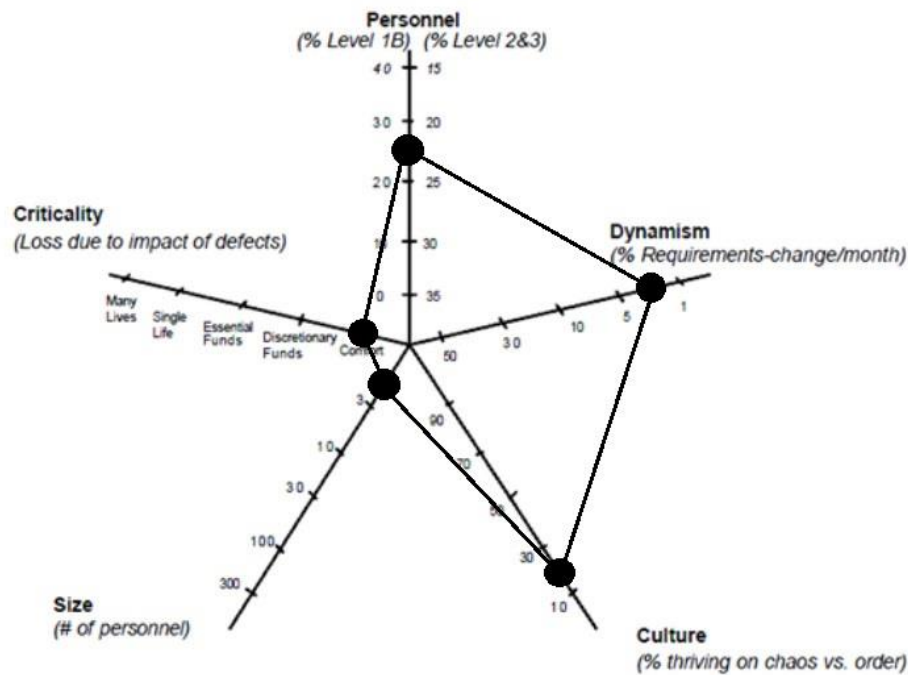


Figura 3 Método de Boehm y Turner

El Proceso Unificado de Rational (RUP, por sus siglas en inglés), es el conjunto de procesos de ingeniería de software que dan guía para conducir las actividades de desarrollo del equipo. Como una plataforma de procesos que abarca todas las prácticas de la industria, RUP permite seleccionar fácilmente el conjunto de componentes de proceso que se ajustan a las necesidades específicas del proyecto. Está pensado para adaptarse a cualquier proceso y no solo al de software. (21)

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones. RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline (Línea Base) de la arquitectura. Durante la fase de inicio las iteraciones hacen mayor énfasis en

actividades de modelado del negocio y de requisitos. En la fase de elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requisitos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura. En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se seleccionan algunos Casos de Uso, se refinan su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan iteraciones hasta que se termine la implementación de la nueva versión del producto. En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios. (22)

RUP genera una amplia documentación del desarrollo de software y contribuye a la organización del equipo de trabajo. Para la solución en cuestión es necesario dejar una documentación detallada del sistema. La misma permitirá a la vicedecana o a las personas que trabajarán con el software entender la aplicación, poder realizar cambios en la misma y dejarlos documentados. El equipo de desarrollo no posee experiencia con metodologías ágiles.

La ingeniería de software, bajo restricciones de tiempo, costo y calidad, trata sobre la aplicación de prácticas y métodos para construir productos de software que cumplan las expectativas de clientes y usuarios. En ocasiones, la mala selección de los métodos no permite obtener los resultados esperados en los proyectos de desarrollo de software. Sin embargo en la actualidad, ya se cuenta con técnicas que teniendo en cuenta las características de estos proyectos permiten realizar una selección más acertada del método de desarrollo a utilizar. Sistema Gestor de Base de Datos PostgreSQL 9.2

Se decide emplear PostgreSQL debido a que es un potente sistema de bases de datos objeto-relacional de código abierto. Está distribuido bajo licencia BSD (Berkeley Software Distribution) y con su código fuente está disponible libremente. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada, que se ha ganado una sólida reputación de fiabilidad e integridad de datos. Funciona en los principales sistemas operativos, incluyendo Linux, UNIX, MacOS y Windows. Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados en varios idiomas. Soporta almacenamiento de objetos binarios grandes,

como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para Java, C / C + +, .NET, Perl, Python, entre otros, y una amplia documentación. (23)

1.6.7 Herramienta de Modelado Visual Paradigm 5.0

Visual Paradigm for UML (VP-UML) es una herramienta de diseño UML y herramienta CASE del inglés (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) diseñada para ayudar al desarrollo de software. Soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML), y la notación de Modelado de Procesos de Negocio (BPMN). Ofrece un completo conjunto de herramientas de equipos de desarrollo de software necesario para la captura de requisitos, planificación de software, la planificación de controles, el modelado de clases y modelado de datos. Además, permite modelar todos los tipos de diagramas de clases, código inverso y generar código desde los diagramas. (24)

1.6.8 Notación de Modelado de Procesos de Negocio (BPMN)

La notación de modelado de procesos de negocio empleada ha sido Business Process Modeling Notation (BPMN, por sus siglas en inglés). Proporciona la comprensión de los procesos de negocio en una notación gráfica, permitiendo la comunicación de estos procedimientos de manera estándar. Esta notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. (25)

Conclusiones del Capítulo

En el capítulo se definen, a través de conceptos, las características que intervienen en el proceso de conformación de los tribunales de defensa de trabajos de diploma para el acto de defensa de esta actividad en la Facultad 2, permitiendo clasificarlo como un problema de asignación de recursos. Además, se realizó un análisis del proceso de conformación de tribunales de defensa de trabajos de diploma, permitiendo identificar la necesidad inmediata de la automatización del proceso. También fueron analizadas varias técnicas para dar solución a la situación planteada, basada en un problema de optimización combinatoria, escogiendo la técnica metaheurística Algoritmo Genético. Fueron seleccionados en el proceso de construcción del sistema como metodología de desarrollo RUP, como herramienta Visual Paradigm y PostgreSQL como sistema gestor de base de datos y las tecnologías Apache como servidor web, PyCharm como entorno de desarrollo para el lenguaje Python y Django como como framework de desarrollo.

Capítulo 2. Características del Sistema informático para la conformación automática de tribunales de defensa de trabajos de diploma en la Facultad 2.

El presente capítulo aborda temas relacionados con la modelación del proceso de negocio y características del sistema informático. Se definen los requisitos funcionales y los no funcionales que debe cumplir la aplicación. Se describe además la propuesta de solución, mostrando los diagramas de casos de uso del sistema, su priorización y la descripción de los mismos.

2.1 Modelo de Negocio

Un modelo de negocio es un documento compuesto de un conjunto de submodelos; cada uno describe uno o más elementos organizacionales mediante diagramas UML con la Notación de Modelado de Procesos de Negocios (BPMN), y estos submodelos constan de un conjunto de diagramas UML 2.0, UML Business y BPMN. (26)

Para modelar los procesos de negocio identificados se utilizará la notación BPMN donde se identifican los procesos para la conformación de los tribunales de trabajos de diploma. En la figura 4 se muestra el Diagrama de procesos de negocio.

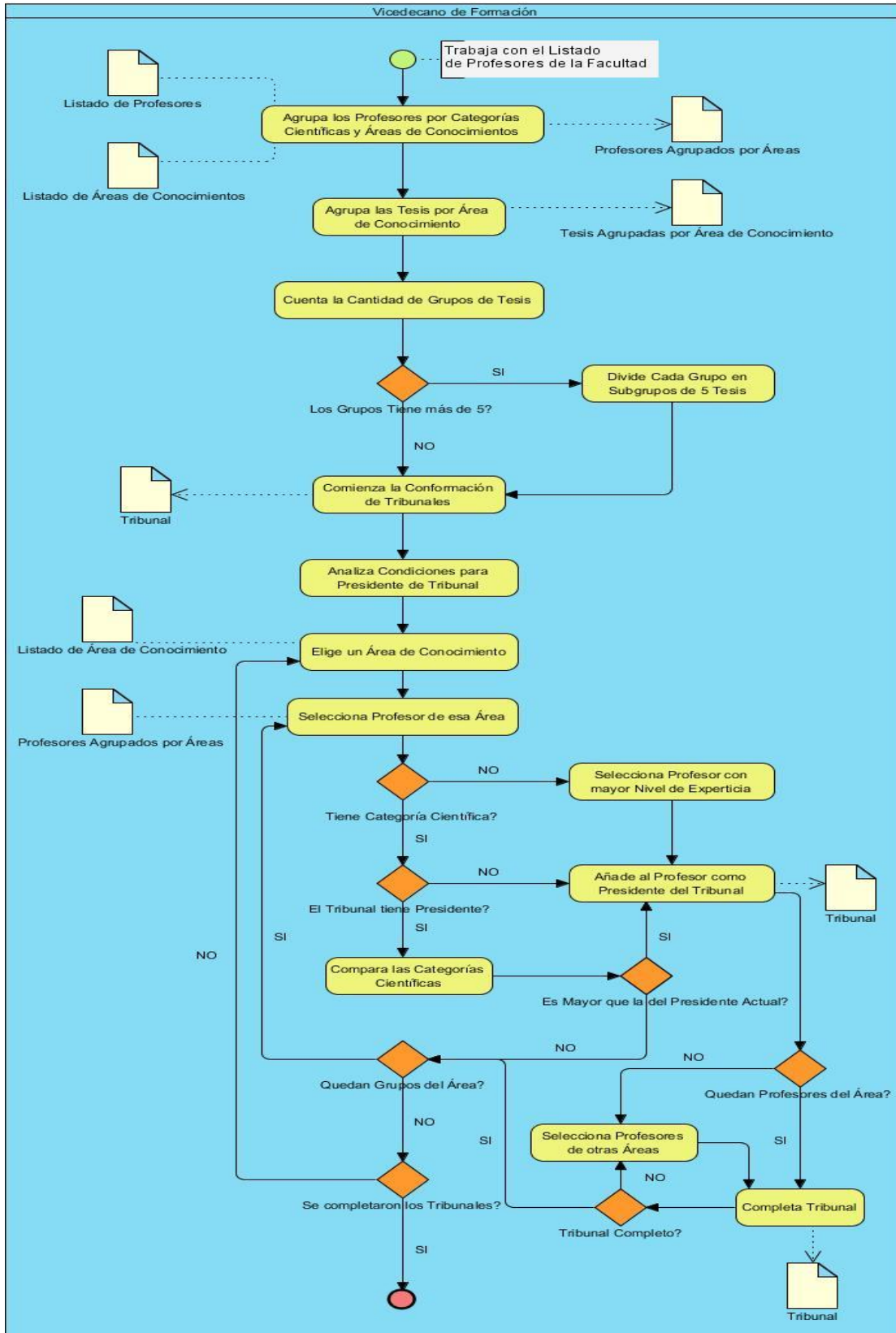


Figura 4 Modelo de Negocio

2.1.1 Descripción del negocio

La asignación de profesores para los tribunales de defensa de trabajos de diploma comienza cuando el Vicedecano de Formación agrupa a los profesores por Áreas de conocimiento ordenados por categoría científica. Con posterioridad, agrupa las tesis por área de conocimiento y las divide en grupos, para comenzar con la conformación de tribunales. Elige un área de conocimiento y selecciona un profesor para comprobar sus condiciones para Presidente de tribunal; si tiene categoría científica y no se ha asignado presidente a ese tribunal se asigna el profesor. Si ya tiene Presidente, se compara la categoría científica, si la del profesor seleccionado es mayor que la del Presidente, se intercambian los roles, si no, se verifica que existan más grupos de esa área y se comienza la asignación de Presidente nuevamente. Si quedan profesores del área escogida, se completan los tribunales. Si no se pueden completar, se añaden profesores de otras áreas. Si el área de conocimiento está completa, se comienza el proceso desde la selección de área, (se sigue este mismo criterio de selección para el oponente). Finaliza cuando están constituidos todos los tribunales.

Las personas que intervienen en el proceso de negocio son:

Tabla 2 Personas que intervienen en el proceso de negocio

Nombre	Justificación
Vicedecano de Formación	Es el encargado de realizar todo el proceso de gestión de tesis, profesores, y de conformar los tribunales de defensa de trabajos de diploma.

2.2 Requisitos Funcionales

Los requisitos funcionales para la realización del sistema posibilitan definir el alcance del sistema, teniendo en cuenta las acciones a realizar y el intercambio de datos entre sus diferentes funciones. Para la aplicación que se implementará se capturaron los siguientes requerimientos funcionales con su descripción, complejidad de desarrollo y peoridad para el cliente:

Tabla 3 Requisitos Funcionales

Nº. RF	Nombre	Descripción	Complejidad/Prioridad- Cliente
-----------	--------	-------------	--------------------------------

1	Autenticar Usuario	Se añade el usuario a la base de datos.	Baja/Media
2	Listar Profesores	Se listan todos los profesores del claustro que existen en la base de datos.	Media/Alta
3	Buscar Profesores	Permite buscar un profesor del claustro en la base de datos.	Media/Alta
4	Listar Tesis	Se listan todas las tesis que existen en la base de datos.	Media/Alta
5	Buscar Tesis	Permite buscar una tesis en la base de datos.	Media/Alta
6	Listar Tribunales	Lista automáticamente los tribunales confeccionados.	Alta/Alta
7	Buscar Tribunales	Permite buscar uno o varios tribunales en la base de datos siguiendo criterios de selección.	Alta/Alta
8	Generar Tribunales	Permite crear automáticamente los tribunales y añadirlos a la base de datos.	Alta/Alta
9	Gestionar Categorías Científicas <ul style="list-style-type: none"> • Añadir Categoría Científica • Eliminar Categoría Científica • Modificar Categoría Científica 	Se gestionan las categorías científicas.	Media/Alta
10	Gestionar Categoría Docente <ul style="list-style-type: none"> • Añadir Categoría Docente • Eliminar Categoría Docente • Modificar Categoría Docente 	Se gestionan las categorías docentes.	Media/Alta
11	Gestionar Niveles de Experticia	Se gestionan los niveles de experticia.	Media/Alta

	<ul style="list-style-type: none"> • Añadir Nivel de Experticia • Eliminar Nivel de Experticia • Modificar Nivel de Experticia 		
12	<p>Gestionar Profesores</p> <ul style="list-style-type: none"> • Añadir Profesor • Eliminar Profesor • Modificar Profesor 	Se gestionan los profesores existentes en la base de datos.	Media/Alta
13	<p>Gestionar Tribunales</p> <ul style="list-style-type: none"> • Añadir Tribunales • Eliminar Tribunales • Modificar Tribunales 	Se gestionan los tribunales existentes en la base de datos.	Media/Alta
14	<p>Gestionar Rol</p> <ul style="list-style-type: none"> • Añadir Rol • Eliminar Rol • Modificar Rol 	Se gestiona el rol de los profesores como miembros de los tribunales.	Media/Alta
15	<p>Gestionar Tesis</p> <ul style="list-style-type: none"> • Añadir Tesis • Eliminar Tesis • Modificar Tesis 	Se gestionan las tesis existentes en la base de datos.	Media/Alta
16	<p>Gestionar Áreas de Trabajo</p> <ul style="list-style-type: none"> • Añadir Área de Trabajo • Eliminar Área de Trabajo • Modificar Área de Trabajo 	Se gestionan las Áreas de Trabajo en la base de datos.	Media/Alta
17	<p>Gestionar Áreas de Conocimientos</p> <ul style="list-style-type: none"> • Añadir Área de Conocimiento 	Se gestionan las Áreas de Conocimientos en la base de datos.	Media/Alta

	<ul style="list-style-type: none"> • Eliminar Área de Conocimiento • Modificar Área de Conocimiento 		
18	Exportar a Excel	Exporta a Excel todos los listados que se muestren.	Media/Alta
19	Exportar a XML	Exporta a XML todos los listados que se muestren.	Media/Alta

2.3 Requisitos No Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Son las características que hacen al producto atractivo, usable, rápido y confiable. Para la aplicación que se desea desarrollar se identificaron los siguientes requerimientos no funcionales:

Hardware

- Microprocesador Pentium 4 o superior.
- Memoria RAM \geq 1GB.

Software

- Debe estar instalada en la computadora del cliente Python en su versión 2.7.6 y el Framework de desarrollo Web Django 1.7.1.
- El servidor de base de datos PostgreSQL versión 9.2 debe estar instalado.
- El servidor web Apache en su versión 2.2.22-1ubuntu1.8 debe estar instalado.

Usabilidad

- El sistema podrá ser utilizado por usuarios con conocimientos básicos de informática (saber trabajar con un navegador Web).
- Se cuenta con un Manual de instalación del sistema.

Apariencia o interfaz

- Las ventanas del sistema: contienen claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.

- Interfaz: Se emplearon colores cómodos a la vista, sin acumulación de imágenes u objetos que distraigan al usuario de su objetivo. Permite a los usuarios interactuar con el sistema de forma fácil.

Seguridad

- El sistema requiere de la autenticación como primera acción, con un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica. A través de una identificación se pone en práctica los principios de la seguridad: disponibilidad, integridad y confidencialidad.

2.4 Propuesta de Solución

En la siguiente figura se muestra la solución de una aplicación Web que de forma automática generará una propuesta de tribunales de defensa de trabajos de diploma aplicando técnicas de Inteligencia Artificial, identificando de cada uno el Presidente, Secretario, Vocal y los Oponentes de las tesis asignadas al mismo, atendiendo al nivel de experticia en el tema, Categoría Docente y Categoría Científica de los profesores, agrupados todos por Áreas de Conocimientos, esta información debe ser entregada al Vicedecanato de Formación por los profesores. La aplicación además permitirá exportar a XML y Excel los tribunales y listados que se muestren, el archivo XML servirá de utilidad a otros sistemas como “Sistema informático de planificación automática de pre-defensas y defensas de los trabajos de diploma en la Facultad 2”.

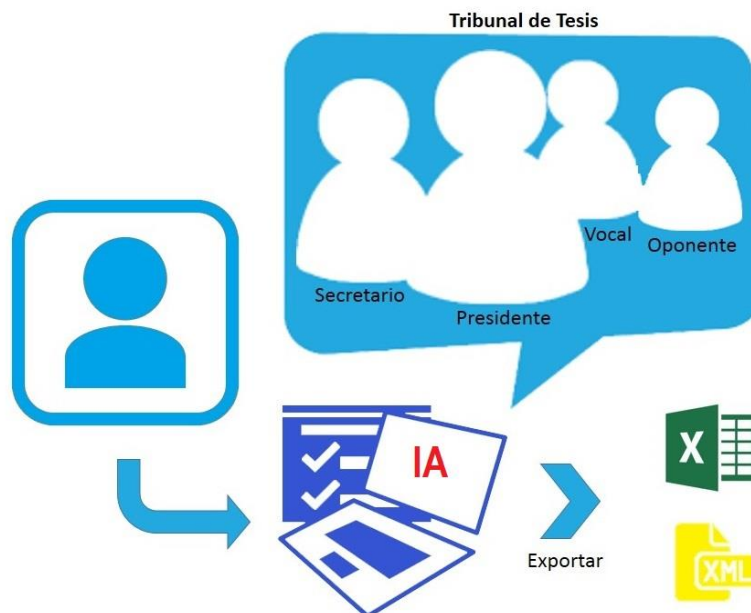


Figura 5 Propuesta de Solución

2.5 Descripción del Sistema

Para describir el sistema se han agrupado las funcionalidades en paquetes, conteniendo estos las funcionalidades que se encuentran en la misma área de interés. Del paquete Autenticación dependen Gestionar Profesores, Gestionar Tesis y Gestionar Nomencladores, porque en Autenticación está contenida la funcionalidad Autenticar usuario, de la que depende el acceso al resto de las funcionalidades del sistema. Por otra parte, Gestionar Tribunales depende de Gestionar Profesores y Gestionar Tesis, porque para generar un tribunal, es necesaria la información referente a las tesis y los profesores existentes.

El **Paquete Autenticación** contiene la funcionalidad: Autenticar Usuario.

El **Paquete Gestionar Profesores** contiene las funcionalidades: Gestionar Profesor, Listar Profesores y Buscar Profesores.

El **Paquete Gestionar Tesis** contiene las funcionalidades: Gestionar Tesis, Listar Tesis y Buscar Tesis.

El **Paquete Gestionar Tribunales** contiene las funcionalidades: Generar Tribunales, Gestionar Tribunales, Listar Tribunales y Buscar Tribunales.

El **Paquete Exportar** contiene las funcionalidades: exportar a XML y exportar a Excel todos los listados que se muestren. Estas funcionalidades son realizadas por todos los usuarios que acceden al sistema a diferencia de exportar a XML que solo puede hacerlo el administrador que responde al cargo de Vicedecano de Formación.

El **Paquete Gestionar Nomencladores** contiene las funcionalidades: Gestionar Categoría Científica, Gestionar Categoría Docente, Gestionar Áreas de Trabajo, Gestionar Rol, Gestionar Áreas de Conocimiento y Gestionar Nivel de Experticia.

Diagrama de Paquetes de Casos de Uso del Sistema

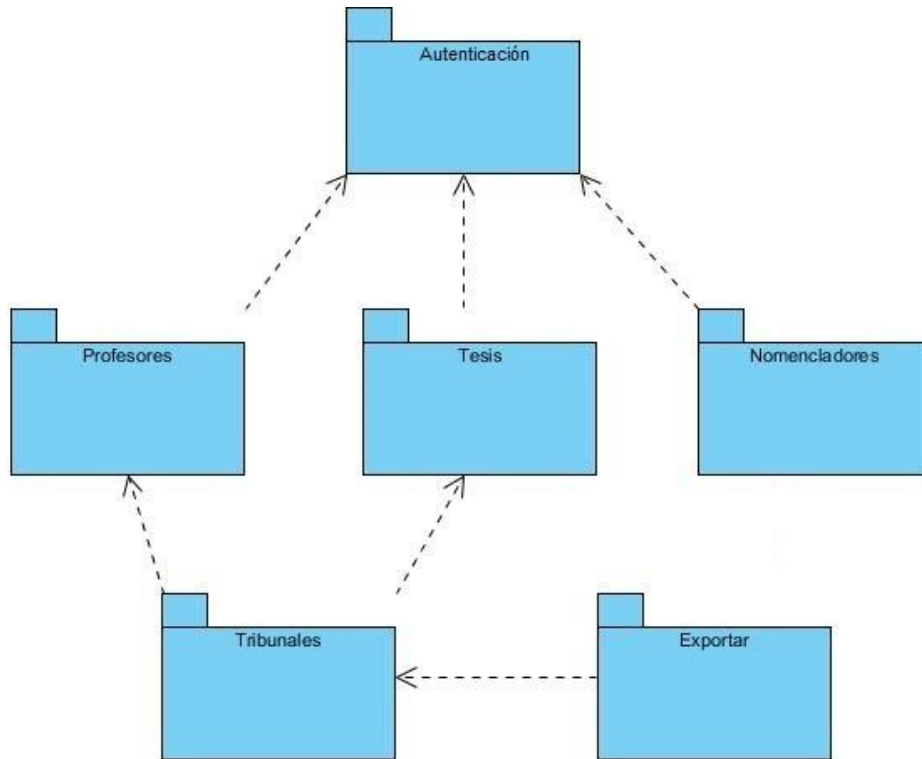


Figura 6 Diagrama de Paquetes de Caso de Uso del Sistema

A continuación se muestran los Diagramas de Casos de Uso del Sistema (DCUS) de cada uno de los paquetes:

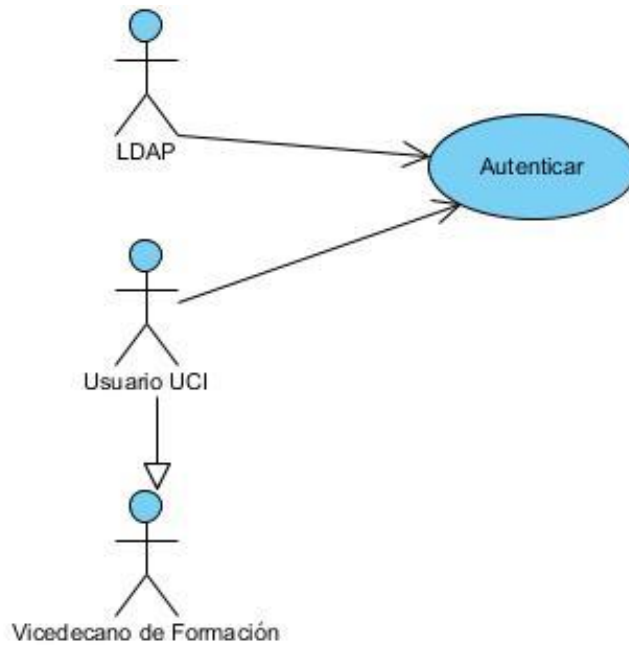


Figura 7 DCUS del Paquete Autenticación

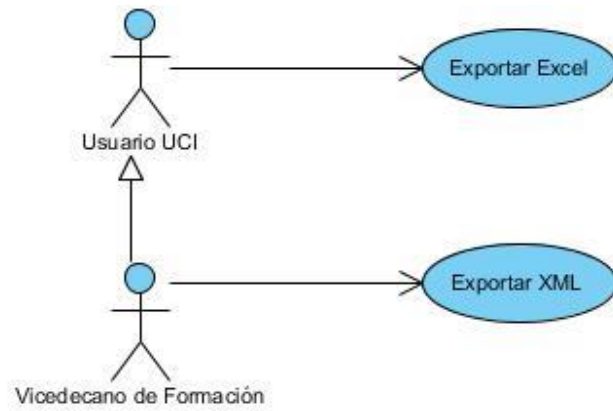


Figura 8 DCUS del Paquete Exportar

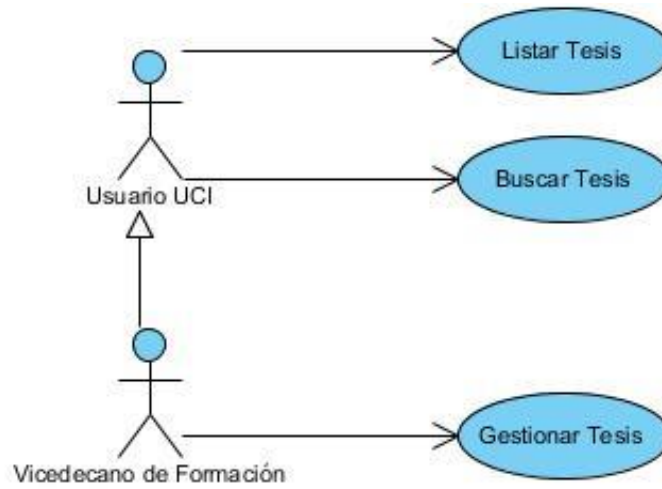


Figura 9 DCUS del Paquete Gestionar Tesis

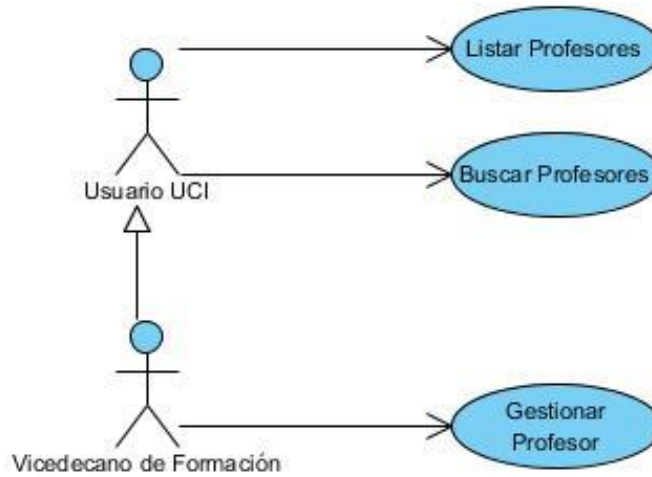


Figura 10 DCUS del Paquete Gestionar Profesores

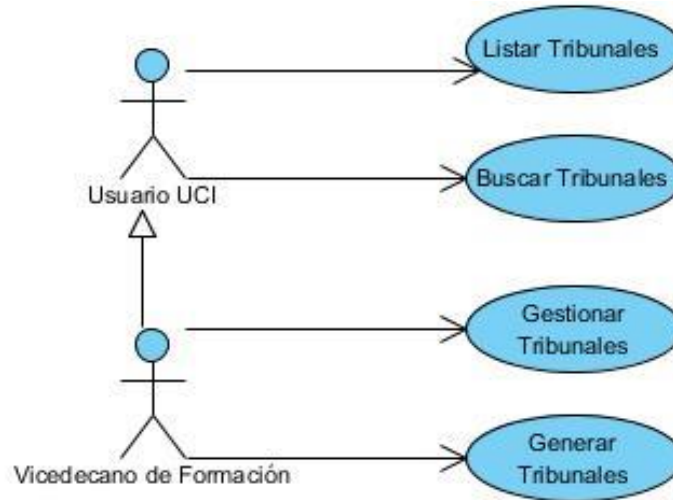


Figura 11 DCUS del Paquete Gestionar Tribunales

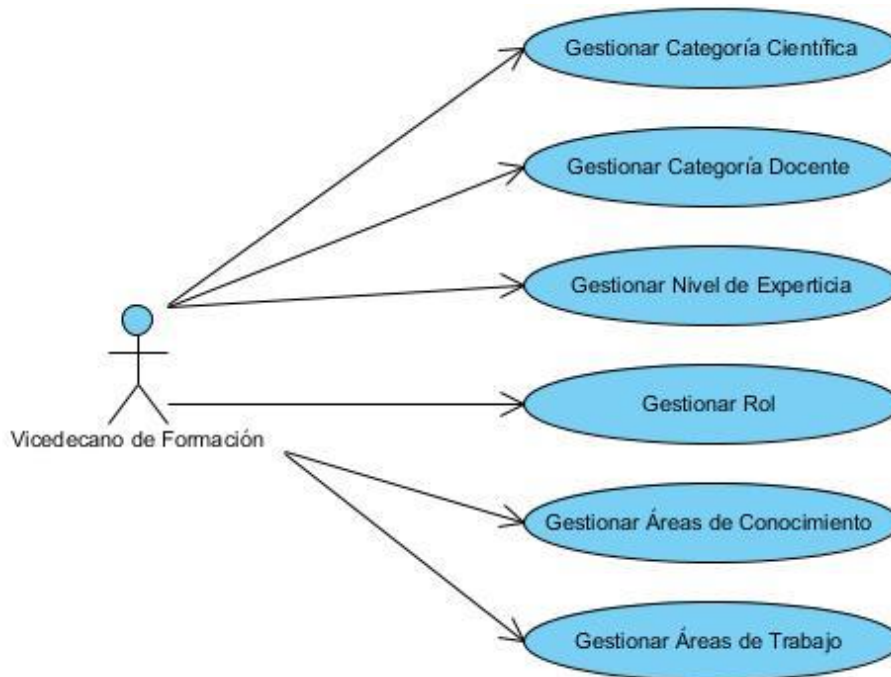


Figura 12 DCUS del Paquete Gestionar Nomencladores

2.6 Priorización de los Casos de Uso

La Priorización de los casos de uso es importante para dejar claras las funcionalidades indispensables, sin las que la aplicación no tiene razón de ser, que deben ser implementadas en el primer ciclo de desarrollo del software. En la Tabla 4 se puede apreciar la prioridad que se le dio a las funcionalidades del sistema a partir del criterio

del cliente, y los ciclos de desarrollo a las cuales están asociadas.

Tabla 4 Priorización de los Casos de Uso

Casos de Uso	Ciclo	Justificación
Generar Tribunales	1	Es la funcionalidad más importante del sistema, pues cumple con el objetivo general de la investigación.
Gestionar Profesores Listar Profesores Buscar Profesores		La gestión de los datos de los profesores es indispensable para la conformación de los tribunales, por ser la cantera para seleccionar los miembros de dichos tribunales.
Gestionar Tesis Listar Tesis Buscar Tesis		La gestión de las tesis es primordial, pues de ellas se parte para conformar los tribunales automáticamente. Sus datos son usados por la funcionalidad implícita en el caso de uso Generar Tribunales
Gestionar Tribunales Listar Tribunales Buscar Tribunales		La gestión de los tribunales es importante pues le permite al Vicedecano de Formación, luego de conformados los tribunales poder modificarlos, eliminarlos, adicionarlos y listarlos sin necesidad de generarlos nuevamente.
Autenticar Usuario		Garantiza la seguridad del sistema al ser solo el usuario “administrador” quien acceda a la información y la modifique. Además existen usuarios que solo pueden consultar.
Gestionar Categorías Científicas Gestionar Categoría Docente Gestionar Niveles de Experticia Gestionar Rol Gestionar Áreas de Trabajo Gestionar Áreas de Conocimiento	2	La gestión de nomencladores es necesaria a largo plazo, pues si fuese preciso modificar o eliminar alguno de estos elementos, o se necesita adicionar nuevos, con acceder a cada una de estas funcionalidades sería suficiente, sin actuar directamente sobre el código de la aplicación. Además, son imprescindibles para generar los tribunales.

Exportar a Excel		Son formatos que utilizan los usuarios para distribuir los tribunales conformados y puedan usarse en otros sistemas.
Exportar a XML		

2.7 Descripción de Casos de Uso Crítico

A continuación se describe el Caso de Uso **Generar Tribunales** y **Gestionar Roles**. El primero por ser el de mayor importancia para el cliente y por contener el algoritmo que va a dar solución a la situación problemática planteada y el segundo como ejemplo de comportamiento de todos los gestionar, en los que solo cambia la entidad del caso de uso. Las demás descripciones se encuentran en los Anexos.

Tabla 5 Descripción del Caso de Uso Generar Tribunales

Objetivo	Generar de forma automática los tribunales de defensa de trabajos de diploma.	
Actores	Vicedecano de Formación.	
Precondiciones	El usuario debe estar autenticado en el sistema.	
Resumen	El caso de uso se inicia cuando el Vicedecano de Formación necesita generar de forma automática los tribunales de defensa de trabajos de diploma. Para esto, selecciona la opción Generar Tribunales, y se generan los tribunales cumpliendo de esta manera con el objetivo.	
Flujo de eventos		
Flujo básico Generar Tribunales		
	Actor	Sistema
	1. El Administrador selecciona la opción "Generar".	2. Obtiene de la base de datos la información referente a los profesores y a las tesis.
		3. Realiza el algoritmo para generar los tribunales.
		4. Obtiene el resultado de dicho algoritmo.
		5. Genera los siguientes datos de cada tribunal: <ul style="list-style-type: none"> • Área de conocimiento del Tribunal • Presidente • Secretario • Vocal • Oponente • Tesis que le corresponden a ese Tribunal


		6. Muestra el mensaje “Los tribunales se han generado satisfactoriamente”.
Relaciones	CU Incluidos	No aplica.
	CU Extendidos	No aplica.
Requisitos funcionales	Generar Tribunales.	
Asuntos pendientes	La descripción del algoritmo se realiza en el capítulo 3.	
Prototipo		

Tabla 6 Descripción del Caso de Uso Gestionar Roles

Objetivo	Gestionar los roles de los miembros del tribunal.	
Actores	Vicedecano de Formación.	
Precondiciones	El usuario debe estar autenticado en el sistema.	
Resumen	El caso de uso se inicia cuando el usuario necesita gestionar los roles de los miembros de un tribunal. Para esto, selecciona la opción Administración y gestiona los roles cumpliendo de esta manera con el objetivo.	
Flujo de eventos		
Flujo básico Gestionar Roles		
	Actor	Sistema
	1. El administrador selecciona la opción “Administración”.	2. El sistema muestra las siguientes opciones en el panel de administración: <ul style="list-style-type: none"> • Categoría Científica. • Categoría Docente.

	<ul style="list-style-type: none"> • Niveles de Experticia. • Profesores. • Roles. • Tesis. • Tribunales. • Áreas de Conocimiento. • Áreas de Trabajo. • Usuarios.
3. El administrador selecciona la opción "Roles".	4. El sistema muestra la interfaz "Gestionar Roles" con el listado de roles existentes y las opciones: <ul style="list-style-type: none"> • Añadir Rol. • Borrar elementos seleccionados.
5. El administrador selecciona una de las siguientes opciones: Añadir Rol, Borrar elementos seleccionados o selecciona un rol. <p>a) Si selecciona la opción Añadir Rol, ir a la sección "Añadir Rol "</p> <p>b) Si decide Borrar elementos seleccionados, ir a la sección "Eliminar Rol".</p> <p>c) Si decide seleccionar un rol, ir a la sección "Modificar Rol".</p>	
Sección "Añadir Rol"	
Flujo básico Añadir Rol	
	6. El sistema muestra la interfaz "Añadir Rol" con el siguiente campo: <ul style="list-style-type: none"> • Nombre.
7. El Administrador introduce el nombre del rol y presiona el botón Grabar.	8. El sistema verifica que el campo no esté vacío.
	9. El sistema verifica que el rol no exista.
	10. El sistema almacena en la base de datos el rol introducido.

	11. El sistema muestra la interfaz “Gestionar Roles” con los roles añadidos.
Flujo Alternativo 7a “Botón Grabar y continua editando”	
7a.1. El Administrador presiona el botón Grabar y continuar editando.	7a.2. El sistema almacena en la base de datos el rol introducido y mantiene los campos llenos.
Flujo Alternativo 7b “Botón Grabar y añadir otro”	
7b.1. El Administrador presiona el botón Grabar y añadir otro.	7b.2. El sistema almacena en la base de datos el rol introducido y limpia los campos.
Flujo Alternativo 8a “Campos vacíos”	
	8a.1. El sistema verifica que el campo está vacío.
	8a.2. El sistema muestra el siguiente mensaje: “Este campo es obligatorio” y mantiene la misma interfaz para que el Administrador introduzca el rol.
Flujo Alternativo 9a “Rol existente”	
	9a.1. El sistema muestra el siguiente mensaje: “Ya existe rol con ese nombre” y mantiene la misma interfaz para que el usuario corrija los campos con errores.
Sección “Eliminar Rol”	
Flujo básico Eliminar Rol	
	6. El sistema verifica que exista al menos un rol seleccionado.
	7. El sistema muestra el mensaje de confirmación “¿Está seguro que quiere eliminar el rol seleccionado?” y el botón “Si, estoy seguro”.
8. El usuario presiona el botón “Si, estoy seguro”.	9. El sistema elimina el rol o roles seleccionados y los objetos y sus elementos relacionados. Muestra el mensaje “Eliminado/s rol/es satisfactoriamente”.
Flujo Alternativo 6a “No existe rol seleccionado”	

		6a.1. El sistema verifica que no exista al menos un rol seleccionado.
		6a.2. El sistema muestra el mensaje “Se deben seleccionar elementos para poder realizar acciones sobre estos”.
6a.3. El Administrador selecciona los roles que desee eliminar y presiona el botón “Borrar elemento seleccionado”.		6a.4. Ir a la acción 7 de la sección “Eliminar Rol”.
Sección “Modificar Rol”		
Flujo básico Modificar Rol		
		6. El sistema muestra la interfaz “Modificar Rol” con el siguiente campo: <ul style="list-style-type: none"> • Nombre.
7. El Administrador modifica el nombre del rol y presiona el botón Grabar.		8. Ir a la acción 8 de la sección “Añadir Rol”
Flujo Alterno 7a “Botón Grabar y continuar editando”		
7a.1. Ir a la acción 7a.1 del flujo alternativo 7a de la sección “Añadir rol”		
Flujo Alterno 8a “Campos vacíos”		
		8a.1. Ir a la acción 8a.1 del flujo alternativo 8a de la sección “Añadir rol”
Flujo Alterno 9a “Rol existente”		
		9a.1. Ir a la acción 9a.1 del flujo alternativo 9a de la sección “Añadir rol”
Relaciones	CU Incluidos	No aplica
	CU Extendidos	No aplica
Requisitos funcionales	No aplica.	
Asuntos pendientes	No aplica.	



Conclusiones del Capítulo

En el capítulo se obtuvieron las características del sistema que se presenta, lo cual contribuye a una correcta documentación y diseño de la aplicación. Se definieron los requisitos funcionales y no funcionales necesarios para el desarrollo del sistema. Se describieron el modelo de negocio y el diagrama de paquetes del sistema con sus respectivos diagramas de casos de uso. Se justificó la priorización de los casos de uso y se describieron dos de ellos: Generar Tribunales y Gestionar Roles, las descripciones de los casos de uso que faltan se encuentran en los anexos.

Capítulo 3. Diseño del Sistema informático para la conformación automática de tribunales de defensa de trabajos de diploma en la Facultad 2.

En el presente capítulo se define y describe la arquitectura propuesta para el diseño del sistema y se listan los patrones de diseño empleados para la correcta implementación del mismo. Además, se especifica el modelo de optimización y se describe el algoritmo utilizado.

3.1 Patrón de Arquitectura

La arquitectura de software es una descripción de los subsistemas y componentes de un sistema software, y de las relaciones entre ellos. Los subsistemas y componentes son especificados en diferentes vistas para mostrar las propiedades funcionales y no funcionales relevantes del sistema. La arquitectura de software es un artefacto; es el resultado de la actividad de diseño de software. (27)

Un patrón describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que puede reutilizarse continuamente. (28)

Los patrones de arquitectura expresan los esquemas de organización estructural fundamental para sistemas software. Además, proveen de un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen reglas y guías para la organización de las relaciones entre ellos. (27)

Patrón Arquitectónico seleccionado

El patrón Modelo-Vista-Controlador (MVC) es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un **modelo** que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de **vistas** que representa los formularios de entrada y salida de información, el tercero es un conjunto de **controladores** que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema. (29)

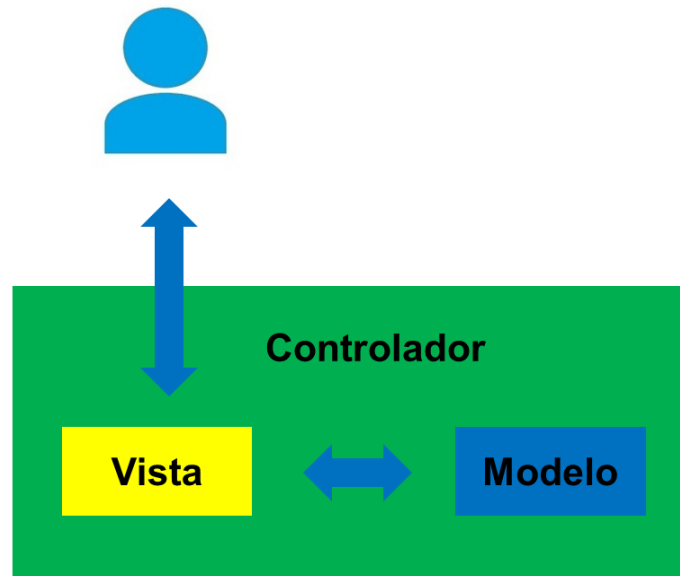


Figura 13 Modelo-Vista-Controlador

La Figura 14, muestra cómo se utiliza el patrón MVC en la aplicación.

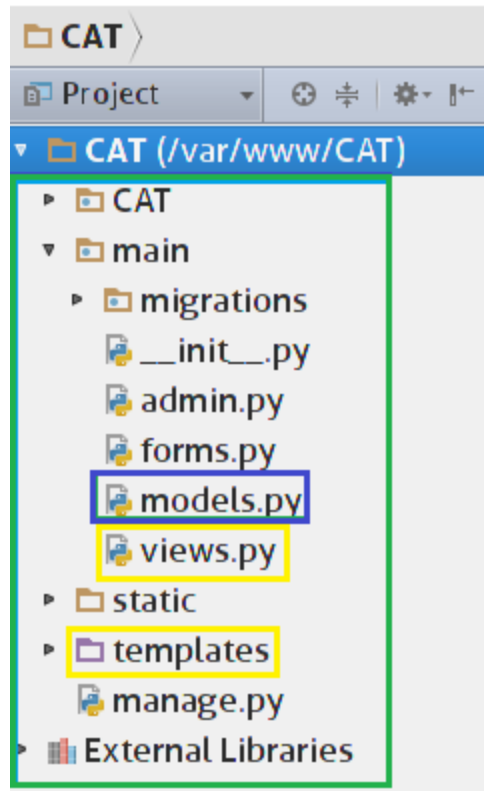


Figura 14 Representación del MVC en el sistema

La vista es conformada por las clases *views.py* y *templates*, el modelo se refleja en la clase *model.py* y por último el controlador está presente en el framework de desarrollo utilizado.

A continuación la Figura 15 y 16, muestran los Diagrama de Clases del Sistema para el Caso de Uso crítico Generar tribunales y el Gestionar Profesores, todos los Gestionar se comportan de la misma manera solo cambia la entidad con la que se relacionan y se encuentran en el archivo *diagramas.vpp* de la documentación del sistema.

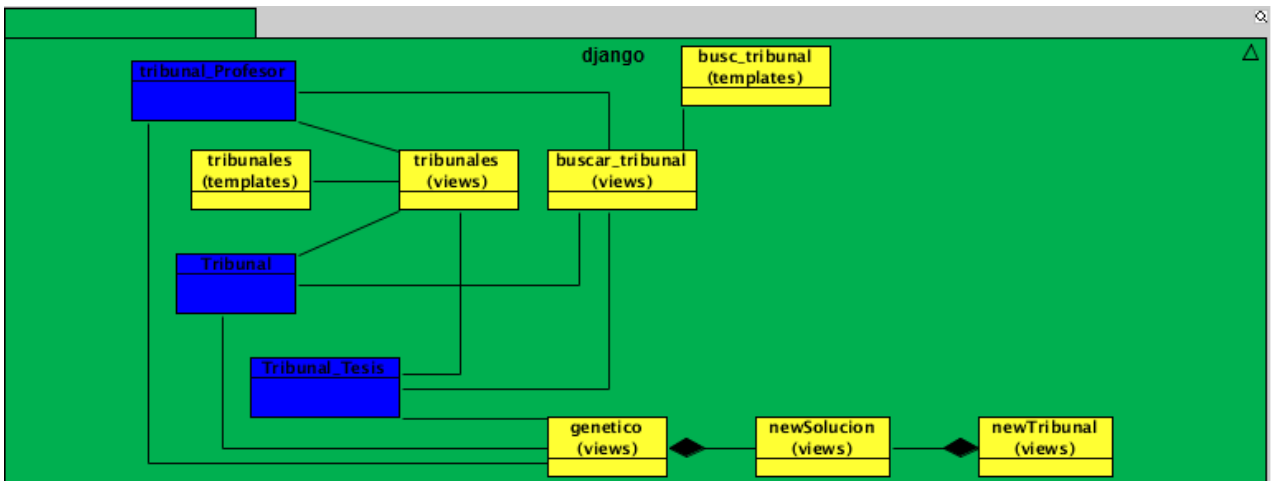


Figura 15 Diagrama de Clases del Sistema del CU Generar Tribunales

Este diagrama permite organizar estructuralmente el sistema partiendo del patrón arquitectónico seleccionado, mostrando una organización estructurada donde, las **vistas** se representan de color amarillo, el **modelo** por el color azul y el **controlador** por el color verde, para un mejor entendimiento del diagrama.

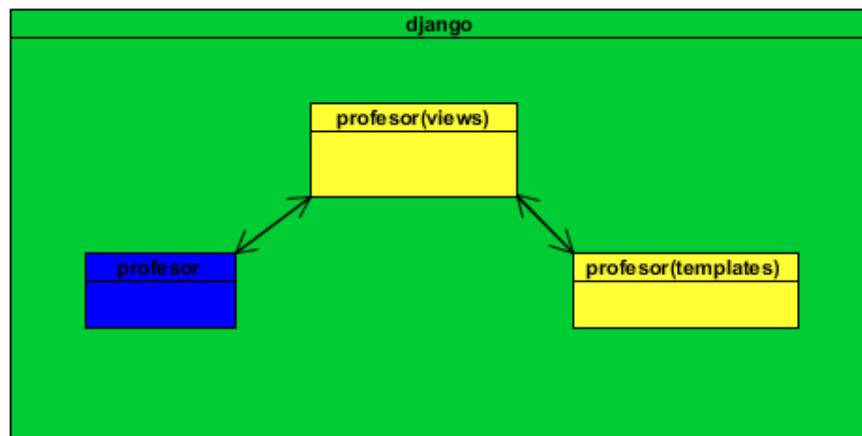


Figura 16 Diagrama de Clases del Sistema del CU Gestionar Profesores

En la clase **views** están las acciones para gestionar los profesores, con la clase **templates** interactúa el usuario y es la que muestra los datos, la clase **models** es la que accede a los datos y los envía a **views** y esta a **templates**.

3.2 Patrones de Diseño

Un patrón de diseño es una solución a un problema de diseño no trivial, que es efectiva y reusable. El uso de los patrones de diseño beneficia en gran medida el desarrollo y la calidad del software, pues evita la reiteración en la búsqueda de soluciones a problemas ya conocidos y resueltos anteriormente. Para el diseño de la aplicación se hizo uso de los Patrones Generales de Software para Asignar Responsabilidades (GRASP). A continuación, se listan algunos de los utilizados durante el desarrollo de la aplicación y se muestra cómo se evidencian en el código.

El patrón **Controlador** es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado, este patrón se encuentra ejemplificado es la clase *view.py*, ya que la misma contiene la lógica de ejecución.

El patrón **Experto** se utiliza para asignarle una responsabilidad a una clase experta en algún tipo de información, esta clase es la que contiene la información necesaria para realizar la labor que tiene encomendada, como es el caso de la clase *Tribunal* encargada de gestionar todo lo relacionado con la gestión de los tribunales.

```
class newTribunal:
    def __init__(self,area,tesis):
        self.area_conocimiento=area
        self.tesis=tesis
        self.integrantes=[]
```

El patrón **Creador** ayuda a identificar quién debe ser el responsable de la creación de nuevos objetos o clases. Tiene como propósito fundamental encontrar un creador para conectar con el objeto producido en cualquier evento. Crea una instancia de solución. Al escogerlo como creador, se da soporte al bajo acoplamiento.

```
s=newSolucion(self.tesisxTribunal)
```

El patrón **Bajo Acoplamiento** soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad.

3.3 Modelo de Datos

El modelo de datos se construye a partir del Diagrama de clases persistentes y en él, se representan las tablas en la Base de Datos. En la Figura 16, se muestra el modelo de datos del Sistema Informático para la conformación automática de tribunales de defensa de trabajos de diploma.

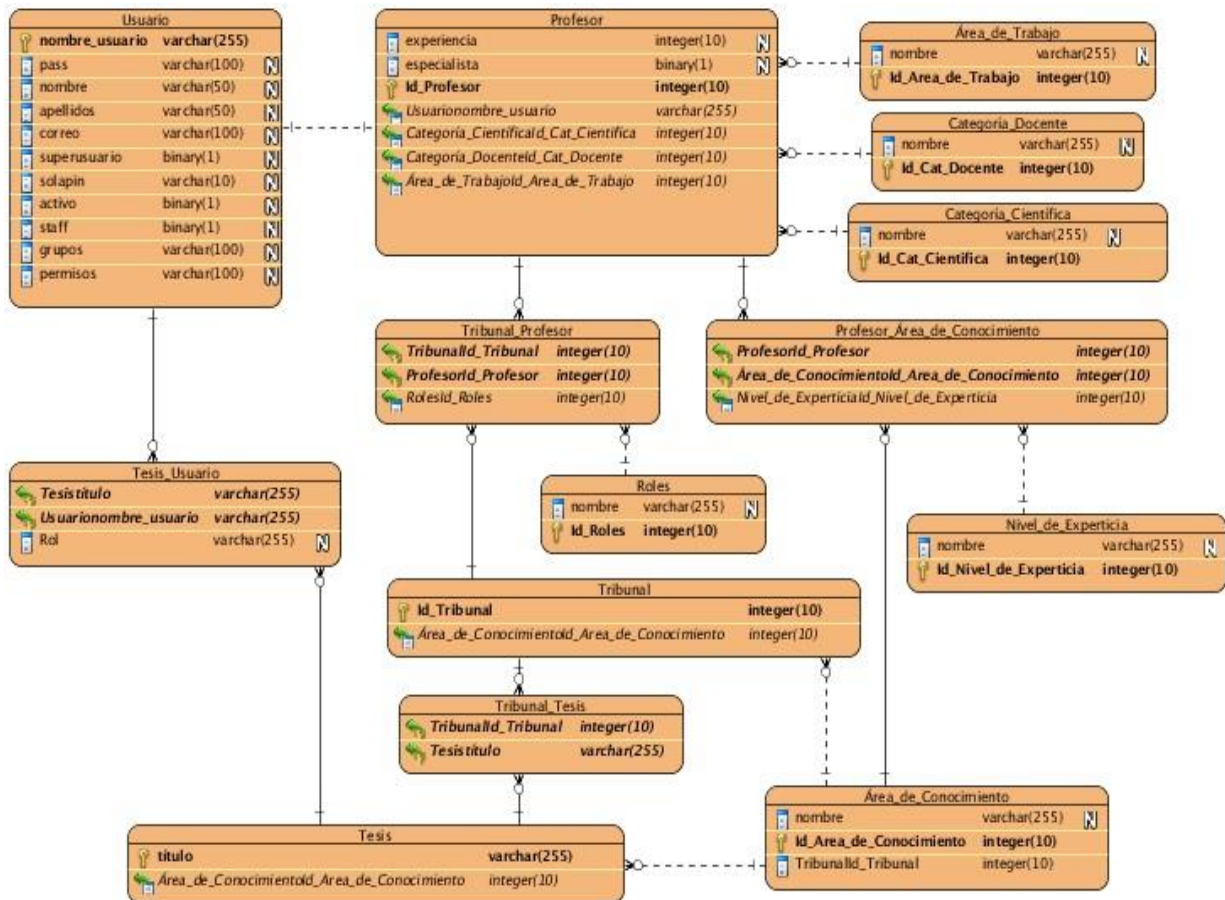


Figura 17 Modelo de Datos

El modelo contiene una tabla Usuario (contiene los datos de todos los usuarios que se autentican en la aplicación). Esta tabla tiene una relación con Profesor (contiene los datos de los profesores de la Facultad 2). Profesor está relacionada con Categoría_Docente (nomenclador con Instructor, Asistente, Auxiliar y Titular), también con Categoría_Científica (nomenclador con Ingeniero, Licenciado, Máster y Doctor), además con Área_de_trabajo (nomenclador con Centro CESIM, Centro CIGED, Centro

TLM, Colaboradores, Dpto. Ciencias Básicas, Dpto. Ciencias Sociales, Dpto. IGSW y Dpto. Programación), con la tabla Tribunal_Profesor (contiene los Id de las tablas Tribunal y Profesor), esta tabla se forma debido a la relación de muchos a muchos que tienen las clases Tribunal y Profesor. Además, profesor tiene relación con la tabla Profesor_Area_de_Conocimiento (contiene los Id de las tablas Profesor y Area_de_Conocimiento, la cantidad de años que lleva desarrollando esa área el profesor y el Id del Nivel de Experticia asociado), esta tabla se forma debido a la relación de muchos a muchos entre Profesor y Area_de_Conocimiento (nomenclador que contiene las áreas de conocimiento que se les asocia a las tesis y son las que han trabajado los profesores), Profesor tiene relación con Tesis_Usuario (contiene los Id de las tablas Tesis y Usuario y el rol que desempeña este en la tesis siendo Tutor o Autor). La tabla Roles (nomenclador con Oponente, Presidente, Secretario y Vocal). La tabla Tesis se relaciona con la tabla Tribunal, pues un tribunal puede atender a varias tesis y una tesis puede ser asignada a un tribunal. Además, se relaciona con la tabla Area_de_Conocimiento, pues el tema de la tesis está asociado a un área de conocimiento.

3.4 Modelo de Optimización a utilizar

La situación problemática que justifica la investigación se puede modelar como un problema de optimización combinatoria, particularmente un problema de distribución de recursos. En este caso, se deben distribuir recursos humanos, que son los profesores, para lograr la conformación de los tribunales, cumpliendo con el principio de integralidad (el tribunal como un todo, debe evaluar las habilidades metodológicas y las técnicas). Un problema de optimización combinatoria tiene como objetivo encontrar el máximo o el mínimo de una función específica, en un conjunto muy grande de soluciones discretas, sujeta a restricciones. El conjunto muy grande sería el compuesto por las combinaciones de profesores al conformar tribunales, y en este caso, habría que maximizar la capacidad de evaluación del tribunal definida por la función objetivo. (12)

Sea Soluciones = [Solución₁, Solución₂, ..., Solución₅₄] lista de soluciones a generar,

Donde

Solución = ([Tribunales], Costo, tesisxTribunal),

Donde

Tribunal = ([Tesis], Area_de_Conocimiento, [Integrantes]) es un elemento de la lista de Tribunales.

Donde

Integrantes = (Profesores) es el listado de profesores de cada tribunal.

Tesis = (Título, Área_de_Conocimiento, Autores, Tutores) es un elemento la lista de tesis que tiene un tribunal.

Se desea minimizar la **Función Objetivo** definida por:

$$F(\text{Solución}) = E_{\text{conocimiento}}(\text{Solución}) + E_{\text{disponibilidad}}(\text{Solución}) + E_{\text{grado_cient}}(\text{Solución}).$$

Donde

$$E_{\text{conocimiento}}(\text{Solución}) = \sum_{i=0}^{\text{S.T}} \sum_{j=0}^{\text{T.I}} 1 - \text{gradoConoc}(\text{Tribunales}_i, \text{Integrantes}_j, \text{area_conoc})$$

S.T = Solución.Tribunales (Tribunales de esa solución)

T.I = Tribunales.Integrantes (Integrantes de ese tribunal)

	Experticia	Años	Grado
gradoConoc(profesor, área) =	Avanzado	>= 10	1
	Avanzado	< 10	0.8
	Intermedio	>= 8	0.64
	Intermedio	< 8	0.48
	Básico	>= 5	0.32
	Básico	< 5	0.16

$$E_{\text{disponibilidad}}(\text{Solución}) = \sum_{i=0}^{\text{S.P}} (\text{profexTribunal} - \text{TribConProfe}(\text{Solución.Profesores}_i))^2$$

S.P = Solución.Profesores (Profesores que están en esa solución).

TribConProfe(Profesor) = Cantidad de tribunales en esa solución que tienen a ese profesor.

$$E_{\text{grado_cient}}(\text{Solución}) = \sum_{i=0}^{\text{S.T}} (\text{grad_Ideal} - \text{gradoEnTribunal}(\text{Solución.Tribunales}_i))^2$$

gradoEnTribunal(Tribunal) = Suma del grado de cada profesor del tribunal.

Grado(Profesor) =

Categoría Científica	Categoría Docente	Grado
Doctor	Titular	1
Doctor	Auxiliar	0.9
Doctor	Asistente	0.8
Doctor	Instructor	0.7
Máster	Auxiliar	0.6
Máster	Asistente	0.5
Máster	Instructor	0.4
Otros	Asistente	0.3
Otros	Instructor	0.2
Otros	Auxiliar	0.1

Todo lo anterior está sujeto a las siguientes restricciones:

- Las probabilidades estarán entre 0 y 1.
- La cantidad máxima de tesis que podrá atender un tribunal es 5 ($tesisxTribunal$).
- La cantidad máxima de tribunales en los que podrá estar un profesor es 3 ($profexTribunal$).

Para un mejor entendimiento de la Función Objetivo es necesario tener en cuenta las siguientes variables y constantes con sus significados:

Luego de varias pruebas realizadas al algoritmo arrojó que los valores asignados a las constantes son los más óptimos en cuanto a tiempo de ejecución del algoritmo y exactitud del resultado.

$np = 2$ (Cantidad de padres para realizar el cruzamiento).

$nh = 10$ (Cantidad de hijos a generar a partir de los padres).

$pm = 0,2$ (Probabilidad de mutación).

$pc = 0,7$ (Probabilidad de cruzamiento).

$maxEval = 3$ (Cantidad máxima de evaluaciones).

$errorPermitido = 40$ (Valor de la función objetivo que se considera como una solución óptima).

$aMutar = 2$ (Soluciones que van a mutar en caso de que haya mutación).

$tesisxTribunal = 5$ (Cantidad máxima de tesis que debe atender un tribunal, este número puede variar a 6 en dependencia de la disponibilidad de las tesis que existan).

$gradoldeal = 4$ (Grado ideal de un tribunal con respecto a las categorías científicas y docentes de sus integrantes).

$costo =$ variable o valor calculado mediante la Función Objetivo.

$profexTribunal = 3$ (Cantidad óptima de tribunales en la que debe estar un profesor).

Cantidad de tribunales (Cantidad de grupos que resultan al particionar las tesis por áreas de conocimiento y dividir cada una de estas particiones entre la cantidad de tesis por tribunal, de esta manera, al crear una nueva solución crea todos los tribunales necesarios para abarcar todas las tesis y asignar a cada uno las tesis y el área de conocimiento).

3.5 Algoritmo a Desarrollar

Para la solución del problema de optimización combinatoria modelado anteriormente, se empleará la metaheurística Algoritmo Genético.

Algoritmo:

```

Generar soluciones aleatoriamente
Evaluar()
Selección()
For(i = 0, i < maxEval):
    Cruzamiento()
    Mutación()
    Evaluar()
    Selección()
    Si soluciones[0].costo < errorPermitido:
        Break

```

Métodos utilizados en el algoritmo

Selección(): Ordena según el costo las soluciones de menor a mayor y coge los primeros np para la próxima evolución.

Cruzamiento(): De haber cruzamiento (número aleatorio < pc) se generan nh soluciones cada una con su conjunto de tribunales a la de un padre seleccionado al azar.

Mutación(): De haber mutación (número aleatorio < pm) se escoge al azar una parte en proporción a la constante aMutar de las soluciones y a cada una de ellas se le genera sus tribunales nuevamente al azar.

Evaluar(): Le aplica la Función Objetivo a cada una de las soluciones y le rellena el costo.

A continuación se describe el **algoritmo diseñado** aplicando esta metaheurística.

Primeramente se generan soluciones aleatorias donde se dividen todas las tesis por áreas de conocimiento, luego se conforman tribunales en dependencia de la cantidad de tesis por tribunal, después se toman los profesores en los que sus áreas de conocimiento coincidan con la de las tesis que atiende cada tribunal; si no existen profesores para llenar un tribunal, se escogerán de forma aleatoria profesores sin tener

en cuenta sus áreas de conocimiento. Luego se genera la cantidad de soluciones definidas en la suma de las constantes (np , nh).

Al tener las soluciones se evalúan en la Función Objetivo cada una de estas y se les agrega el costo calculado por dicha función.

Luego se realiza un proceso de selección ordenando las soluciones según el costo ascendentemente, tomándose las np primeras como mejores soluciones.

Posteriormente se ejecuta el ciclo desde cero hasta el número $maxEval$.

A continuación a las mejores soluciones se les aplica el método *Cruzamiento()*, para que haya cruzamiento se toma un número aleatorio y se compara con la constante pc , si dicho número es menor se generan nh soluciones, cada una con su conjunto de tribunales igual al de un padre seleccionado al azar generando nuevamente la suma de las constantes (nh , np).

Después se aplica el método *Mutación()*, para que haya mutación se toma un número aleatorio y se compara con la constante pm , si dicho número es menor se escogen al azar una parte en proporción de la constante $aMutar$ de las soluciones y se sustituyen por nuevos tribunales generados nuevamente al azar.

Luego se ejecutan los métodos *Evaluar()* y *Selección()* y se compara el costo de la primera solución de la lista obtenida con la constante $errorPermitido$, de ser menor finaliza la iteración y se retorna dicha solución, de lo contrario pasa a otra iteración ejecutando primeramente el método *Cruzamiento()*.

Conclusiones del Capítulo

Para el desarrollo del sistema se aplicó como patrón de arquitectura el Patrón MVC. Se emplearán los patrones de diseño Controlador, Experto, Creador y Bajo Acoplamiento. La modelación de los diagramas de paquetes y clases del diseño permitió organizar estructuralmente el sistema. Se modeló el problema a resolver como un problema de optimización combinatoria y, finalmente, se diseñó el algoritmo para generar automáticamente los tribunales de defensa de trabajos de diploma empleando la metaheurística AG.

Capítulo 4. Implementación y Prueba del Sistema informático para la conformación automática de tribunales de defensa de tribunales de defensa de trabajos de diploma en la Facultad 2.

El propósito de este capítulo es determinar y aplicar el tipo de prueba, así como mostrar la distribución física del sistema mediante el Diagrama de Despliegue. Se lleva a cabo la construcción de la herramienta, partiendo de los resultados obtenidos en la fase de diseño, así como las pruebas del mismo.

4.1 Diagrama de Despliegue del Sistema informático para la conformación automática de los tribunales de defensa de tribunales de defensa de trabajos de diploma de la Facultad 2.

El modelo de despliegue describe la distribución física del sistema, muestra cómo están distribuidos los componentes de software. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes. En este caso, se muestran todos los activos físicos que deben estar presentes para implantar la propuesta de solución.

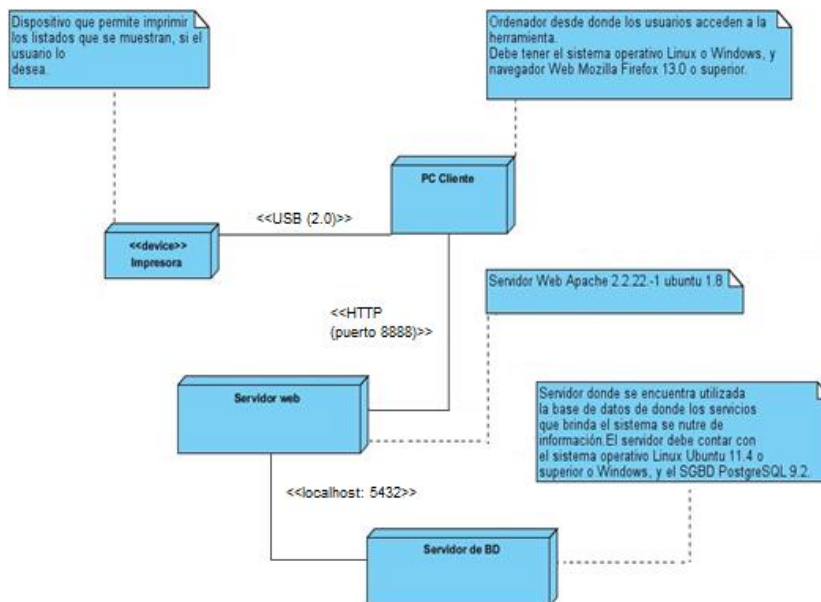


Figura 18 Diagrama de Despliegue

A continuación se muestra la descripción de cada nodo del diagrama de despliegue:

Tabla 7 Descripción de los nodos y enlaces de comunicación

Nodos y enlaces de comunicación	Descripción
Impresora	Dispositivo que permite imprimir los listados que se muestran, si el usuario lo desea.
PC Cliente	Ordenador desde donde los usuarios acceden al sistema. Debe tener el sistema operativo Linux o Windows, y navegador Web Mozilla Firefox 13.0 o superior.
Servidor Web	Apache 2.2.22.-1 ubuntu 1.8
Servidor de BD	Servidor donde se encuentra la base de datos utilizada de donde los servicios que brinda el sistema se nutren de información. El servidor debe contar con el sistema operativo Linux Ubuntu 11.4 o superior o Windows, y el SGBD PostgreSQL 9.2.
HTTP (puerto 8888)	Protocolo utilizado para la conexión entre las pc cliente y el servidor web.
USB (2.0)	Puerto utilizado para la conexión entre la pc cliente y la impresora.
Localhost: 5432	Puerto utilizado para la conexión entre el servidor web y el servidor de base de datos.

4.2 Validación del sistema

Las pruebas de aceptación del cliente realizadas para la validación del Sistema informático para la conformación automática de tribunales de defensa de trabajos de diploma en la Facultad 2 v1.0 se describen a continuación. Las pruebas que se desarrollarán serán del tipo: pruebas de caja negra y pruebas de aceptación del cliente, al efectuarse la entrega de la versión correspondiente. A continuación se abordan los aspectos significativos de estas pruebas y la recolección y valoración de no conformidades y solicitudes de cambio de la prueba de aceptación del cliente.

4.2.1 Prueba del sistema propuesto

El principal objetivo que tiene la ingeniería de software es desarrollar un producto informático con calidad. Para dar cumplimiento a ello se le realiza al sistema un conjunto de pruebas llamadas pruebas de software. Estas se hacen con el objetivo de encontrar errores en el producto y de esta manera conocer si el mismo tiene calidad o no en correspondencia a los requisitos funcionales identificados.

Las mismas pueden ser realizadas a lo largo del proceso de desarrollo del software, aunque por lo general se efectúan luego de haberse finalizada la implementación del sistema.

Prueba de Caja Negra

Prueba de Caja Negra (Black-Box Testing) son pruebas funcionales. Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente bajo prueba. Las herramientas básicas son observar la funcionalidad y contrastar con la especificación. (30)

Existen diferentes técnicas de prueba de Caja negra descritas por Pressman para validar la funcionalidad del sistema sin entrar a analizar su ejecución interna.

1. Métodos de prueba basados en grafos.
2. Análisis de valores límites.
3. Tabla ortogonal.
4. Partición equivalente.



Figura 19 Prueba de Caja Negra

Con el objetivo de garantizar la calidad del producto, demostrar que las funcionalidades del software son operativas, que la entrada es aceptada adecuadamente y se produce una salida correcta, al sistema propuesto se le realizaron pruebas de caja negra y dentro de esta prueba se realizó la técnica de Partición Equivalente, la cual es una de

las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software.

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada:

- Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada.
- Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. (30)
- Los mismos criterios se aplican a las salidas esperadas: hay que intentar generar resultados en todas y cada una de las clases.

Los resultados de las pruebas de caja negra se ven reflejados una vez llenado los diseños de casos de pruebas.

4.3 Diseño de pruebas

Un caso de prueba se diseña según las funcionalidades descritas en los casos de usos. Este diseño se elabora previamente a realizar las pruebas exploratorias, o de interfaz como también pueden llamarse, se parte de la descripción de los casos de usos del sistema, como apoyo para las revisiones. Cada planilla de caso de prueba recoge la especificación de un caso de uso, dividido en secciones y escenarios, detallando las funcionalidades descritas en él y describiendo cada variable que recoge el caso de uso en cuestión, además quedan plasmadas las revisiones realizadas al caso de prueba; así como un registro de todo aquello que no satisface y corresponde a la calidad del software. (31)

A continuación se muestra un caso de prueba para uno de los CU crítico **Generar Tribunales**.

CU: Generar Tribunales

Descripción general: Con este CU se genera una propuesta de tribunales de defensa de trabajos de diploma.

Condiciones de ejecución: El usuario tiene que estar autenticado como Administrador.

Matriz de datos

SC Generar Tribunales

Tabla 8 Caso de Prueba Generar Tribunales

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Generar Tribunales.	Permite generar una propuesta de tribunales de defensa de trabajos de diploma.	Se muestra una animación de “cargando” mientras se generan los tribunales y muestra el mensaje “Los tribunales se han generado satisfactoriamente” al terminar.	<ol style="list-style-type: none"> 1. Acceder al sitio web. 2. Autenticar usuario. 3. Seleccionar la opción “Generar”.
EC 1.2 Tribunales generados	En caso de que existan tribunales generados se le pregunta al usuario si desea generar nuevamente.	Muestra el mensaje “Al generar los tribunales se borrarán los existentes” para continuar con el proceso.	
EC 1.3 Faltan datos.	En caso que no se introduzcan con anterioridad los nomencladores, tesis o profesores se muestra un mensaje alertando sobre esta situación.	Muestra el mensaje “No fue posible generar los tribunales. (verificar datos de Nomencladores, Profesores y Tesis)” mostrando los datos inexistentes.	

4.3.1 Resultado del modelo de prueba de caja negra

A partir de la constatación del modelo de prueba de caja negra realizada se pudo determinar que: se realizaron 3 iteraciones, los resultados de la misma fueron los siguientes:

1. En la primera iteración se encontraron 7 NC, las más significativas fueron las siguientes:
 - Errores en la implementación de algunos vínculos entre las interfaces.
 - Faltas de ortografía en los nomencladores.
2. En la segunda iteración se encontraron 3 NC, la más significativa fue la siguiente:
 - La información de algunos listados generados no se mostraban de forma organizada.
3. En la tercera iteración los resultados fueron satisfactorios.

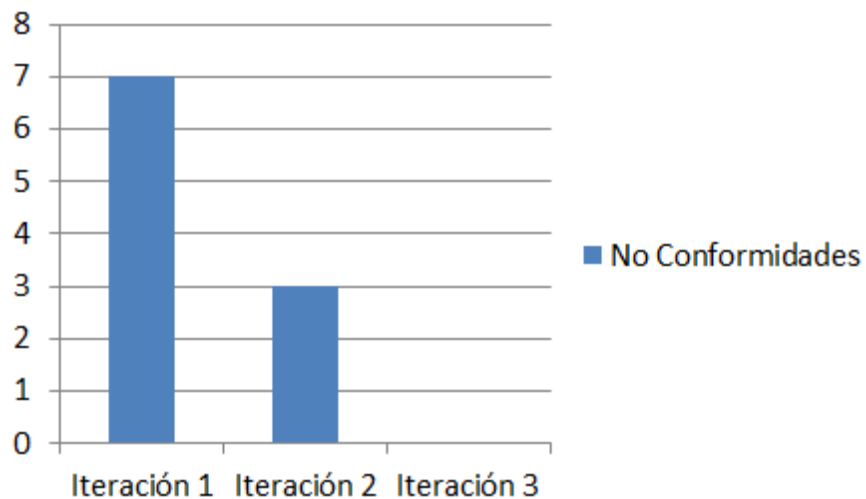


Figura 20 Resultados del método de prueba de caja negra

Como base de la solución se añadieron los tribunales con datos reales los cuales fueron generados satisfactoriamente en la aplicación atendiendo a los criterios utilizados en el algoritmo.

4.4 Prueba de aceptación del cliente

Las pruebas de aceptación del cliente son las encargadas de asegurar el cumplimiento de los Procesos Elementales del Negocio, los Casos de Uso, lo aceptado en los prototipos y lo planteado en el proyecto técnico.

Fases para la realización de las pruebas

Todas las pruebas se realizarán de forma manual y se probarán todas las funcionalidades implementadas en el módulo según la metodología del cliente. Estas pruebas se harán teniendo en cuenta los siguientes pasos:

- Organización de los escenarios de pruebas y capacitación del equipo de pruebas.
- Realización de pruebas de funcionalidad.
- Realización de las pruebas de interfaz de usuario.

4.4.1 Resultados de las pruebas de aceptación del cliente

Los resultados de las pruebas de aceptación realizadas al Sistema informático para la conformación automática de tribunales de defensa de trabajos de diploma de la Facultad 2 una vez concluida su implementación fueron las siguientes: las inquietudes encontradas fueron de tipo No Conformidad (NC). No se encontraron inquietudes de tipo Solicitud de Cambio. En general se encontraron 2 No Conformidades dentro del sistema que consistían en: faltas de ortografía en algunas interfaces y la información se mostraba desorganizada; cada una de estas no conformidades fueron correctamente resueltas por el equipo de desarrollo encargado.

4.4.2 Resultados de las pruebas de carga

Las pruebas de carga se llevan a cabo para determinar el comportamiento de los sistemas bajo condiciones tanto normales como de carga máxima.

Tabla 9 Pruebas de Carga del sistema

Cantidad máxima de evaluaciones	Tiempo de ejecución para 20 profesores y 12 tesis	Tiempo de ejecución para 93 profesores y 88 tesis
1	18 segundos	48 segundos
2	36 segundos	56 segundos
3	1 minuto	1 minuto y 30 segundos
4	1 minuto y 28 segundos	2 minutos y 11 segundos
5	1 minutos y 57 segundos	2 minutos y 17 segundos

Las pruebas de carga realizadas al sistema consistieron en aumentar la cantidad de profesores y tesis existentes a valores actuales y la cantidad de evaluaciones a realizar por el sistema, donde se evidencia el incremento del tiempo que demora la aplicación en realizar cada evaluación. Con el análisis de los datos representados, se define que la cantidad máxima de evaluaciones es 3, siendo la media de todos los tiempos calculados.

Conclusiones parciales

Una vez analizados los diferentes tipos de pruebas, se determinó utilizar el método de prueba de caja negra específicamente la técnica de partición equivalente, resolviéndose las NC detectadas lo cual permitió verificar el correcto funcionamiento de todos los requisitos funcionales. La prueba de aceptación final del cliente contribuyó a la validación de la herramienta verificando que los objetivos pactados con el cliente fueran resueltos satisfactoriamente.

Conclusiones

La realización de este trabajo responde a la necesidad de buscar una solución al problema planteado en la Facultad 2 de la Universidad de las Ciencias Informáticas. Una vez culminada la investigación se da cumplimiento al objetivo general planteado evidenciándose las siguientes conclusiones generales:

1. La revisión del estado del arte de los sistemas de distribución automática de recursos, evidencian la inexistencia de una solución concreta que permita la conformación automática de tribunales de defensa de trabajos de diploma en la facultad 2.
2. La solución propuesta se basó en un problema de asignación de recursos, permitiendo maximizar la disponibilidad de los profesores, el grado del conocimiento y el grado científico. El empleo de la técnica metaheurística Algoritmo Genético agilizó el proceso de implementación del algoritmo logrando mejores soluciones que las generadas por la aplicación inicial en el proceso de conformación de tribunales de la facultad 2.
3. La aplicación web implementada genera una propuesta de tribunales de trabajos de diploma, que cumple con todas las condiciones mínimas exigidas.
4. Las pruebas aplicadas al sistema arrojan las NC detectadas donde la corrección de las mismas permitió que el sistema quede libre de errores.

Recomendaciones

- Aplicar el sistema en la Facultad 2 en el proceso de culminación de estudio del curso 2015-2016, y realizar un estudio de los resultados obtenidos. Sobre la base de estos resultados, generalizar la aplicación de la propuesta a nivel de universidad.
- Realizar un análisis para definir la cantidad de tesis a asignar por cada tribunal.
- Utilizar otras metaheurísticas en la solución del problema y comparar las soluciones obtenidas con la propuesta realizada.
- Luego de modificar los tribunales generados adicionar validaciones teniendo en cuenta todos los criterios utilizados en la solución.

Referencias Bibliográficas

1. Superior, Ministerio de Educación. RESOLUCIÓN No. 210 del 2007. *Reglamento para el Trabajo Docente y Metodológico*. Habana : Edición, 2006.
2. Zayas, C Alvarez de. *Hacia una escuela de excelencia*. La Habana : Editorial Academia, 1996.
3. *The Nature of Mathematical Programming*. s.l. : INFORMS Computing Society. Vol. Mathematical Programming Glossary.
4. Martí, Rafael. *Procedimientos Metaheurísticos en Optimización Combinatoria*. Valencia : s.n.
5. Enrique Castillo, Antonio J. Conejo, Pablo Pedregal, Ricardo García y Natalia Alguacil. *Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia*. 2002.
6. Coppin, Ben. *Artificial Intelligence Illuminated*. [ed.] Jones and Barlett Publishers. Sudbury : 1, 2004. págs. 768. 0-7637-3230-3.
7. Díaz, Adenso. *Optimización Heurística y Redes Neuronales*. s.l. : Paraninfo, 1996.
8. P., Osman I. H. and Kelly J. *Meta-Heuristics: theory and applications*. Boston : Kluwer Academic, 1996. págs. 1-21.
9. MSC(ia)., Ing. Fernando Mejía P. [En línea] [En línea] <http://nando1-utb.blogspot.com/p/algoritmos-geneticos.html>.
10. Rushil Raghavjee, Nelishia Pillay. *A Comparison of Genetic Algorithms and Genetic Programming in Solving School Timetabling Problems*. s.l. : IEEE Press, 2012.
11. Holland, H. J. *Adaptation in Natural and Artificial Systems*. . University of Michigan : s.n., 1975.
12. Hernández, Suany Leyva. *Sistema informático para la conformación automática de Tribunales de Tesis de la Facultad 2*. La Habana : s.n., 2012.
13. Patón. *D.E.F.-M*. Alarcos : s.n., 2006.
14. Omg.org. Introduction to OMG UML. [En línea] 2009. [En línea] http://www.omg.org/gettingstarted/what_is_uml.htm.
15. Qué es Python. [En línea] [En línea] <http://www.desarrolloweb.com/articulos/1325.php>.
16. Argentina, Comunidad Python. PyAr. *Comparación de entornos de desarrollo*. [En línea] [Citado el: 5 de 5 de 2015.] <http://python.org.ar/wiki/IDEs>.

17. Qué es PyCharm. [En línea] [En línea] <http://hipertextual.com/archivo/2014/06/pycharm-ide-python/>.
18. Cristalab. *PyCharm: El mejor IDE para tus proyectos en Python*. [En línea] 2014. <http://www.cristalab.com/tutoriales/pycharm-el-mejor-ide-para-tus-proyectos-en-python-c114084/>.
19. Qué es Django. [En línea] [En línea] <http://computerhoy.com/noticias/internet/descubre-que-es-django-framework-web-modas-8641>.
20. Apache. *HTTP server project*. [En línea] 1997-2015. http://httpd.apache.org/ABOUT_APACHE.html.
21. IBM Rational Unified Process (RUP). [En línea] [En línea] <http://www-01.ibm.com/software/awdtools/rup/>.
22. Proceso Unificado de Rational. [En línea] 4 de 4 de 2015. http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational.
23. PostgreSQL-es. Portal en español sobre PostgreSQL. [En línea] [En línea] http://www.postgresql.org.es/sobre_postgresql.
24. Visual Paradigm for UML. UML tool for software application development. [En línea] [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
25. *Curso Modelado de Negocio con BPMN 2.0 y UML*. 2011.
26. Educación Eficaz en Ingeniería de Software. [En línea] 2005. <https://educacioningesoftware.wordpress.com/libre/>.
27. Buschmann, Frank. *Pattern-Oriented Software Architecture*.
28. Christopher, Alexander. *ISHIKAWA, Sara y SILVERSTEIN, Murray. A pattern language*. 1977.
29. Gutiérrez, Javier J. ¿Qué es un framework web?
30. Pressman. *Calidad de Software*. 2002.
31. —. *Ingeniería de Software. Un enfoque práctico*. 2000.

Bibliografía

1. Superior, Ministerio de Educación. RESOLUCIÓN No. 210 del 2007. *Reglamento para el Trabajo Docente y Metodológico*. Habana : Edicación, 2006.
2. Zayas, C Alvarez de. *Hacia una escuela de excelencia*. La Habana : Editorial Academia, 1996.
3. *The Nature of Mathematical Programming*. s.l. : INFORMS Computing Society. Vol. Mathematical Programming Glossary.
4. Martí, Rafael. *Procedimientos Metaheurísticos en Optimización Combinatoria*. Valencia : s.n.
5. Enrique Castillo, Antonio J. Conejo, Pablo Pedregal, Ricardo García y Natalia Alguacil. *Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia*. 2002.
6. Coppin, Ben. *Artificial Intelligence Illuminated*. [ed.] Jones and Barlett Publishers. Sudbury : 1, 2004. págs. 768. 0-7637-3230-3.
7. Díaz, Adenso. *Optimización Heurística y Redes Neuronales*. s.l. : Paraninfo, 1996.
8. P., Osman I. H. and Kelly J. *Meta-Heuristics: theory and applications*. Boston : Kluwer Academic, 1996. págs. 1-21.
9. MSC(ia)., Ing. Fernando Mejía P. [En línea] [En línea] <http://nando1-utb.blogspot.com/p/algoritmos-geneticos.html>.
10. Rushil Raghavjee, Nelishia Pillay. *A Comparison of Genetic Algorithms and Genetic Programming in Solving School Timetabling Problems*. s.l. : IEEE Press, 2012.
11. Holland, H. J. *Adaptation in Natural and Artificial Systems*. . University of Michigan : s.n., 1975.
12. Hernández, Suany Leyva. *Sistema informático para la conformación automática de Tribunales de Tesis de la Facultad 2*. La Habana : s.n., 2012.
13. Patón. *D.E.F.-M*. Alarcos : s.n., 2006.
14. Omg.org. Introduction to OMG UML. [En línea] 2009. [En línea] http://www.omg.org/gettingstarted/what_is_uml.htm.
15. Qué es Python. [En línea] [En línea] <http://www.desarrolloweb.com/articulos/1325.php>.
16. Argentina, Comunidad Python. PyAr. *Comparación de entornos de desarrollo*. [En línea] [Citado el: 5 de 5 de 2015.] <http://python.org.ar/wiki/IDEs>.

17. Qué es PyCharm. [En línea] [En línea]
<http://hipertextual.com/archivo/2014/06/pycharm-ide-python/>.
18. Cristalab. *PyCharm: El mejor IDE para tus proyectos en Python*. [En línea] 2014.
<http://www.cristalab.com/tutoriales/pycharm-el-mejor-ide-para-tus-proyectos-en-python-c114084/>.
19. Qué es Django. [En línea] [En línea]
<http://computerhoy.com/noticias/internet/descubre-que-es-django-framework-web-modas-8641>.
20. Apache. *HTTP server project*. [En línea] 1997-2015.
http://httpd.apache.org/ABOUT_APACHE.html.
21. IBM Rational Unified Process (RUP). [En línea] [En línea] <http://www-01.ibm.com/software/awdtools/rup/>.
22. Proceso Unificado de Rational. [En línea] 4 de 4 de 2015.
http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational.
23. PostgreSQL-es. Portal en español sobre PostgreSQL. [En línea] [En línea]
http://www.postgresql.org.es/sobre_postgresql.
24. Visual Paradigm for UML. UML tool for software application development. [En línea] [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
25. *Curso Modelado de Negocio con BPMN 2.0 y UML*. 2011.
26. Educación Eficaz en Ingeniería de Software. [En línea] 2005.
<https://educacioningesoftware.wordpress.com/libre/>.
27. Buschmann, Frank. *Pattern-Oriented Software Architecture*.
28. Christopher, Alexander. *ISHIKAWA, Sara y SILVERSTEIN, Murray. A pattern language*. 1977.
29. Gutiérrez, Javier J. ¿Qué es un framework web?
30. Pressman. *Calidad de Software*. 2002.
31. —. *Ingeniería de Software. Un enfoque práctico*. 2000.
32. Cargua, Angel. Sistema de asignación de recursos: implementado usando la arquitectura tcp/ip. [En línea] 5 de enero de 1999. [Citado el: 24 de febrero de 2015.]
<http://www.dspace.espol.edu.ec/handle/123456789/3184>.
33. cargua, Angel. Sistema de asignación de recursos: implementando usando la arquitectura tcp/ip. [En línea] [En línea] 15 de 01 de 1999.
<http://www.dspace.espol.edu.ec/handle/123456789/3184>.

34. Facundo E. Cancelo, Pablo N. Cababie, Gabriel Barrera, Daniela López De Luise. *Un nuevo enfoque para asignación óptima de múltiples recursos*. Buenos Aires : Universidad de Palermo.
35. Largo, Faraón Llorens. *Prácticas Prolog*. Alicante : s.n.
36. Java. Learn About Java technology. [En línea] [En línea] <http://www.java.com/en/about/>.
37. SWI-Prolog's Home. [En línea] [En línea] <http://www.swi'prolog.org/>.
38. NetBeans. [En línea] [En línea] http://netbeans.org/community/releases/72/index_es.html.
39. Ecured. [En línea] <http://www.ecured.cu/index.php?title=Especial%3ABuscar&search=a&Submit=Buscar+en+Ecured>.
40. Qué es Django. [En línea] [En línea] <http://django.es/>.
41. Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve? [En línea] 11 de junio de 2014. <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.
42. *Propuesta de técnicas evolutivas para la confección automática de tribunales de trabajos de diploma*. Hernández, Pavel Novoa. 2013, Revista Cubana de Ciencias Informáticas.
43. Fernando Borrás Rocher, José Vicente Segura Heras, Joaquín Sánchez Soriano, Manuel A. Pulido Cayuela, Jesús Tadeo Pastor Ciurana. Confección automática de tribunales para las pruebas de acceso a la universidad. XXVI Congreso Nacional de Estadística e Investigación Operativa, 6-9 de Noviembre de 2001, ISBN 84-8439-080-2, 2001.
44. Jorge Alberto Ospina Falla, Rafael Ricaurte Bernal. *Solución de un problema de asignación de horarios de mantenimiento en máquinas, equipos e instalaciones a la empresa Powel Continental Ltda. mediante la programación de un algoritmo genético*. Chia : s.n, 2004.

Anexos

Tabla 10 Descripción del Caso de Uso Autenticar Usuario

Objetivo	Permitir al usuario el acceso al sistema una vez autenticado.	
Actores	Usuario UCI.	
Precondiciones	-	
Resumen	El caso de uso se inicia cuando el Usuario UCI inicia el sistema, que le muestra la ventana de autenticación para poder interactuar con el mismo.	
Flujo de eventos		
Flujo básico Autenticar		
	Actor	Sistema
	1. Inicia el sistema.	2. Muestra una ventana de autenticación con los campos a completar: <ul style="list-style-type: none"> • Usuario • Contraseña
	3. Introduce el usuario y la contraseña y presiona “Entrar”.	4. El sistema compara los datos introducidos con los existentes en la base de datos.
		5. Le da acceso al usuario al resto de las funcionalidades.
Flujo Alternativo 4a “Datos incorrectos”		
		4a. 1. Muestra un mensaje de error notificando que los datos son incorrectos. 4a. 2. Vuelve al paso 3.
Relaciones	CU Incluidos	No aplica.
	CU Extendidos	No aplica.
Requisitos funcionales	No aplica.	
Asuntos pendientes	No aplica.	



Tabla 11 Descripción del Caso de Uso Exportar a Excel

Objetivo	Permitir al usuario exportar los listados que se muestren.
Actores	Usuario UCI.
Precondiciones	El usuario debe estar autenticado en el sistema y los listados deben estar generados.
Resumen	El caso de uso se inicia cuando el Usuario UCI lista la información referente a los profesores, tesis o tribunales.
Flujo de eventos	
Flujo básico Autenticar	
Actor	Sistema
<p>1. Selecciona una de las siguientes opciones: Listar profesores, Listar tesis o Listar tribunales.</p> <ul style="list-style-type: none"> a) Si selecciona la opción Listar profesores, ir a la sección “Listar profesores” b) Si decide Listar tesis, ir a la sección “Listar tesis”. c) Si decide Listar tribunales, ir a la sección “Listar tribunales”. 	
Sección “Listar profesores”	
Flujo básico Listar profesores	
	2. El sistema muestra la interfaz “Listar profesores”
3. Selecciona la opción “Exportar a Hoja de Cálculo”	4. El sistema exporta el listado mostrado.
Sección “Listar tesis”	


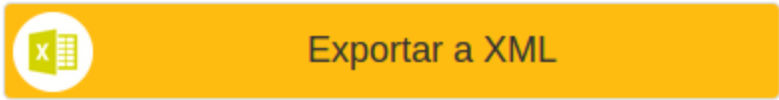

Flujo básico Listar tesis		
	5. El sistema muestra la interfaz “Listar tesis”	
6. Selecciona la opción “Exportar a Hoja de Cálculo”	7. El sistema exporta el listado mostrado.	
Sección “Listar tribunales”		
Flujo básico Listar tribunales		
	8. El sistema muestra la interfaz “Listar tribunales”	
9. Selecciona la opción “Exportar a Hoja de Cálculo”	10. El sistema exporta el listado mostrado.	
Relaciones	CU Incluidos	No aplica.
	CU Extendidos	No aplica.
Requisitos funcionales	Listar profesores, Listar tesis y Listar tribunales.	
Asuntos pendientes	No aplica.	
Prototipo		

Tabla 12 Descripción del Caso de Uso Exportar a XML

Objetivo	Permitir al usuario exportar los listados que se muestren.
Actores	Vicedecano de Formación.
Precondiciones	El usuario debe estar autenticado en el sistema y los listados deben estar generados.
Resumen	El caso de uso se inicia cuando el Vicedecano de Formación lista la información referente a los profesores, tesis o tribunales.
Flujo de eventos	
Flujo básico Autenticar	
Actor	Sistema
1. Selecciona una de las siguientes opciones: Listar profesores, Listar tesis o Listar tribunales. a) Si selecciona la opción Listar	

<p>profesores, ir a la sección “Listar profesores”</p> <p>b) Si decide Listar tesis, ir a la sección “Listar tesis”.</p> <p>c) Si decide Listar tribunales, ir a la sección “Listar tribunales”.</p>		
Sección “Listar profesores”		
Flujo básico Listar profesores		
		2. El sistema muestra la interfaz “Listar profesores”
3. Selecciona la opción “Exportar a XML”		4. El sistema exporta el listado mostrado.
Sección “Listar tesis”		
Flujo básico Listar tesis		
		5. El sistema muestra la interfaz “Listar tesis”
6. Selecciona la opción “Exportar a XML”		7. El sistema exporta el listado mostrado.
Sección “Listar tribunales”		
Flujo básico Listar tribunales		
		8. El sistema muestra la interfaz “Listar tribunales”
9. Selecciona la opción “Exportar a XML”		10. El sistema exporta el listado mostrado.
Relaciones	CU Incluidos	No aplica.
	CU Extendidos	No aplica.
Requisitos funcionales	Listar profesores, Listar tesis y Listar tribunales.	
Asuntos pendientes	No aplica.	
Prototipo		

Acta de aceptación del cliente:



Acta de aceptación

ACTA DE ACEPTACIÓN

Producto: Sistema informático para la conformación automática de tribunales de trabajos de diploma en la Facultad 2.



Involucrados en el proceso:

- **Estudiantes:** Yerandy Fernández García.
Reydel Capote Coipel.
- **Tutores:** Ing. Bárbara Triana Morales
Ing. Suany Leyva Hernández.

Observaciones del proceso:
Las no conformidades detectadas en el proceso de revisión fueron resueltas. Se comprobó la correcta implementación del Sistema informático para la conformación automática de tribunales de trabajos de diploma en la Facultad 2. Por tanto se acepta con fecha de 22 de mayo de 2015 la aplicación web propuesta.

Lista de productos que son aceptados y que deben ser entregados:

- Código fuente de la aplicación.
- Dependencias de la aplicación.
- Manual de instalación de la aplicación.

Entrega	Recibe
Nombre y Apellidos: Yerandy Fernández García Reydel Capote Coipel	Nombre y Apellidos: Dra. Aymée Hernández Calzada
Cargo: Estudiante	Cargo: Vicedecana de Formación
Firma: 	Firma: 
Comentarios: Los productos aceptados deben ser entregados al cliente previo a la defensa.	