



PROYECTO DE GRADO

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

DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE PROPUESTAS DE
PROYECTOS DE GRADOS Y PERSONAL DEL DEPARTAMENTO DE COMPUTACIÓN
DE LA ESCUELA DE INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD DE LOS
ANDES BASADO EN MANEJADORES DE BASES DE DATOS NOSQL.

Por

Br. Requadri Parra Rhona Fabiana

Tutor: Prof. Hernández Domingo

Febrero 2017

©2017 Universidad de Los Andes Mérida, Venezuela

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY - NC - SA 3.0 VE)

**Desarrollo De Un Sistema Web Para La Gestión De Propuestas De
Proyectos De Grados Y Personal Del Departamento De Computación De
La Escuela De Ingeniería De Sistemas De La Universidad De Los Andes
Basado En Manejadores De Bases De Datos NoSQL.**

Br. Requadri Parra Rhona Fabiana

Proyecto de Grado — Sistemas Computacionales, 120 páginas

Resumen: El presente proyecto está basado en el desarrollo de un sistema web para el Departamento de Computación de la Escuela de Sistemas de la Facultad de Ingeniería de la Universidad de Los Andes. El sistema permite tener una mejor manipulación y control de la información facilitando así las actividades que se deben realizar continuamente en dicho Departamento, la finalidad de este proyecto es mejorar los procesos que se manejan de forma manual haciéndolos que sean lentos y ocasionen pérdidas de información, este trabajo permite a los usuarios obtener información requerida como lo son: las propuestas y proyectos de grado, actas, agendas, personal, noticias y documentos de dicha sección, el desarrollo de la investigación estará basada en la metodología otorgada por el método WATCH conjuntamente con el Lenguaje Modelado Unificado (UML), además se utilizará un manejador de base de datos NoSQL bajo la plataforma de MongoDB..

Palabras clave: Sistema Web, NoSQL, MongoDB, Método Watch

Este trabajo fue procesado en $\text{\LaTeX}$.

Índice general

Índice de Tablas	VII
------------------	-----

Índice de Figuras	VIII
-------------------	------

Agradecimientos	XI
-----------------	----

1. Introducción	1
1.1. Antecedentes	2
1.2. Contexto de la Organización	3
1.3. Planteamiento del Problema	5
1.4. Justificación	5
1.5. Objetivos	6
1.5.1. Objetivo General	6
1.5.2. Objetivos Específicos	6
1.6. Metodología	7
1.7. Alcance	8
1.8. Distribución de Capítulos	8
2. Marco Teórico	10
2.1. Sistemas de Información	10
2.1.1. Funcionamiento de los Sistemas de Información	10
2.1.2. Clasificación	11
2.1.3. Sistemas de Información Web	12
2.2. Método <i>Watch</i>	12

2.2.1.	Modelo de Procesos WATCH	14
2.3.	UML	16
2.4.	Bases de Datos	18
2.4.1.	Base de Datos	18
2.4.2.	Bases de datos no relacionales	18
2.4.3.	MongoDB	19
2.4.3.1.	Cómo Funciona MongoDB	20
2.5.	Evaluación Y Comparación Del Rendimiento Del Manejador De Base De Datos MongoDB Para Aplicaciones Con Grandes Cantidades De Datos.	20
2.5.1.	Primer Test	22
2.5.2.	Segundo Test	23
2.5.3.	Tercer Test	24
2.6.	Modelo Vista Controlador (MVC)	26
2.6.1.	Modelo	27
2.6.2.	Vistas	27
2.6.3.	Controlador	27
2.7.	Express	28
3.	Modelado Y Análisis De Requerimientos	29
3.1.	Modelado de Negocios	29
3.1.1.	Visión del Sistema de Negocios	29
3.1.2.	Misión del Sistema de Negocios	30
3.2.	Identificación de Actores del Sistema	30
3.2.1.	Jerarquía de actores	31
3.3.	Modelado de los Procesos del Sistema	31
3.3.1.	Descripción de los Procesos de Negocio Asociados a los Actores del Sistema	32
3.3.2.	Reglas de Negocio	37
3.4.	Definición y Especificación de Requisitos	37
3.4.1.	Requisitos Funcionales	38
3.4.2.	Requisitos No Funcionales	39
3.5.	Casos de Uso	40

3.5.1. Definición de los Casos de Uso	40
3.5.2. Descripción de los Casos de Uso	40
3.5.3. Diagramas de Casos de Uso	47
4. Diseño Del Sistema	51
4.1. Definición de los Componentes.	51
4.2. Vista Estructural.	52
4.3. Vista de Implementación o Desarrollo.	52
4.4. Vista de Ambiente.	53
4.5. Vista de Comportamiento o Despliegue.	55
4.6. Diseño de la Interfaz del Usuario.	56
4.6.1. Estructura de Navegación.	56
4.6.1.1. Estilo visual de las páginas.	58
4.7. Estructura de la base de datos.	60
5. Implementación y Pruebas Del Sistema	62
5.1. Interfaz Gráfica	62
5.2. Ensamblaje de Componentes	67
5.3. Pruebas del Sistema	75
5.3.1. Pruebas de caja blanca	75
5.3.2. Pruebas caja negra	80
5.3.2.1. Prueba de Registro de Propuesta de Proyecto de Grado	80
5.3.2.2. Prueba de Registro de Currículum de Profesores	81
5.3.2.3. Prueba de Registro de Noticias	82
5.3.2.4. Prueba de Registro de Proyecto de Grado	82
5.3.3. Pruebas de Navegación	83
6. Conclusiones Y Recomendaciones	86
6.1. Conclusiones	86
6.2. Recomendaciones	87
Bibliografía	88

Anexos	90
A. Método WATCH	91
B. Diagramas de Caso de Uso y Especificación Textual	98
B.1. Diagramas de Caso de Uso	98
B.2. Descripción textual de Diagramas de Caso de Uso	101
C. Diagramas de Secuencia	106
D. Ensamblaje de Componentes	109
D.1. Procedimientos para el manejo de datos relacionados con profesores: . .	109
D.2. Procedimientos para el manejo de datos relacionados con preparadores:	111
D.3. Procedimientos para el manejo de datos relacionados con documentos: .	112
D.4. Procedimientos para el manejo de datos relacionados con actas:	113
E. Instalación y Creación de una Base de Datos en MongoDB	115
F. Implementación en Servidor	118
F.1. <i>Instalación de Node.js</i>	118
F.2. <i>Instalando NVM</i>	118
F.3. <i>Instalando Node.js</i>	119
F.4. <i>Instalando Express</i>	120

Índice de Tablas

3.1. Diagramas de actividades realizadas por los actores del sistema	37
3.2. Actividades realizadas por los actores del sistema	40
3.3. CU6. Inicio de Sesión	41
3.4. CU3. Gestionar Propuesta de Proyecto de Grado	42
3.5. CU8. Consultar Propuesta de Proyecto de Grado	43
3.6. CU11. Editar Propuesta de Proyecto de Grado	44
3.7. CU5. Gestionar Noticias o Documentos	45
3.8. CU10. Consultar Noticias o Documentos	46
3.9. CU12. Editar Noticias o Documentos	47
5.1. Pruebas de Navegación.	85
B.1. CU13. Creación de Cuentas de Usuarios	101
B.1. CU14. Consultar Propuesta de Proyecto de Grado	102
B.1. CU15. Gestión de Profesores	103
B.1. CU16. Consulta de Agendas o Actas	104
B.1. CU17. Gestión de Personal	105

Índice de Figuras

1.1. Estructura Organizativa de EISULA	4
2.1. Funcionamiento de los Sistemas de Información.	11
2.2. Estructura de Funcionamiento de una Base de Dato.	19
2.3. Preparación de la Base de Datos para el Manejador PostgreSQL.	21
2.4. Preparación Base de Datos para el Manejador de MySQL.	21
2.5. Preparación de Base de Dato para el Manejador MongoDB.	21
2.6. Primer Test Conteo de Tuits y Usuarios.	22
2.7. Segundo Test " <i>Select</i> " buscando por id.	23
2.8. Tercer Test Aplicado en un CRUD para PostgreSQL.	24
2.9. Tercer Test Aplicado en un CRUD para MySQL.	25
2.10. Tercer Test Aplicado en un CRUD para MongoDB.	25
2.11. Diagrama MVC	28
3.1. Jerarquía de Actores.	31
3.2. Modelo de los Procesos del Sistema de Negocio.	32
3.3. Autenticación de Usuarios.	48
3.4. Diagrama Caso de Uso Gestión General del Sistema.	48
3.5. Diagrama Caso de Uso Gestión Graduando.	49
3.6. Diagrama Caso de Uso Gestión Secretaria.	49
3.7. Diagrama Caso de Uso Gestión Profesor.	50
4.1. Diagrama de Colecciones de la Base de Datos Db6.	52
4.2. Diagrama de Componentes.	53
4.3. Diagrama de Secuencia Registro de Usuario.	54

4.4. Diagrama de Secuencia Registro de Actas.	54
4.5. Diagrama de Secuencia de Registro de Propuesta de Proyecto de Grado.	55
4.6. Diagrama de Despliegue.	55
4.7. Grafo Navegación Usuario General.	57
4.8. Grafo de Navegación Usuario Administradores.	58
4.9. Colecciones de Base de Datos.	61
5.1. Vista Principal del Sistema.	63
5.2. Vista de Acceso al Sistema	64
5.3. Acceso Correcto al Sistema como Jefe de Departamento.	65
5.4. Mensaje de Alerta de Contraseña Incorrecta.	65
5.5. Registro de Usuario.	68
5.6. Verificar Usuario.	68
5.7. Registro de Graduando.	69
5.8. Vista de Información de Graduando.	70
5.9. Registro Noticias.	71
5.10. Consulta de Noticias.	72
5.11. Registro de Agendas.	73
5.12. Consulta de Agendas.	73
5.13. Integración Componentes del Sistema	75
5.14. Prueba de Registro de Propuesta de Proyecto de Grado.	81
5.15. Prueba de Registro de Currículum de Profesores.	81
5.16. Prueba de Registro de Noticias.	82
5.17. Prueba de Registro de Proyecto de Grado.	83
A.1. Modelo 3 Capas	91
A.2. Modelo de una aplicación empresarial basada en componentes	92
A.3. Modelo de procesos WATCH	93
A.4. El flujo de trabajo del Modelado del Negocio	93
A.5. Procesos de definición y especificación de requisitos	94
A.6. Procesos del Diseño Arquitectónico	94
A.7. Procesos del Diseño de Componentes	95

A.8. Procesos del Aprovisionamiento de Componentes	95
A.9. Procesos del Ensamblaje de Componentes	96
A.10. Procesos de pruebas del subsistema	96
A.11. Procesos de la Entrega de la Aplicación	97
 B.1. Creación de Cuentas de Usuarios.	 98
B.2. Consulta de Propuesta de Proyecto de Grado.	99
B.3. Gestión de Profesores.	99
B.4. Consulta Información de Profesores.	100
B.5. Gestión Agendas.	100
 C.1. Diagrama de Secuencia de Ingreso al Sistema.	 106
C.2. Diagrama de Secuencia Edición de Propuesta de Proyecto de Grado. . .	107
C.3. Diagrama de Secuencia Edición de Actas.	107
C.4. Diagrama de Secuencia Registro Currículum Profesor.	108
 D.1. Registro de Profesor.	 109
D.2. Información Profesor.	110
D.3. Registro de Preparador.	111
D.4. Vista de Información de Preparadores	112
D.5. Registro de Documentos.	112
D.6. Consulta de Documentos.	113
D.7. Registro de Actas.	113
D.8. Consulta de Actas.	114

Agradecimientos

www.bdigital.ula.ve

Capítulo 1

Introducción

Los sistemas web han brindado una colaboración importante dentro de la sociedad y la educación, siendo las universidades una parte fundamental y necesaria en el proceso de comunicar y divulgar información, donde las aplicaciones web se convierten en un recurso imprescindible en nuestros días tanto para el aprendizaje como para difusión de información.

El Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes se perfila en afinidad con estos nuevos avances tecnológicos por medio del desarrollo de proyectos a través de plataformas web, que permitirán facilitar y satisfacer las necesidades de información de toda la comunidad universitaria.

Actualmente el Departamento de Computación presenta el problema de control y gestión de: las propuestas de proyectos de grado, manejo y divulgación de las agendas y actas de consejo de Departamento, información del personal así como también las noticias y documentos que forman parte de dicha dependencia ya que no se cuenta con un sistema automatizado que permitan un fácil y rápido manejo de dichos servicios, por lo que hace que sea un proceso lento y engorroso, es por eso que se plantea la creación de un sistema web que tiene como propósito agilizar todos esos procesos con el cual los estudiantes, profesores, secretarías y demás personal del departamento podrán tener acceso a esta información desde cualquier equipo con internet.

Para la elaboración de este proyecto se empleará como enfoque de desarrollo el método Watch [Montilva \(2004\)](#), para el desarrollo de aplicaciones empresariales. En

éste método se describen los procesos técnicos, gerenciales y de soporte que deben emplear los grupos de desarrollo de aplicaciones empresariales, las fases empleadas son cuatro, dedicadas a: modelo del producto, modelo del proceso, modelo del grupo de desarrollo.

Se modelara el sistema web con el Lenguaje unificado de modelado (UML) [UML \(2015\)](#), es un lenguaje que permite expresar de forma gráfica un sistema de una manera fácil y versátil gracias a que permite especificar, visualizar y documentar esquemas de sistemas de software orientados a objetos, lo que hace que cualquier persona lo pueda comprender, permite la obtención de un prototipo claro y explícito de las características del sistema en la etapa previa a su desarrollo. Todos los elementos gráficos sirven como documentación del sistema para futuras revisiones y mejoras del mismo.

Para la construcción de la base de datos se hará uso de un manejador de bases de datos NOSQL [Enríquez et al. \(2012\)](#). Estos sistemas de almacenamiento de dato no cumplen con el esquema entidad-relación, son adecuados para una escalabilidad realmente enorme, y tienden a utilizar modelos de consistencia relajados, no garantizando la consistencia de los datos, con el fin de lograr una mayor performance y disponibilidad; dicha base de datos se creara haciendo uso de MongoDB (MONGODB, 2007), este manejador de base de datos es conocido como una base de datos orientada a documento; esto quiere decir que en lugar de guardar los datos en registros, guarda los datos en documentos bajo el formato "*JSON Binario*" (*BSON*) [MongoDB \(2016b\)](#) representación binaria de "*JavaScript Object Notation*" (*JSON*) [json.org \(2013\)](#) , este manejador de bases de datos es de código abierto, lo que facilita el desarrollo y escalamiento.

1.1. Antecedentes

En el presente trabajo se expondrá una breve reseña de las investigaciones y trabajos realizados, que ha servido como soporte contextual a la hora de realizar el planteamiento de este proyecto, en este sentido se logró analizar los diferentes estudios realizados con anterioridad, encontrando notables aportes que dan base al desarrollo de este proyecto.

De los fundamentos teóricos revisados se plantearon como variables de estudio a: (Sistemas web para gestiones académicas, sistemas web con manejadores de bases de datos NoSQL).

Rodriguez (2008), “Sistema de Gestión Académica para el Personal Docente y de Investigación de La Universidad de Los Andes”. El Proyecto se elaboró con el objetivo de construir un sistema de información web de gestión académica para la Universidad de Los Andes, que soportara las actividades académicas, de extensión y de mejoramiento académico.

Segnini (2009), “Desarrollo de un sistema automatizado bajo entorno web para el control de la programación académica en la Universidad de Oriente núcleo de Anzoátegui”, proyecto que tenía como objetivo principal desarrollar un sistema basado en tecnología web, para el control de la programación académica de los estudiantes regulares de la Universidad de Oriente, Núcleo de Anzoátegui.

Guariman (2009), “Diseño de una aplicación web para la gestión en línea de los servicios académicos de una institución de educación superior”, el proyecto se elaboró con el propósito de diseñar una aplicación web para la gestión en línea de los servicios académicos de una institución de educación superior.

Márquez (2016), “Desarrollo de un sistema de recuperación multimedia basado en manejadores de bases de datos NoSQL”. Teniendo por objetivo recuperar de manera individual o completa la información textual y multimedia que poseen la tesis que han sido realizadas en el Departamento de Computación.

Linares (2016), “Sistema web para la gestión de proyectos de grado en la EISULA“. Teniendo por objetivo desarrollar un sistema web que permita la gestión y seguimiento de los proyectos de grados en EISULA.

1.2. Contexto de la Organización

La Universidad de Los Andes es una universidad pública y autónoma ubicada en los andes venezolanos con su sede principal y rectorado en la ciudad de Mérida; fundada por el clero como casa de estudios el 29 de marzo de 1785, elevada luego a seminario

y finalmente reconocida como Universidad el 21 de septiembre de 1810 bajo decreto expedido por la Junta Gubernativa de la provincia de la Corona de España.

La universidad está conformada por 11 facultades repartidas en el núcleo Mérida; el 15 de octubre de 1936, el Consejo Universitario acuerda crear la Facultad de Ingeniería, para el año de 1974-1975 se crea la Escuela de Sistemas, dicha entidad está dividida en tres departamentos: Investigación de Operaciones, Control y Automatización y el Departamento de Computación el cual tiene como misión formar profesionales integrales con habilidades, aptitudes, valores por el trabajo, capacidades y conocimientos teóricos prácticos sólidos en computación.

Tiene como objetivos brindar apoyo, gestión y control sobre las actividades académicas y administrativas de dicho departamento.

El Departamento de Computación está compuesto por (Véase la figura 1.1):

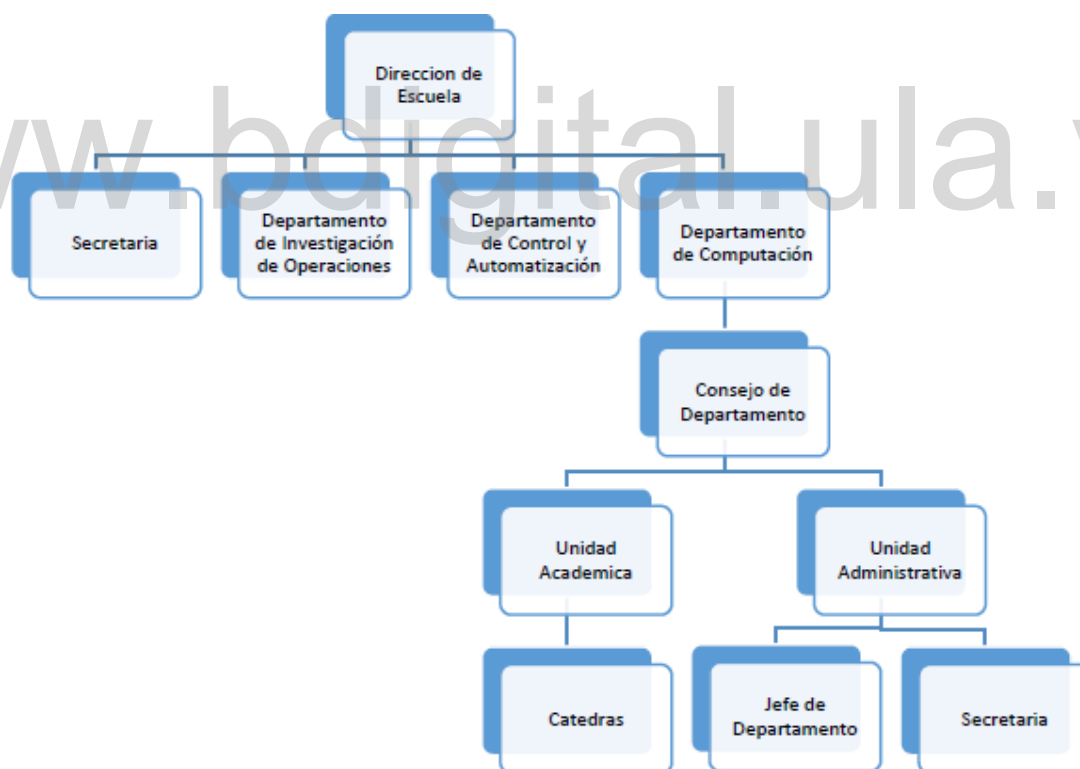


Figura 1.1: Estructura Organizativa de EISULA

1.3. Planteamiento del Problema

El Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes (EISULA), cuenta con una oficina de secretaría, dependencia que esta autorizada a recibir, registrar y dar a conocer la información asociada a dicho departamento, esta oficina a principio de cada semestre presenta una gran demanda por parte de los estudiantes que están próximos a inscribir sus temas de propuestas de proyectos de grado, además que durante todo el año académico se encarga de manejar información sobre actas, agendas, información de personal (profesores, preparadores y personal administrativo), noticias y documentos; por lo que es importante resaltar que el departamento carece de un recursos tecnológico como lo es un sistema web que proporcione un mejor desempeño en las tareas y servicios ofrecidos por esta oficina que permita controlar de manera efectiva y con mayor rapidez los procesos descritos anteriormente, el cual hace que el trabajo se torne un poco engorroso, generando un conjunto de inconvenientes como:

1. Manejo inadecuado de información.
2. Deficiencia en el control y manejo de información.
3. Demora en la ejecución de las actividades.

Es por esas razones que se consideró la creación de una herramienta tecnológica que pueda ayudar a conseguir un mejor control de las actividades que se llevan a cabo apoyando los procesos de recepción y registro, además de dar a conocer la información. Es por esto que se propuso el desarrollo de un sistema automatizado para el Departamento de Computación de EISULA a través del uso de bases de datos no relacionales.

1.4. Justificación

El desarrollo de un sistema de información web permitirá la obtención de beneficios y soluciones, en función de los problemas e inconvenientes presentes en la sección de secretaría del Departamento de Computación de (EISULA). El hecho de reformar los procedimientos utilizados actualmente para realizar sus actividades, representa un crecimiento importante en el servicio prestado por dicha oficina, ya que los

procesos serán sustituidos, por un sistema global con un entorno dinámico y de fácil manejo, acorde con las necesidades determinadas, logrando aumentar los procesos y el rendimiento del Departamento en general.

El desarrollo del sistema, permitirá optimizar el tiempo de ejecución de las labores, obtener mayor cantidad de información en menor tiempo. Con respecto a los beneficios que el sistema de información otorgará a EISULA, se considera que ésta obtendrá un mayor crecimiento, debido a los avances tecnológicos y mejoramiento de los procesos que se llevan a cabo en una de las entidades que la conforman.

1.5. Objetivos

1.5.1. Objetivo General

Desarrollar un Sistema Web que permita la automatización de los procesos referidos al manejo de las propuestas de proyectos de grado y administrativo que se llevan a cabo en el Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes.

1.5.2. Objetivos Específicos

- Analizar la plataforma actual en la que se encuentra el manejo de los procesos académicos y administrativos a fin de determinar las necesidades de automatización, comunicación de información que debe satisfacer el nuevo sistema.
- Modelar los distintos procesos que se llevan a cabo dentro del Departamento.
- Definir los requerimientos que permitan satisfacer las necesidades de los usuarios.
- Evaluar el rendimiento del manejador de base de datos MongoDB para aplicaciones con grandes cantidades de datos.
- Diseñar la Base de Datos no relacional del sistema propuesto.
- Diseñar las colecciones que formaran parte del sistema propuesto.

- Implementar el Sistema Web y poner a prueba cada uno de sus elementos.

1.6. Metodología

El sistema web planteado, utilizara como guía el método "*Watch*" [Montilva \(2004\)](#) para el desarrollo de aplicaciones empresariales, el uso de este método le agrega visibilidad al proyecto; pues permite que el grupo de desarrollo y los usuarios del sistema puedan conocer en qué estado se encuentra en cualquier momento, facilitando así al líder las labores de planificación y control, establece un marco metodológico único que estandariza el proceso de desarrollo y unifica la documentación que se produce a lo largo del proyecto, está fundamentada en modelos de procesos de ingeniería del software basada en componentes y además emplea las mejores prácticas, técnicas y notaciones utilizadas regularmente en la industria del software. Este método consta de los siguientes componentes:

1. **Modelo del Producto:** Describe el tipo de producto que el método "*Watch*" ayuda a producir y las características arquitectónicas generales de una aplicación.
2. **Modelo del Proceso:** Describe el conjunto de actividades que el grupo de desarrollo deberá seguir para el desarrollo de la aplicación.
3. **Modelo del Grupo de Desarrollo:** Describe el cómo el grupo de desarrollo debe estar organizado y cada uno de sus roles.

El modelado del sistema web, se realizara haciendo uso de "*Unified Modeling Language*" (*UML*) [UML \(2015\)](#), es un lenguaje gráfico desarrollado para el modelado de sistemas de software que permite expresar de forma gráfica un sistema de una manera fácil y versátil, para que cualquier persona lo pueda entender, expresar de manera explícita y clara las características de un sistema en la etapa previa a su construcción, y a la vez, permite que los modelos diseñados se pueda construir el sistema, además, todos los diagramas junto con sus elementos gráficos sirven como documentación y como utilidad para la revisión y evolución del sistema.

1.7. Alcance

Este proyecto será aplicado en el Departamento de Computación, específicamente para el uso administrativo por parte de: jefe de departamento, secretaria, profesores y graduandos de dicha dependencia, además de los usuarios generales que tendrán acceso a parte de la información que se encuentre almacenada en el sistema, y se delimitará en la construcción de un sistema de información para optimizar los procesos administrativos llevados a cabo en dicha sección.

1.8. Distribución de Capítulos

Capítulo 1, Introducción. Se describe al Departamento de Computación de La Escuela de Ingeniería de Sistemas, organización donde se realizara el proyecto. En este capítulo definiremos los antecedentes, el planteamiento del problema, la justificación, los objetivos, el alcance del proyecto y la metodología que se utilizara para su desarrollo.

Capítulo 2, Marco teórico. Presenta el marco teórico necesario para entender y emprender el desarrollo del proyecto de grado, el cual consta de información referente las tecnologías que serán usadas a lo largo del desarrollo proyecto, este método al ser iterativo se realizaron cinco iteraciones y para efecto del documento se describe la versión final.

Capítulo 3, Modelado y análisis de requerimientos. Comprende el desarrollo de las dos primeras fases del método "*Watch*", con el propósito de identificar y detallar el sistema de para el cual se desarrolla el sistema web y determinar las necesidades de información y automatización de los procesos que tienen los usuarios del sistema web en desarrollo.

Capítulo 4, Diseño del sistema. Se especifica la arquitectura de la aplicación, diseño de la interfaz, modelado de los componentes y módulos, es decir, describe el desarrollo del diseño arquitectónico del sistema, así como también, del diseño y aprovisionamiento de los componentes, a partir del conjunto de requerimientos

funcionales y no funcionales obtenidos en el capítulo anterior.

Capítulo 5, Implementación y pruebas del sistema. Mostrará la codificación de los módulos y componentes utilizados, *"script"* de la base de datos y ámbito de la aplicación, además de mostrar las pruebas de integración de todos los componentes y módulos, probándolos conjuntamente, verificando su funcionamiento.

Capítulo 6, Conclusiones y recomendaciones. Se describe las conclusiones generales del trabajo realizado y las recomendaciones que contribuyan a la mejora en el desarrollo de trabajos posteriores.

www.bdigital.ula.ve

Capítulo 2

Marco Teórico

En este capítulo, se describen los fundamentos teóricos necesarios para el entendimiento y comprensión del proyecto. Se define de manera detallada el método "Watch" para el desarrollo de aplicaciones empresariales, así como también, bases de datos no relacionales y el lenguaje de Modelado Unificado, utilizado para el desarrollo del sistema web. Formando así una base teórica, con la finalidad de tener una introducción a las herramientas de diseño y permitir al lector tener una idea clara sobre el contenido del documento.

2.1. Sistemas de Información

Según Barrios (2013) es un sistema que a través de la tecnología informática manipula un conjunto de datos (hechos o eventos organizacionales originados por las transacciones, entidades y/o procesos) y los convierte en la información requerida para apoyar la ejecución de actividades operativas y la toma de decisiones organizacionales.

2.1.1. Funcionamiento de los Sistemas de Información

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información. Barrios (2013)

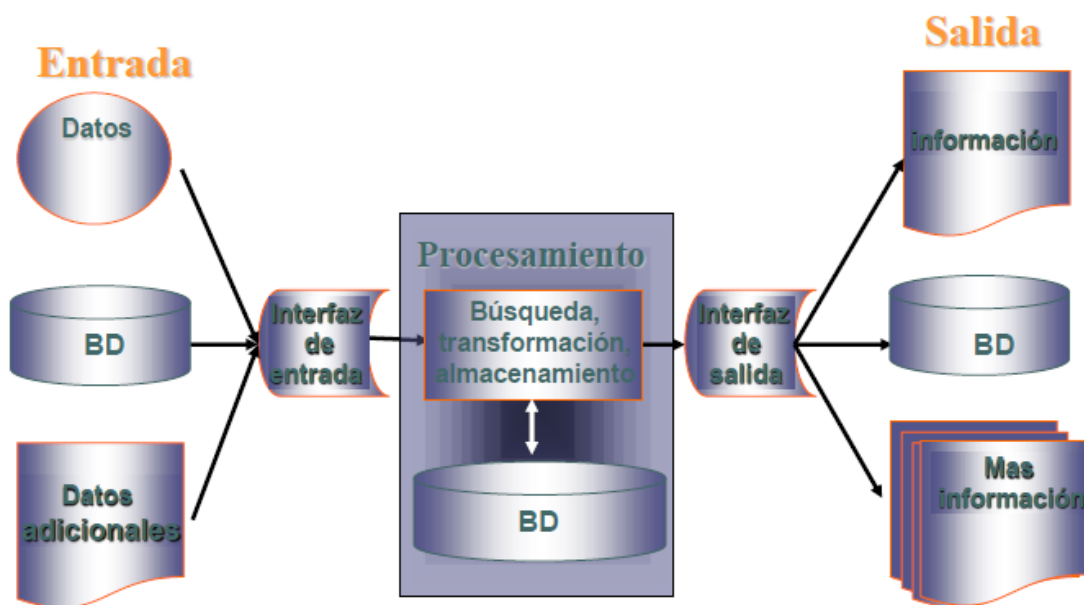


Figura 2.1: Funcionamiento de los Sistemas de Información.

2.1.2. Clasificación

Los sistemas de información cumplen tres objetivos básicos dentro de las organizaciones: Automatización de procesos operativos, proporcionar información que sirva de apoyo al proceso de toma de decisiones y lograr ventajas competitivas a través de su implantación y uso.

Los sistemas de información que logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente sistemas transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, etc. Por otra parte, los sistemas de información que apoyan el proceso de toma de decisiones son los sistemas de soporte a la toma de decisiones, sistemas para la toma de decisión de grupo, sistemas expertos de soporte a la toma de decisiones y sistema de información para ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos que cumplen, es el de los sistemas estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información. [Enciclopedia \(2016\)](#)

2.1.3. Sistemas de Información Web

Es un sistema de información que utiliza una arquitectura web para proporcionar información (datos) y funcionalidad (servicios) a usuarios finales, a través de una interfaz de usuario basada en presentación e interacción sobre dispositivos con capacidad de trabajar en la Web. En este orden de ideas los SIW (Sistemas de Información Web) varían ampliamente en su ámbito, desde sistemas de información, hasta sistemas de transacciones, incluso sistemas de servicios Web distribuidos. En virtud de la definición anterior los SIW se clasifican como sigue: Las Intranets, (las cuales dan apoyo al trabajo interno dentro de la Empresa) Los sitios de presencia en la Web, (los cuales son herramientas utilizadas para alcanzar consumidores fuera de la empresa).

Al igual que los SI tradicionales, más allá que una infraestructura para la entrega de información (en tiempo de ejecución), los SIW deben proporcionar una infraestructura de desarrollo y mantenimiento que permita manejar e interpretar los datos y que proporcione funcionalidades a los usuarios finales para capturar, almacenar, procesar y mostrar la información dando solución a sus requerimientos. [Muñoz \(2003\)](#)

2.2. Método *Watch*

El método "*Watch*" se enfoca particularmente en el desarrollo de aplicaciones empresariales. Una aplicación empresarial es aquella que apoya los procesos de negocio de una empresa de forma distribuida. Dos tipos de aplicaciones empresariales son las de comercio electrónico y sistemas de información web (SIW) que dan soporte a un o mas conjuntos de procesos de negocio, por medio de una interfaz que permite el intercambio de información a través de una red Intranet, Extranet o Internet.

Algunas de las características generales de las aplicaciones empresariales son [Montilva \(2004\)](#):

- *Están dirigidas a apoyar procesos de negocios de una empresa.*
- *Automatizan el flujo de trabajo de los procesos que ellas soportan.*

- *Establecen una clara diferencia entre: La presentación de la aplicación (interfaz U/S) La lógica del negocio (automatización del flujo de trabajo) La gestión de los datos (las bases de datos)*
- *Requieren integrarse a otras aplicaciones, tales como: Aplicaciones legadas, aplicaciones empaquetadas (ERP) Sistemas de información gerencial.*
- *Acceden a diferentes bases de datos locales o distribuidas*
- *Sus componentes están distribuidos en varios servidores de propósito específico*
- *La aplicaciones empresariales se instalan y corren sobre plataformas distribuidas compuestas por: Servidores web que se encargan del manejo de la capa de presentación de las aplicaciones empresariales Servidores de aplicaciones que manejan la lógica del negocio de las aplicaciones empresariales Servidores de datos que gestionan los datos de las aplicaciones empresariales*

Las aplicaciones empresariales siguen un patrón arquitectónico similar al denominado arquitectura 3 capas, donde la lógica de negocio se instala y se ejecuta separado del manejo de datos y de interfaz de usuario(Ver Figura A.1 en el anexo).

El método *WATCH* utiliza el paradigma de desarrollo basado en la reutilización de componentes de software. Con este paradigma, una aplicación empresarial posee una arquitectura de software de tres o más capas, en la que cada está compuesta de un conjunto de componentes de software interrelacionados(Ver Figura A.2 en el anexo)Montilva (2004).

- *La capa de presentación implementa la interfaz U/S de la aplicación empresarial. Está formada por dos tipos de componentes: Componentes del lado del cliente.- Son los componentes de la interfaz U/S que se instalan y corren en las máquinas clientes; por ejemplo, los applets en una aplicación web. Componentes del lado del servidor web.- Son los componentes de la interfaz U/S que se instalan en el servidor web. Por ejemplo, los componentes JSP y Servlets que implementan los aspectos dinámicos de la interfaz web.*

- *La capa de lógica de negocios implementa la funcionalidad de la aplicación empresarial. Está formada por dos tipos de componentes: Componentes de procesos.- Implementan las funciones que requieren los usuarios y automatizan los flujos de trabajo. Componentes de entidades de negocios (componentes de negocio).- Manejan los datos asociados a los objetos o entidades de negocio de la aplicación.*
- *La capa de datos se encarga de la administración de los datos de la aplicación. Está formada por: Una o más bases de datos o almacenes de datos XML que pueden ser locales o distribuidas.*

2.2.1. Modelo de Procesos WATCH

Es un marco metodológico que describe un conjunto de actividades necesarias para desarrollar una aplicación empresarial. El modelo *WATCH* organiza las actividades en procesos gerenciales y procesos de desarrollo que son 2 tipos de procesos diferentes pero complementarios. Los procesos gerenciales describen las actividades que la gerencia del proyecto debe realizar para [Montilva \(2004\)](#).

Los procesos gerenciales describen las actividades que la gerencia del proyecto (ó, líder del proyecto) debe realizar para:

- *Planificar, organizar, dirigir, manejar el grupo de desarrollo y controlar el proyecto de desarrollo de un sistema o aplicación empresarial.*
- *Asegurar la calidad del sistema.*
- *Gestionar la configuración del sistema*
- *Adiestrar el grupo de desarrollo durante el proceso de ejecución del proyecto.*

Los procesos de desarrollo son los procesos técnicos que describen que debe hacer el grupo de desarrollo para producir una aplicación empresarial. Estos procesos se organizan en una estructura jerárquica formada por fases, pasos y actividades.

La metodología es cíclica, controlada e iterativa, en cada ciclo los procesos de desarrollo producen nuevas versiones del sistema o subsistema, de haber errores se puede iterar entre las fases para corregir errores e introducir nuevos requisitos, en el centro de del marco metodológico se encuentra los procesos gerenciales que controlan y planifican los procesos de desarrollo(Ver Figura A.3 en el anexo). A continuación, se describen cada una de las ocho fases del marco metodológico que han de ser aplicadas por el grupo para desarrollar una aplicación empresarial [Montilva \(2004\)](#):

- *Fase 1: MODELADO DEL NEGOCIO* Obtener un conocimiento global y detallado del dominio de la aplicación empresarial; esto es, del sistema de negocios para el cual se desarrolla la aplicación. Este conocimiento se logra a través de un proceso de modelado empresarial que determina los objetivos, procesos, actores, objetos, reglas, eventos y unidades organizacionales del Sistema de Negocios (SN)(Ver Figura A.4 en el anexo).
- *Fase 2: INGENIERÍA DE REQUISITOS* Determinar las necesidades de información y automatización de procesos de negocios, que tienen los usuarios de la aplicación empresarial en desarrollo, mediante la definición y especificación de sus requisitos(Ver Figura A.5 en el anexo).
- *Fase 3: DISEÑO ARQUITECTÓNICO* Elaborar un diseño de la arquitectura de la aplicación empresarial que sea apropiada a los requisitos especificados y que establezca los subsistemas de la aplicación, los componentes de cada subsistema, las conexiones entre estos componentes y las restricciones que regulan la arquitectura(Ver Figura A.6 en el anexo).
- *Fase 4: DISEÑO DE COMPONENTES* Elaborar los diseños detallados de los componentes que integran cada uno de los subsistemas de la aplicación(Ver Figura A.7 en el anexo).
- *Fase 5: APROVISIONAMIENTO DE COMPONENTES*(Ver Figura A.8 en el anexo):
 - *Búsqueda y adaptación de componentes de software reutilizables que cumplan con las especificaciones de componentes*

- *Desarrollo de aquellos componentes que no puedan ser localizados o que no satisfagan adecuadamente las especificaciones de componentes*
- *Fase 6: ENSAMBLAJE DE COMPONENTES Implementar cada uno de las tres capas de la aplicación empresarial mediante el ensamblaje de componentes que fueron adquiridos, adaptados, suscritos o desarrollados en la Fase 5(Ver Figura A.9 en el anexo).*
- *Fase 7: PRUEBAS DE LA APLICACIÓN Comprobar que la aplicación empresarial cumple con los requisitos funcionales y no funcionales establecidos en la fase 2(Ver Figura A.10 en el anexo).*
- *Fase 8: ENTREGA DE LA APLICACIÓN EMPRESARIAL Instalar la aplicación empresarial en su ambiente de producción, ponerla en operación y entregarla a sus usuarios(Ver Figura A.11 en el anexo).*

El modelo del grupo de desarrollo de una aplicación empresarial puede estar organizado de diferentes maneras. Una de ellas es según una estructura jerárquica en la que los actores del proyecto se agrupan en base a las actividades del proceso de desarrollo de la aplicación. Otra manera de organizar el grupo es de acuerdo a las competencias requeridas para desarrollar aplicaciones empresariales.

2.3. UML

Los lenguajes de modelado permiten representar y comunicar conocimiento acerca de un sistema, "Unified Modeling Language", es un lenguaje gráfico de modelado de sistemas de software que integra y unifica diferentes notaciones y lenguajes formales, facilitando la representación del conocimiento acerca de un sistema y la comunicación de dicho conocimiento, es un estándar administrado por el consorcio OMG "Object Management Group".UML (2015)

Es un lenguaje gráfico para visualizar, especificar, diseñar, comunicar, construir y documentar sistemas de software y aplicaciones. Consta de un conjunto de notaciones

gráficas que son usadas para modelar la estructura, funcionalidad, comportamiento e implementación de un sistema y organizar los modelos producidos y de un lenguaje formal que permite expresar formalmente restricciones acerca de los elementos modelados de un sistema. UML nace de la unificación de tres metodologías de análisis y diseño orientada a objeto, la metodología de Grady Booch para la descripción de conjuntos de objetos y sus relaciones, la técnica de modelado orientada a objetos OMT: "*Object-Modeling Technique*" de James Rumbaugh y la aproximación de Ivar Jacobson OOSE: "*Object Oriented Software Engineering*" mediante la metodología de casos de uso. De estas tres metodologías de partida, las de Booch y Rumbaugh pueden ser descritas como centradas en objetos, ya que sus aproximaciones se enfocan hacia el modelado de los objetos que componen el sistema, su relación y colaboración. Por otro lado, la metodología de Jacobson es más centrada a usuario, ya que todo en su método se deriva de los escenarios de uso. [Guillen \(2003\)](#)

Diagramas de Casos de Uso Es una descripción de un proceso fin-a-fin relativamente largo que típicamente incluye varias etapas o transacciones, NO es una etapa o actividad individual de un proceso, describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario; permiten definir los límites del sistema y las relaciones entre el sistema y su entorno.

Diagrama de clases Son utilizados durante el proceso de análisis y diseño de sistemas informáticos, donde se crea el diseño conceptual de la información que se manejará en el sistema, estos diagramas son estructuras estáticas que dan una visión general del conjunto de clases existentes en el sistema modelado y las relaciones existentes entre cada una de ellas. Son estáticos porque muestra las relaciones entre las distintas clases, pero no especifica que sucede cuando ocurre alguna interacción entre ellas.

Diagrama de paquetes Un paquete es un mecanismo de agrupamiento usado para organizar los elementos modelados en UML y facilitar el manejo de los modelos de un sistema. Define un espacio de nombres. Dos paquetes diferentes pueden contener elementos con el mismo nombre, se puede contener otros paquetes y elementos de

modelado, sin límite de anidamiento, nos permiten dividir un modelo y reagrupar y encapsular los elementos de modelado.

Diagramas de actividades Son diagramas usados para elaborar modelos de flujos de trabajo (*workflow*) de un sistema, expresando que acciones se requieren, qué hacen estas acciones, cuando tienen lugar y donde estas acciones tienen lugar.

2.4. Bases de Datos

2.4.1. Base de Datos

Es una colección de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones. Una base de datos nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. [Européens \(1977\)](#)

Existen los Sistemas de Gestión de Base de Datos ("*DataBase Management System*") estos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

2.4.2. Bases de datos no relacionales

Es un término que se dio a conocer en el año 1998 por Carlos Strozzi y en el año 2009 tomo nuevamente auge, estas describe un conjunto de base de datos que difiere de lo modos de bases de datos tradicionales, es decir, no cumplen con el esquema de entidad-relación aun así son del tipo de almacenamiento estructurado; haciendo uso de otros formatos como clave-valor, columnas o grafos.

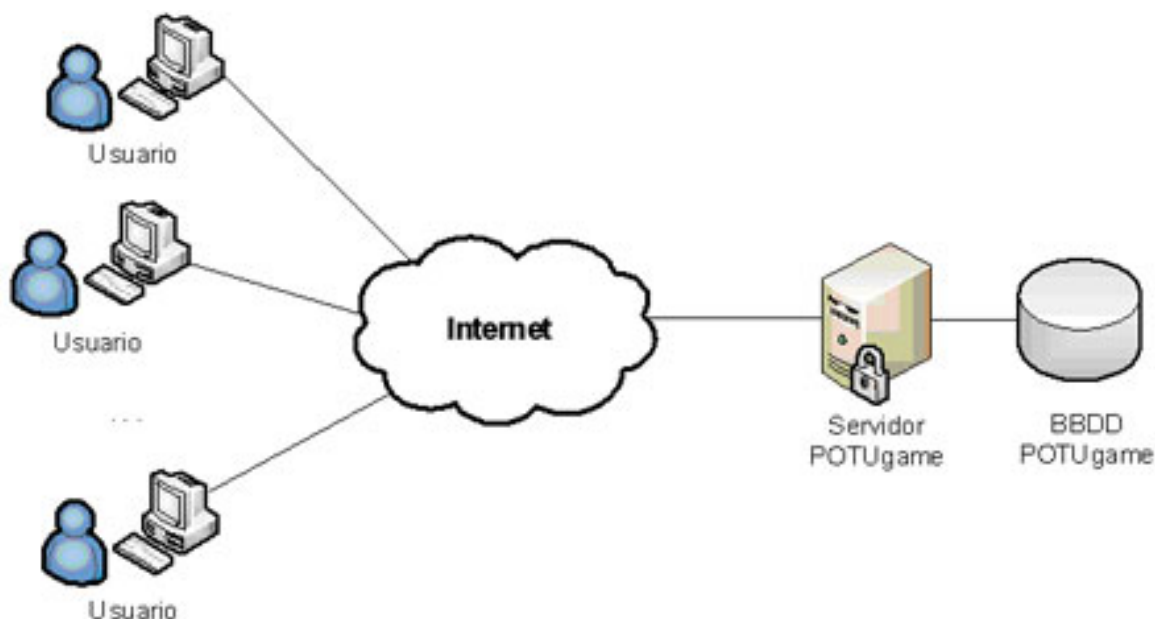


Figura 2.2: Estructura de Funcionamiento de una Base de Dato.

2.4.3. MongoDB

Es una base de datos ágil que permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta. Ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidados. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria. La replicación nativa de MongoDB y la tolerancia a fallos automática ofrece fiabilidad a nivel empresarial y flexibilidad operativa.

Organizaciones de todos los tamaños están usando MongoDB para crear nuevos tipos de aplicaciones, mejorar la experiencia del cliente, acelerar el tiempo de comercialización y reducir costes. [MongoDB \(2016a\)](#)

2.4.3.1. Cómo Funciona MongoDB

MongoDB en lugar de guardar los datos en registros, guarda los datos en documentos utilizando una codificación binaria de *"JavaScript Object Notation"* (JSON), llamada *"JSON Binario"* (BSON). El núcleo de la base de datos es capaz de interpretar su contenido, de modo que lo que a simple vista parece un contenido binario, realmente es un documento que contiene varios elementos. MongoDB permite la realización de operaciones de modificación de documentos enviando solamente el diferencial de datos, esto es, la modificación del mismo se realiza dentro del servidor, no en la parte cliente. MongoDB está escrito en C++, aunque las consultas se hacen pasando objetos JSON como parámetro. Es algo bastante lógico, dado que los propios documentos se almacenan en BSON. [MongoDB \(2016a\)](#)

2.5. Evaluación Y Comparación Del Rendimiento Del Manejador De Base De Datos MongoDB Para Aplicaciones Con Grandes Cantidades De Datos.

Por medio de la evaluación realizada por [Correa \(2013\)](#), se comprobó la velocidad de MongoDB con respecto a los motores de bases de datos PostgreSQL y MySQL, donde se utilizó una base de datos real de tuits que se recolectaron durante dos años con un bot en @cuxibamba que utiliza MySQL y para el momento contaba con 69.255 tuits y 12313 usuarios; para migrar la base de datos @cuxibamba desde MySQL a MongoDB se utilizó MongoHub que cuenta con una utilidad nativa para importar desde MySQL, para migrar desde PostgreSQL se utilizaron taps.

El hardware de la Pc donde se realizaron las pruebas es:

1. Procesador: 2.3 Ghz Intel Core i7
2. Memoria: 8GB 1600Mhz DDR3
3. HD: APPLE SSD SM256E Media (~500MB/s)

Inicio de la Base de Datos:

La base de datos se preparo tanto para PostgreSQL, MySQL y MongoDB, para realizar el conteo de tuits y usuarios pertenecientes a la cuenta @cuxibamba, como se observa en la figura 2.3

```
benchmark_1=# select count(*) from tweets;
count
-----
69255
(1 row)

benchmark_1=# select count(*) from users;
count
-----
12313
(1 row)
```

Figura 2.3: Preparación de la Base de Datos para el Manejador PostgreSQL.

```
mysql> select count(*) from tweets;
+-----+
| count(*) |
+-----+
|    69255 |
+-----+
1 row in set (0.01 sec)

mysql> select count(*) from users;
+-----+
| count(*) |
+-----+
|    12313 |
+-----+
1 row in set (0.00 sec)
```

Figura 2.4: Preparación Base de Datos para el Manejador de MySQL.

```
> db.tweets.count()
69255
> db.users.count()
12313
```

Figura 2.5: Preparación de Base de Dato para el Manejador MongoDB.

2.5.1. Primer Test

Se cuentan 10.000 veces los tuits y usuarios, obteniendo los resultados de la figura :2.6

```
cuxibamba:postgresql macool$ ruby count.rb
Rehearsal -----
count 10000 times  5.620000  0.390000  6.010000 (165.826635)
----- total: 6.010000sec

              user      system      total      real
count 10000 times  5.630000  0.390000  6.020000 (164.970709)
```

PostgreSQL count

```
cuxibamba:mysql macool$ ruby count.rb
Rehearsal -----
count 10000 times  6.250000  0.330000  6.580000 (151.418105)
----- total: 6.580000sec

              user      system      total      real
count 10000 times  6.290000  0.320000  6.610000 (151.667663)
```

MySQL count

```
cuxibamba:mongodb macool$ ruby count.rb
Rehearsal -----
count 10000 times  3.810000  0.300000  4.110000 ( 4.685105)
----- total: 4.110000sec

              user      system      total      real
count 10000 times  3.820000  0.290000  4.110000 ( 4.685353)
```

MongoDB count

Figura 2.6: Primer Test Conteo de Tuits y Usuarios.

1. PostgreSQL: 164.97 s.
2. MySQL: 151.67 s.
3. MongoDB: 4.69 s.

Con esto se concluye que el manejador MongoDB es 35 veces más rápido que PostgreSQL a la hora de realizar el conteo.

2.5.2. Segundo Test

"Select" buscando por id ("primary key") 1.000.000 veces (los ids son generados aleatoriamente por ruby entre 1 y 69255).

```
cuxibamba:postgresql macool$ ruby find.rb
Rehearsal -----
finding rand tweet 1000000 times 573.690000  29.830000 603.520000 (811.059294)
----- total: 603.520000sec

                                user      system      total      real
finding rand tweet 1000000 times 530.520000  28.530000 559.050000 (763.975478)
```

PostgreSQL find by id

```
cuxibamba:mysql macool$ ruby find.rb
Rehearsal -----
finding rand tweet 1000000 times 557.280000  21.470000 578.750000 (677.584838)
----- total: 578.750000sec

                                user      system      total      real
finding rand tweet 1000000 times 536.040000  20.640000 556.680000 (652.392665)
```

MySQL find by id

```
cuxibamba:mongodb macool$ ruby find.rb
Rehearsal -----
finding rand tweet 1000000 times 460.530000  27.210000 487.740000 (548.675625)
----- total: 487.740000sec

                                user      system      total      real
finding rand tweet 1000000 times 464.000000  27.400000 491.400000 (552.508299)
```

MongoDB find by id

Figura 2.7: Segundo Test "Select" buscando por id.

1. PostgreSQL: 763.98 s.
2. MySQL: 652.39 s.
3. MongoDB: 552.51 s.

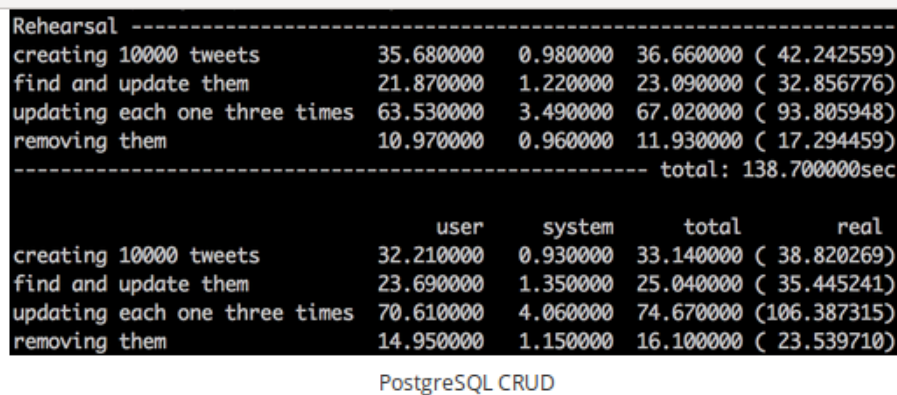
Por medio de estos resultados se observa que MongoDB nuevamente realiza el conteo mas rápido que como lo hacen los demás manejadores.

2.5.3. Tercer Test

Se realizo un CRUD basado en:

1. Crear 10000 (diez mil) tuits.
2. Encontrarlos (por su id único) y actualizar su texto (guardándolo en DB nuevamente).
3. Encontrar nuevamente los 10000 tuits y actualizar tres veces el contenido de cada uno, guardando en DB cada vez que se actualice. (Es decir, tres veces por tuit).
4. Encontrar cada uno de los 10000 tuits (por su id) y eliminarlo.

En las figuras 2.8 2.9 2.10 se observan los resultados obtenidos para cada uno de los manejadores.



```
Rehearsal -----
creating 10000 tweets      35.680000    0.980000    36.660000 ( 42.242559)
find and update them      21.870000    1.220000    23.090000 ( 32.856776)
updating each one three times 63.530000    3.490000    67.020000 ( 93.805948)
removing them             10.970000    0.960000    11.930000 ( 17.294459)
----- total: 138.700000sec

              user      system      total      real
creating 10000 tweets    32.210000    0.930000    33.140000 ( 38.820269)
find and update them     23.690000    1.350000    25.040000 ( 35.445241)
updating each one three times 70.610000    4.060000    74.670000 (106.387315)
removing them            14.950000    1.150000    16.100000 ( 23.539710)
```

PostgreSQL CRUD

Figura 2.8: Tercer Test Aplicado en un CRUD para PostgreSQL.

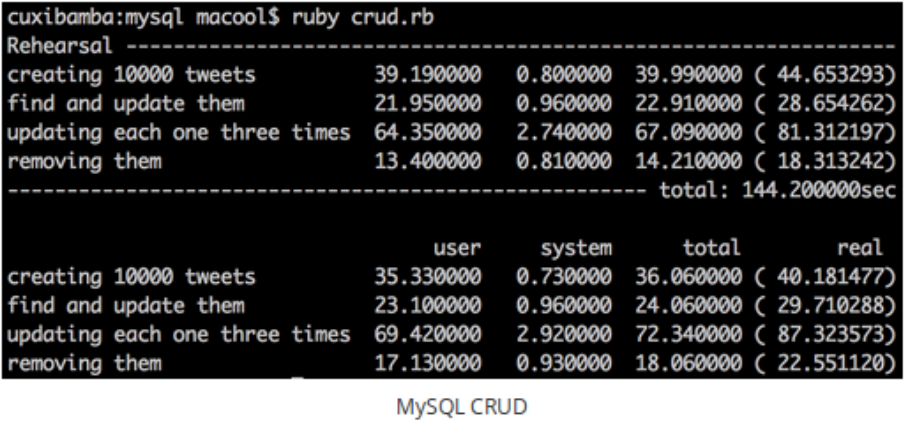


Figura 2.9: Tercer Test Aplicado en un CRUD para MySQL.

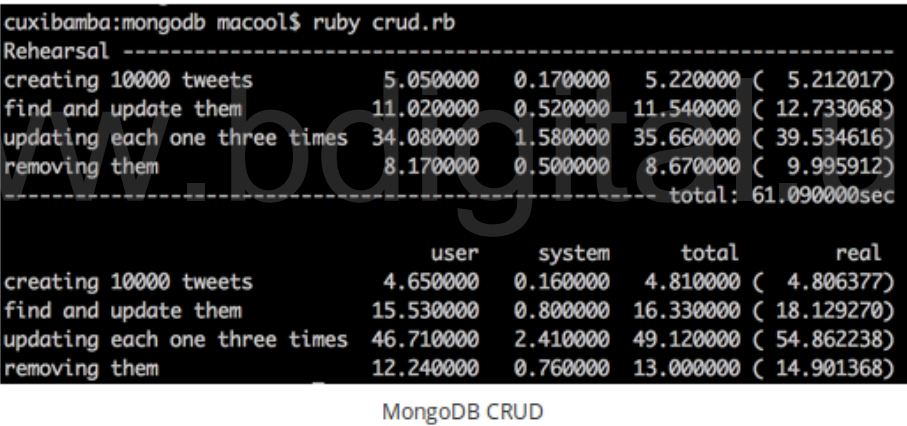


Figura 2.10: Tercer Test Aplicado en un CRUD para MongoDB.

Resultados obtenidos:

1.- PostgreSQL:

- 38.82 segundos para almacenar diez mil tuits
- 35.45 segundos para encontrar y actualizar diez mil tuits
- 106.39 segundos para encontrar los tuits y actualizar tres veces cada uno
- 23.54 segundos para eliminar los tuits

- Total: 204.2 segundos

2.- MySQL:

- 40.18 segundos para almacenar diez mil tuits
- 29.71 segundos para encontrar y actualizar diez mil tuits
- 87.32 segundos para encontrar los tuits y actualizar tres veces cada uno
- 22.55 segundos para eliminar los diez mil tuits
- Total: 179.76 segundos

3.-MongoDB:

- 4.81 segundos para almacenar diez mil tuits
- 18.13 segundos para encontrar y actualizar diez mil tuits
- 54.86 segundos para encontrar los tuits y actualizar tres veces cada uno
- 14.9 segundos para eliminar diez mil tuits
- Total: 92.7 segundos (casi la mitad de MySQL, menos de la mitad de PostgreSQL)

Como conclusión se puede observar que en los tres test que fueron aplicados, MongoDB tuvo un rendimiento mayor en comparación a los demás manejadores de datos, además se puede concluir que MySQL es mas rápido que PostgreSQL.

Se decide hacer uso de MongoDB porque permite almacenar datos semiestructurados, además cuando se manejan Big Data es mejor utilizar MongoDB ya que permite escalar horizontalmente, es decir, tener varias máquinas conectadas a tener una sola máquina, lo que ayuda a disminuir costos.

2.6. Modelo Vista Controlador (MVC)

El Modelo Vista Controlador es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la

necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos, es una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales

Esta basado en la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores.

2.6.1. Modelo

Es la capa donde se trabaja con los datos, por tanto contendrá mecanismos para acceder y actualizar la información. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas o colecciones y harán los correspondientes *selects*, *updates*, *inserts*.

2.6.2. Vistas

Contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

2.6.3. Controlador

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación.

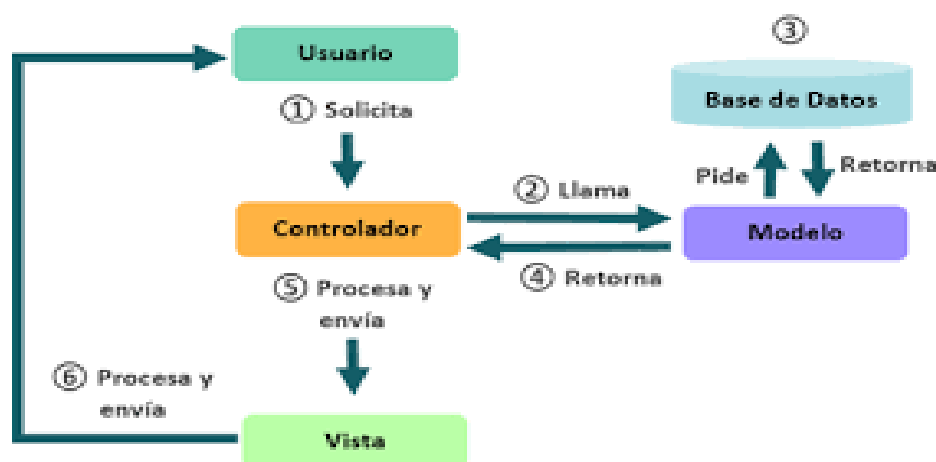


Figura 2.11: Diagrama MVC

2.7. Express

Un *"framework"* es un entorno o ambiente de trabajo para desarrollo; dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo de aplicaciones como el soporte de programa, bibliotecas, plantillas y más, estos ofrecen módulos que originan soluciones rápidas para el desarrollo, manteniendo el código organizado, actualizaciones sencillas de realizar y nos ahorra tiempo.

Express.js es un *"framework"* de desarrollo de aplicaciones web para Node.js, diseñado para la construcción de páginas individuales, de varias páginas y aplicaciones web híbridas. Su creador, TJ Holowaychuk, lo describió como un servidor inspirado en Sinatra, lo que significa que es relativamente mínimo con muchas características disponibles como *"plugins"*. Es robusto, rápido, flexible y muy simple. Entre otras características, ofrece *"Router"* de URL (*Get*, *Post*, *Put*) y facilidades para motores de plantillas (Jade, EJS, JinJS). [StrongLoop \(2016\)](#)

Capítulo 3

Modelado Y Análisis De Requerimientos

Este capítulo comprende el desarrollo de las dos primeras fases del método *Watch*, con el propósito de realizar un análisis que permita determinar las necesidades de información y automatización de los procesos que tienen los usuarios del sistema web en desarrollo.

3.1. Modelado de Negocios

El sistema de negocios está basado en el conjunto de actividades que el jefe de departamento, secretaria, profesores y estudiantes adscritos al Departamento de Computación de EISULA deben realizar para llevar a cabo el control y registro de la información relacionadas con cada una de las tareas que debe cumplir en su respectivo rol de trabajo, para que así las personas que hacen vida o fuera de dicho departamento puedan obtener información de una manera mas eficaz, rápida y accesible.

3.1.1. Visión del Sistema de Negocios

Consolidar un sistema web que forme parte del Departamento de Computación como una aplicación que permita sistematizar el registro y control de la misma por medio de la gestión de: usuarios, actas, agendas, noticias, documentos, personal y

graduando, lo que permitirá agilizar y facilitar la información entre sus usuarios directos e indirectos que sean de beneficio mutuo.

3.1.2. Misión del Sistema de Negocios

Satisfacer las necesidades de los usuarios por medio del desarrollo de una aplicación web, que impulse y promueva el crecimiento tecnológico de la Escuela de Sistemas, convirtiéndola en una referencia en cuanto a la sistematización y automatización de la información que este asociada a ella.

3.2. Identificación de Actores del Sistema

Los actores identificados en el sistema son los siguientes:

Jefe de Departamento: Su función es la de registrar a los usuarios del sistema y de editar y/o eliminar algún usuario.

Secretaria: Se encarga de registrar y editar (graduando y profesores), así como también el registro de: actas, agendas, personal adscrito al departamento, de igual manera publicar las noticias, documentos e información, además de realizar consultas y edición sobre información registrada.

Profesores: Se encarga de registrar los datos asociados a información personal, así como también el registro del currículum vitae y dar a conocer la disponibilidad de proyectos de grado bajo su tutoría.

Graduando: Su función es registrar en el sistema su información personal, y la información relacionada con la propuesta de proyecto de grado que está desarrollando.

Usuario General: Es un actor externo al sistema, teniendo permiso solo para hacer consultas relacionadas con: propuestas de proyectos de grado, mientras que para agendas, actas, información de profesores y preparadores podrá hacer consultas y descarga.

3.2.1. Jerarquía de actores

Ya definido el sistema de negocio y los actores involucrados en él es importante conocer cual es su jerarquía dentro de la organización, en la tabla 3.3.1, se lista cada uno de estos actores con la descripción asociada los procesos que realiza dentro del sistema de negocios.

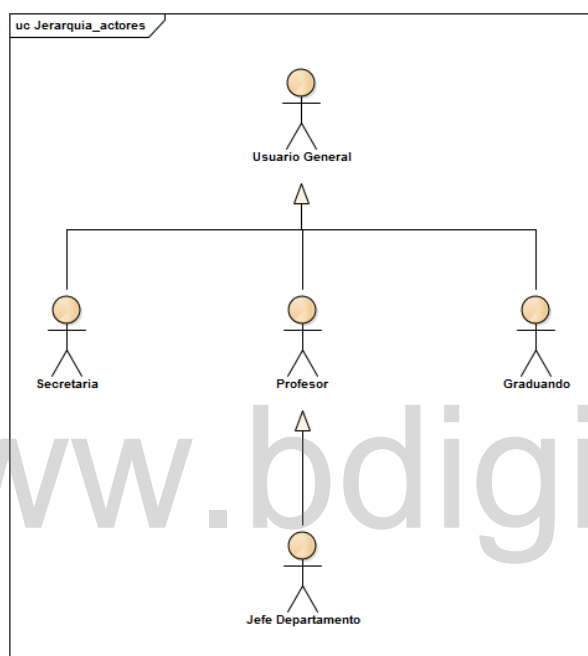


Figura 3.1: Jerarquía de Actores.

3.3. Modelado de los Procesos del Sistema

En la figura 3.2 se describen los procesos que están involucrados en el sistema de negocios por medio de la cadena valor, es decir, muestra la relación entre los procesos principales y los procesos que sirven de apoyo a la organización.

Gestión de Usuarios: Es el proceso encargado de crear, actualizar, y visualizar todo lo referente a la cuenta de un usuario.

Gestión de Proyectos de Grado: Es el proceso encargado de llevar un control de la información de las Propuestas y Proyectos de Grado de los estudiantes del Departamento de Computación.

Gestión de Personal: Es el proceso que se encarga de llevar el control del personal administrativo y académico que forma parte del Departamento.

Gestión de Actas y Agendas: Se encarga de realizar el manejo y control de los documentos e información asociada a las Actas y Agendas.

Gestión de Documentos y Noticias: Este proceso se encarga de llevar el control de los Documentos y Noticias relacionadas al Departamento o la Escuela.

Además de contar con procesos de apoyo como lo son: Gestión Administrativa y Gestión Académica.

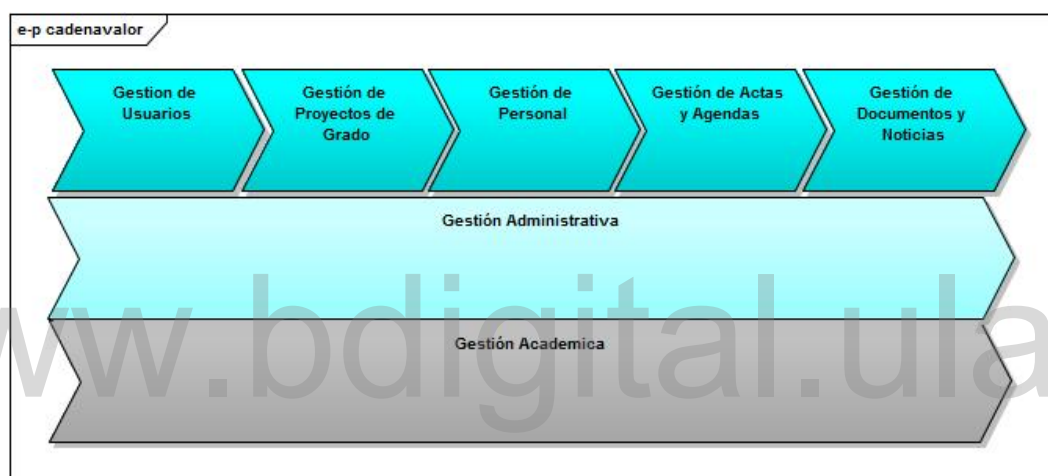


Figura 3.2: Modelo de los Procesos del Sistema de Negocio.

3.3.1. Descripción de los Procesos de Negocio Asociados a los Actores del Sistema

En la tabla 3.3.1 se observan los procesos que están involucrados en el sistema de negocios, es decir, muestra las actividades que debe realizar cada uno de los usuarios del sistema a la hora de interactuar con la aplicación.

Actor	Proceso	Descripción
Jefe de Departamento	Registro Usuarios	Registra los usuarios que tendrán acceso al sistema.

	Consultar Usuarios	Permite verificar todos los usuarios que se encuentran almacenados en la base de datos.
	Eliminar Usuarios	Permite eliminar los usuarios que ya no pertenecen al departamento .
	Registro de Currículum	Registra la información y el archivo de currículum vitae de un profesora.
	Registro de Temas de Proyectos de Grado	Registra en el sistema los temas de proyectos que el profesor ofrezca para desarrollo.
	Consulta de Currículum	Permite visualizar el currículum de un profesor luego de haber sido almacenado en la base de datos .
	Consulta de Temas de Proyectos de Grado	Luego que el profesor registra los temas en la base de datos, permite visualizar que temas están propuestos por dicho profesor.
	Eliminar Currículum	Borra de la base de datos el currículum de un determinado profesor.
	Eliminar Temas de Proyectos de Grado	Elimina de la base de datos los proyectos que ya hayan sido realizados .

Secretaria	Registro Usuario	Registra los usuarios que tendrán acceso al sistema.
	Registro Agendas	Registra la información asociada a la agenda, así como también el archivo donde está el cronograma de los puntos a tratar en el consejo de Departamento.
	Registro Actas	Registra la información asociada al acta que haya sido levantada luego de la discusión de la agenda así como también el archivo donde están las decisiones tomadas para cada punto tratado en el consejo de Departamento.
	Registro de Noticias y documentos	Registra las noticias y documentos asociados al Departamento.
	Registro de Personal	Permite observar cuales serán los puntos a tratar en la reunión de consejo de Departamento .
	Consulta Agendas	Toma en cuenta los movimientos que tiene la caja de ahorros en periodos mensuales o anuales, para así generar un reporte.

	Consulta Actas	Permite tener la información de las decisiones tomadas en el consejo de cada uno de los puntos expuestos en la agenda.
	Consulta de Preparadores	Permite conocer la información relacionada a los preparadores que dictan materias dentro del departamento.
	Consulta de Noticias y Documentos	Permite visualizar todas las noticias y documentos pertenecientes o relacionadas con actividades que involucren al departamento.
	Consulta de Personal	Permite dar a conocer quiénes es el personal del departamento.
	Eliminar Personal	Elimina de la base de datos a los trabajadores, profesores o preparadores que hayan abandonado el cargo o que se hayan graduado respectivamente.
	Eliminar Noticias y Documentos	Borra de la base de datos documentos y noticias que se encuentren desactualizadas
Profesor	Registro de Currículum	Registra la información y el archivo de currículum vitae de un profesora.

	Registro de Temas de Proyectos de Grado	Registra en el sistema los temas de proyectos que el profesor ofrezca para desarrollo.
	Consulta de Currículum	Permite visualizar el currículum de un profesor luego de haber sido almacenado en la base de datos .
	Consulta de Temas de Proyectos de Grado	Luego que el profesor registra los temas en la base de datos, permite visualizar que temas están propuestos por dicho profesor.
	Eliminar Currículum	Borra de la base de datos el currículum de un determinado profesor.
	Eliminar Temas de Proyectos de Grado	Elimina de la base de datos los proyectos que ya hayan sido realizados .
Graduando	Registro de Propuesta y/o Proyectos de Grado	Registra la información personal y la información relacionada a su propuesta de grado.
	Consulta de Propuesta y/o Proyectos de Grado	Permite visualizar la información almacenada en la base de datos sobre la propuesta o los proyectos de grado.

	Editar Datos Personales	Modifica información referente a información personal.
Usuario General	Consultar Información General del Sistema.	Realizar consultas sobre las actas, agendas, noticias, documentos, personal y graduando.

Tabla 3.1: Diagramas de actividades realizadas por los actores del sistema

3.3.2. Reglas de Negocio

Las reglas de negocio que regulan el sistema son:

1. Ser estudiante del décimo semestre de alguno de los Departamentos de la Escuela de Ingeniería de Sistemas.
2. Haber cursado las materias pertenecientes al noveno semestre de la Escuela de Ingeniería de Sistemas de acuerdo al departamento donde esté inscrito.
3. La propuesta y proyecto de grado deberá estar adscrito al Departamento de Sistemas Computacionales.
4. Ser personal adscrito al Departamento como: Jefe de Departamento, secretaria, profesor, preparador, técnico o beca-trabajo asignado.
5. Se rige por las normativas del reglamento correspondiente a las actas y agendas que deben manejarse en el Departamento.

3.4. Definición y Especificación de Requisitos

A partir de los procesos mostrados en la tabla 3.3.1 donde están especificados cada uno de los procesos que desempeñan los actores dentro del entorno del sistema, surgen

los requerimientos que debe satisfacer la aplicación. Estos requerimientos establecen todas las funciones y procesos que debe realizar el sistema para cumplir con los objetivos planteados.

3.4.1. Requisitos Funcionales

En esta fase se describe el comportamiento del sistema, a través del estudio de los actores que interactúan con la aplicación. A continuación se expone la lista de requerimientos funcionales:

1. Crear cuentas de usuarios para los administradores del sistemas.
2. Almacenar información de los datos personales y propuestas de proyectos de grado de los estudiantes del décimo semestre del Departamento de Computación.
3. Registrar el currículum de los docentes adscritos al Departamento.
4. Almacenar la información del personal administrativo, técnicos, beca-trabajos y preparadores.
5. Almacenar la información de las actas y agendas discutidas por el Departamento de Computación.
6. Almacenar noticias, documentos e información asociadas con el Departamento de Computación.
7. Visualizar el listado de los estudiantes, profesores, preparadores, técnicos, beca-trabajos, actas y agendas y noticias pertenecientes al Departamento de Computación.
8. Descargar del sistema: actas, agendas y currículum de profesores, proyectos de grado.
9. Editar usuarios, cambiar de estatus a los profesores, modificar noticias e información.
10. Eliminar usuarios e información del sistema.

3.4.2. Requisitos No Funcionales

Los requisitos no funcionales describen el ambiente de operación, la interfaz del usuario y el ambiente de operación del sistema. Identifican las restricciones en cuanto a seguridad y calidad del software que se utiliza para desarrollar la aplicación.

Servidor

- Procesador Dual Core de 2.2 GHz.
- Disco duro de 8 GB o superior.
- Memoria RAM de 256 MB o superior.
- Tarjetas de red.
- Servidor web Nginx.

Cliente

- Conexión desde cualquier dispositivo con acceso.

Plataforma

- Sistema operativo Linux distribución Debían para el servidor.
- Servidor Nginx. 4. Sistema manejador de base de datos NoSQL, MongoDB.
- Para la elaboración de las páginas web, el gestor de plantillas de Node.js, Jade.
- La conexión entre MongoDB y HTML se da por medio de Node.js, usando la librería mongoose y el gestor de plantillas Jade.
- El sistema de recuperación de información multimedia puede ser usado desde cualquier explorador de internet (Chrome, Firefox, Opera, entre otros).
- Para visualizar los archivos Pdf es necesario tener instalado: Adobe Acrobat Reader, Foxit Reader o cualquier otro visor de documentos Pdf.

3.5. Casos de Uso

3.5.1. Definición de los Casos de Uso

Los casos de uso son una técnica de diagramas UML, que permiten la captura de los requerimientos elementales del sistema, además proporcionan los escenarios donde se indica como debe comportar el sistema a la hora de que los usuarios interactúen con el sistema.

Caso de Uso	Descripción	Requerimiento
CU1	Inicio de Sesión	R1
CU2	Gestionar Usuario	R1
CU3	Gestionar Propuesta de Proyecto de Grado	R2
CU4	Gestionar Currículum Profesor	R3
CU5	Gestionar Actas y Agendas	R5
CU6	Gestionar Personal	R4
CU7	Gestionar Noticias y Documentos	R6
CU8	Consultar Propuesta de Proyecto de grado	R8
CU9	Consultar Personal	R7
CU10	Consultar Noticias y Documentos	R7
CU11	Editar Propuesta de Proyecto de Grado	R9
CU12	Editar noticias o Documentos	R9

Tabla 3.2: Actividades realizadas por los actores del sistema

3.5.2. Descripción de los Casos de Uso

La descripción de los casos de uso se presenta desde la Tabla 3.3 hasta la Tabla 3.9 y en el anexo B.2, donde se describen los actores principales que llevan a cabo el caso de uso, las precondiciones y poscondiciones, los flujos básicos y los alternativos de cada uno de los casos de uso existentes.

Caso de Uso:	CU1. Inicio de Sesión
Actores Principales:	Jefe de Departamento, Secretaria, Profesor y Graduando.
Condiciones de Entrada:	Estar registrado en el Sistema.
Condiciones de Salida:	El actor ingresa al sistema exitosamente.
Flujo de Eventos:	1. El actor selecciona la pestaña Entrar 2. El sistema muestra una ventana donde se solicitan los datos que el usuario debe ingresar. 3. El actor ingresa sus datos y selecciona el botón Entrar.
Flujos Alternativos:	3.a El actor ingresa un dato equivocado. 3.b El usuario debe ingresar nuevamente los datos. 4. El actor olvido su clave y selecciona recuperar clave. 5. El sistema solicitara correo para enviarle su clave.
Notas:	Si el Actor no estar registrado debe seleccionar el botón registrar.

Tabla 3.3: CU1. Inicio de Sesión

Caso de Uso:	CU3. Gestionar Propuesta de Proyecto de Grado
Actores Principales:	Secretaria, Graduando.
Condiciones de Entrada:	La secretaria debe haber aperturado la cuenta de usuario del graduando. El usuario graduando debe haber iniciado sesión.

Condiciones de Salida:	Carga correcta de la propuesta de proyecto de grado.
Flujo de Eventos:	1. El sistema mostrara una ventana donde solicite los datos para el ingreso al sistema. 2. El graduando deberá seleccionar el menú desplegable cargar en sistema y seleccionar propuesta de proyecto de grado. 3. El usuario debe ingresar los datos solicitados. 4. El sistema valida que todos los datos requeridos llenos y valida q los datos sean correctos 5. Seleccionar los PDF desde las carpetas en donde se encuentren localizadas. 6. Hacer clic sobre la palabra guardar 7. Esperar que el sistema actualice.
Flujos Alternativos:	4a.- Si algún campo obligatorio queda vacío; el sistema en envía un mensaje de “Error debe ingresar todos los datos obligatorios solicitados” 4b.- Si alguno de los documentos y/o datos ingresados no son correctos el sistema envía un mensaje “Error los documentos y/o datos deben tener el formato correcto”

Tabla 3.4: CU3. Gestionar Propuesta de Proyecto de Grado

Caso de Uso:	CU8. Consultar Propuesta de Proyecto de Grado
---------------------	--

Actores Principales:	Usuario general, Graduando, Secretaria, Profesores, Jefe de Departamento.
Condiciones de Entrada:	Debe haber propuestas de proyectos de grado almacenadas en el sistema.
Condiciones de Salida:	Hallar satisfactoriamente la información solicitada al sistema.
Flujo de Eventos:	1.-El sistema mostrara una ventana donde se encontraran todas la propuestas de proyectos de grado, donde los usuarios podrán elegir qué propuesta visualizar. 2.- Elegir la propuesta de proyecto de grado que desea consultar.
Flujos Alternativos:	2.a Si no se encontró información sobre la consulta realizada el usuario deberá corroborar su búsqueda.

Tabla 3.5: CU8. Consultar Propuesta de Proyecto de Grado

Caso de Uso:	CU11. Editar Propuesta de Proyecto de Grado
Actores Principales:	Graduando, secretaria
Condiciones de Entrada:	El usuario graduando debe haber iniciado sesión.
Condiciones de Salida:	Modificación correcta de los nuevos datos introducidos.

Flujo de Eventos:	1. El sistema mostrara una ventana donde se encuentran todos los datos 2. Hacer clic sobre el botón editar. 3. Introducir los nuevos datos. 4. Hacer clic sobre el botón guardar para que el sistema se encargue de almacenar los datos introducidos. 5. El sistema se encarga de autenticar los datos. 6. Si los datos son válidos el sistema mostrara nuevamente la ventana principal donde están los datos editados anteriormente.
Flujos Alternativos:	Si quien desea editar es la secretaria el flujo es: 3. Introducir los nuevos datos y/o documentos 4. Hacer clic sobre el botón guardar para que el sistema se encargue de almacenar los datos introducidos. 5. El sistema se encarga de autenticar los datos. 6. Si los datos son válidos el sistema mostrara nuevamente la ventana principal donde están los datos editados anteriormente.

Tabla 3.6: CU11. Editar Propuesta de Proyecto de Grado

Caso de Uso:	CU5. Gestionar Noticias o Documentos
Actores Principales:	Secretaria.
Condiciones de Entrada:	El usuario secretaria debe haber iniciado sesión.

Condiciones de Salida:	Carga correcta de las noticias o documentos.
Flujo de Eventos:	1. El sistema mostrara una ventana donde solicite los datos para el ingreso al sistema. 2. La secretaria deberá seleccionar el menú desplegable cargar en sistema y seleccionar la opción noticias o documentos. 3. El usuario debe ingresar los datos solicitados. 4. El sistema valida que todos los datos requeridos llenos y valida q los datos sean correctos 5. Seleccionar los PDF desde las carpetas en donde se encuentren localizadas. 6. Hacer clic sobre la palabra guardar 7. Esperar que el sistema actualice.
Flujos Alternativos:	4a.- Si algún campo obligatorio queda vacío; el sistema en envía un mensaje de “Error debe ingresar todos los datos obligatorios solicitados” 4b.- Si alguno de los documentos y/o datos ingresados no son correctos el sistema envía un mensaje “Error los documentos y/o datos deben tener el formato correcto”.

Tabla 3.7: CU5. Gestionar Noticias o Documentos

Caso de Uso:	CU10. Consultar Noticias o Documentos
---------------------	--

Actores Principales:	Usuario general, Graduando, Secretaria, Profesores, Jefe de Departamento.
Condiciones de Entrada:	Debe haber propuestas de proyectos de grado almacenadas en el sistema.
Condiciones de Salida:	Hallar satisfactoriamente la información solicitada al sistema.
Flujo de Eventos:	1.-El sistema mostrara una ventana donde se encontraran todas la propuestas de proyectos de grado, donde los usuarios podrán elegir qué propuesta visualizar. 2.- Elegir la propuesta de proyecto de grado que desea consultar.
Flujos Alternativos:	2.a Si no se encontró información sobre la consulta realizada el usuario deberá corroborar su búsqueda.

Tabla 3.8: CU10. Consultar Noticias o Documentos

Caso de Uso:	CU12. Editar Noticias o Documentos
Actores Principales:	Secretaria.
Condiciones de Entrada:	El usuario secretaria debe haber iniciado sesión.
Condiciones de Salida:	Modificación correcta de los nuevos datos introducidos.

Flujo de Eventos:	1. El sistema mostrara una ventana donde se encontraran todos los datos asociados con las noticias o documentos. 2. Elegir el campo que desea editar. 3. Hacer clic sobre el botón editar. 4. Introducir los nuevos datos. 5. Hacer clic sobre el botón guardar para que el sistema se encargue de almacenar los datos introducidos. 6. El sistema se encargara de autenticar los datos. 7. Si los datos son válidos el sistema mostrara nuevamente la ventana principal donde están los documentos editados anteriormente.
Flujos Alternativos:	4a. Si la secretaria desea modificar el PDF de la noticia o documento 4a.1. Elegir del computador o cualquier dispositivo de almacenamiento el nuevo documento PDF . 4a.2. Dar clic en el botón guardar 4a.3. Esperar que el nuevo documento cargue y el sistema se actualice.

Tabla 3.9: CU12. Editar Noticias o Documentos

3.5.3. Diagramas de Casos de Uso

Para una mejor comprensión de los casos de uso descritos anteriormente, desde la Figura 3.3 hasta la Figura 3.7 y en el Anexo B.1 se muestran los diagramas de casos de

uso para cada uno de los actores definidos en la tabla de 3.3.1

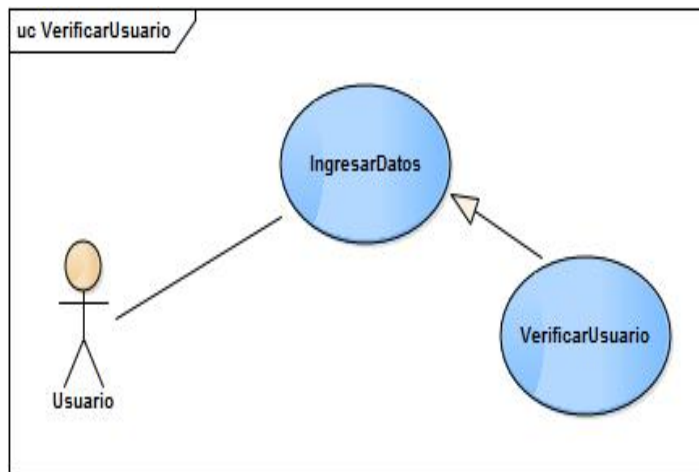


Figura 3.3: Autenticación de Usuarios.

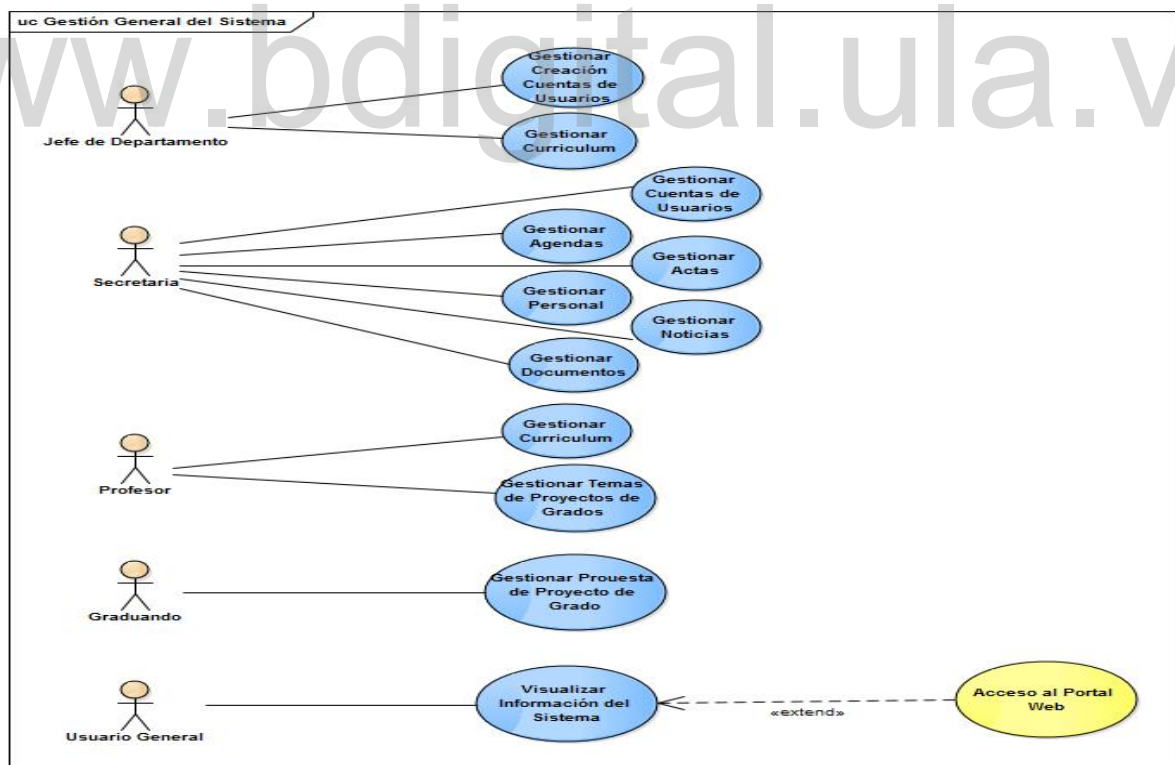


Figura 3.4: Diagrama Caso de Uso Gestión General del Sistema.

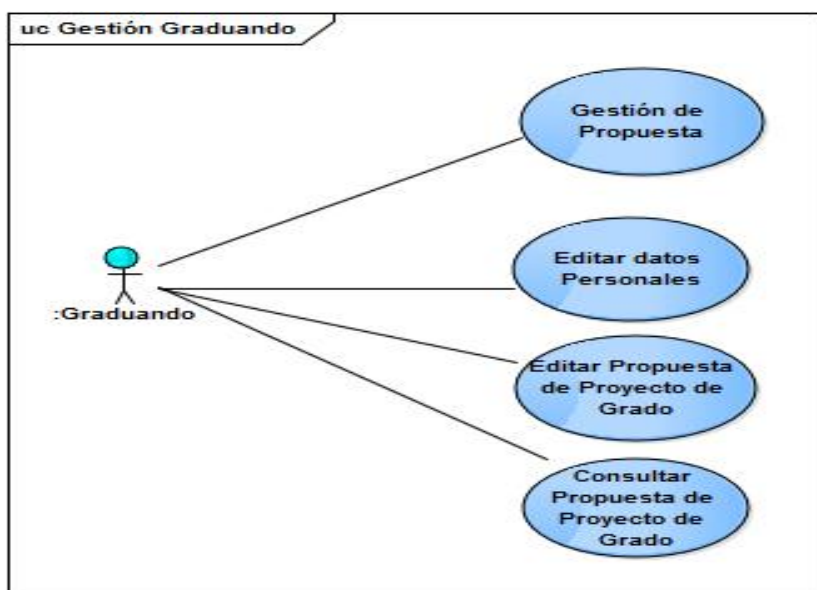


Figura 3.5: Diagrama Caso de Uso Gestión Graduando.

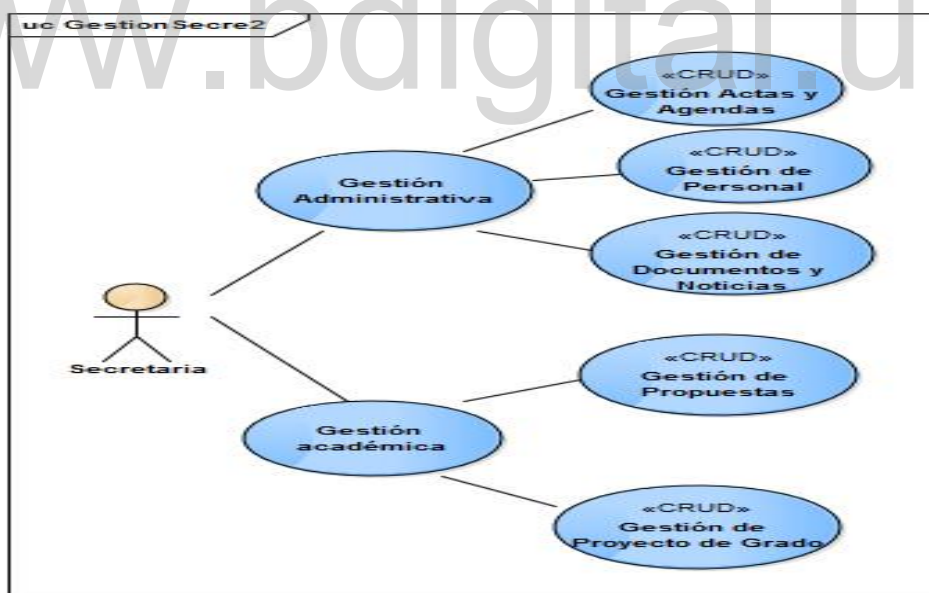


Figura 3.6: Diagrama Caso de Uso Gestión Secretaria.

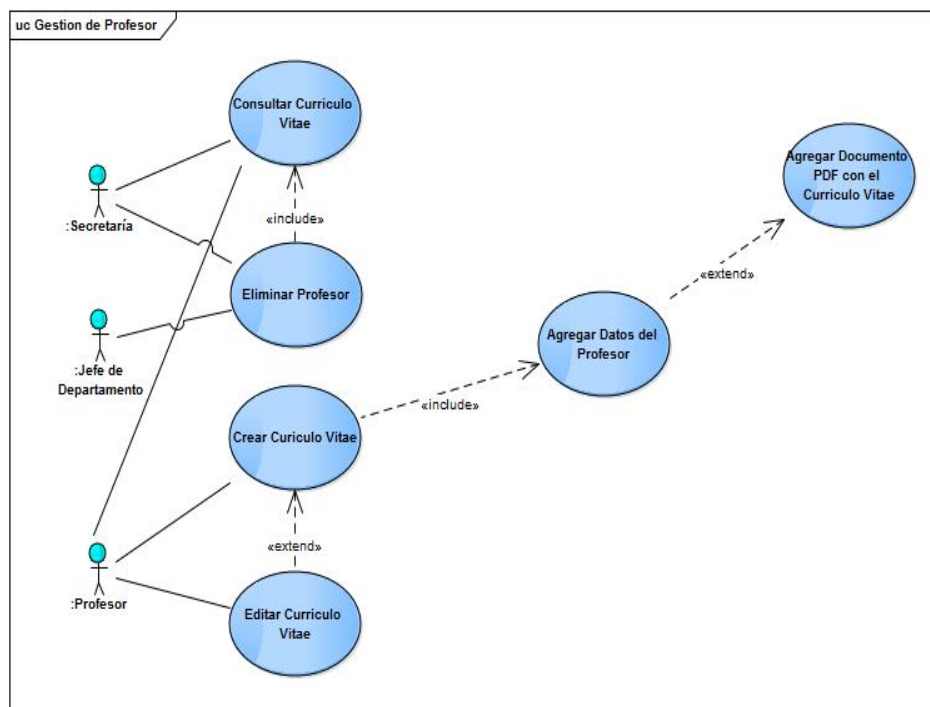


Figura 3.7: Diagrama Caso de Uso Gestión Profesor.

Capítulo 4

Diseño Del Sistema

En esta etapa se especifica la arquitectura de la aplicación, diseño de la interfaz, modelado de los componentes y módulos, es decir, describe el desarrollo del diseño arquitectónico del sistema de forma codificada, así como también, del diseño y aprovisionamiento de los componentes, a partir del conjunto de requerimientos funcionales y no funcionales obtenidos en el capítulo anterior, siempre en búsqueda de alcanzar los objetivos planteados.

4.1. Definición de los Componentes.

En esta fase se realiza una búsqueda de componentes de software reutilizables que cumplan con las especificaciones propuestas por los usuarios, así como también puede existir el desarrollo de componentes que no puedan ser localizados o que no cumplan con el propósito de complacer de manera correcta las especificaciones descritas anteriormente. En el desarrollo del sistema web para el Departamento de Computación, se reutilizó un esquema asociado a la base de datos perteneciente a la Gestión de Proyectos de Grados, que sirve para almacenar información de los usuarios, todos los demás componentes fueron desarrollados para que pudiesen así cumplir con los requerimientos establecidos previamente.

4.2. Vista Estructural.

Describe el sistema en términos de abstracciones (clases, colecciones u objetos y las interacciones entre ellos), forman parte de la vista estructural: el diagrama de Colecciones de la Base de Datos Db6 como se observan en las figuras 4.1 respectivamente.

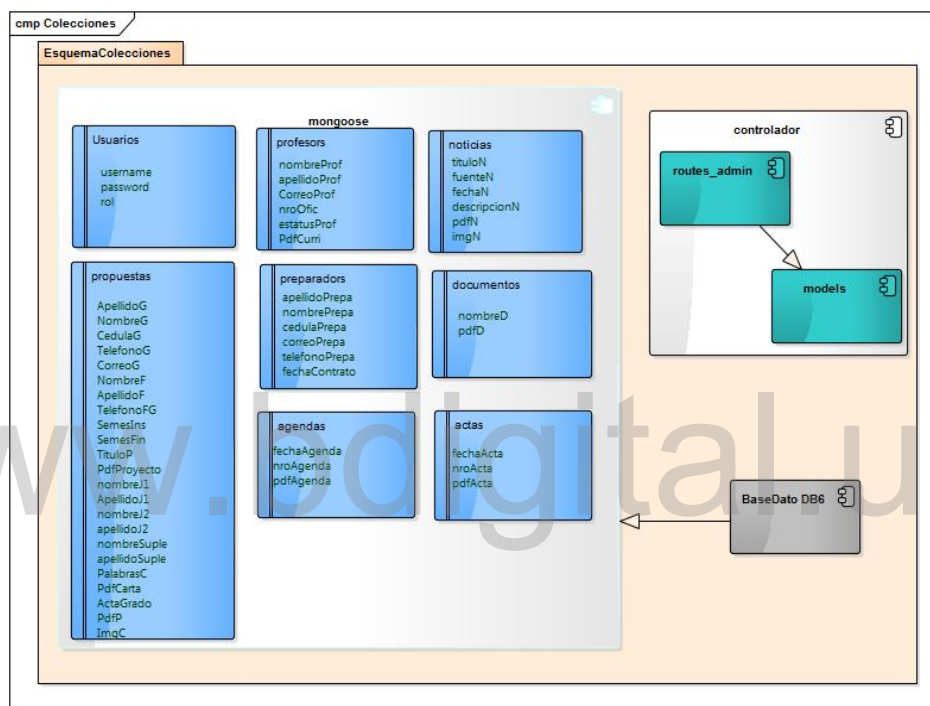


Figura 4.1: Diagrama de Colecciones de la Base de Datos Db6.

4.3. Vista de Implementación o Desarrollo.

Ilustra el sistema desde la perspectiva del programador y se enfoca en la administración de los componentes del software, como se puede observar en la figura 4.2

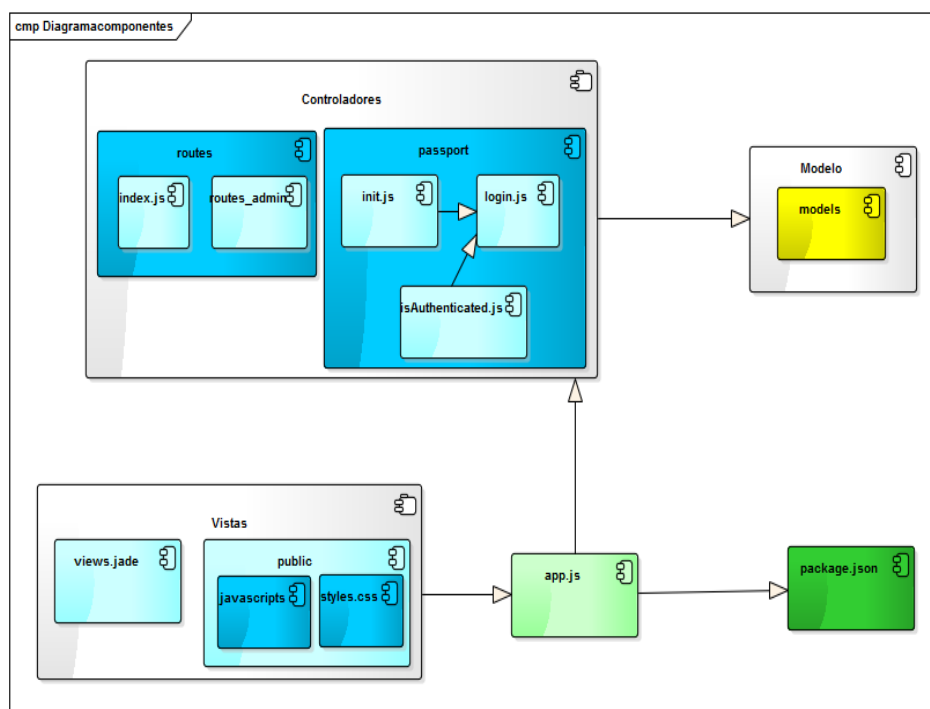


Figura 4.2: Diagrama de Componentes.

4.4. Vista de Ambiente.

Maneja los aspectos dinámicos del sistema, explica los procesos del sistema y cómo se relacionan, también se enfoca en el comportamiento del sistema en tiempo de ejecución, ver las figuras 4.3 4.5 4.4

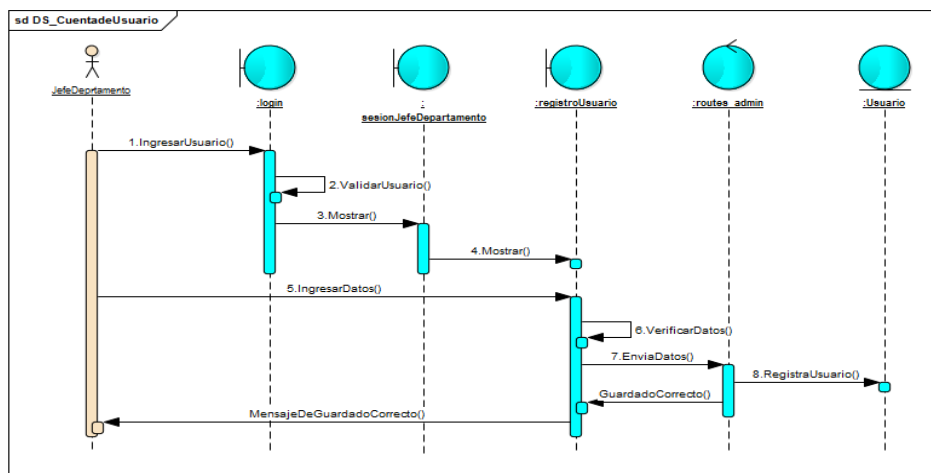


Figura 4.3: Diagrama de Secuencia Registro de Usuario.

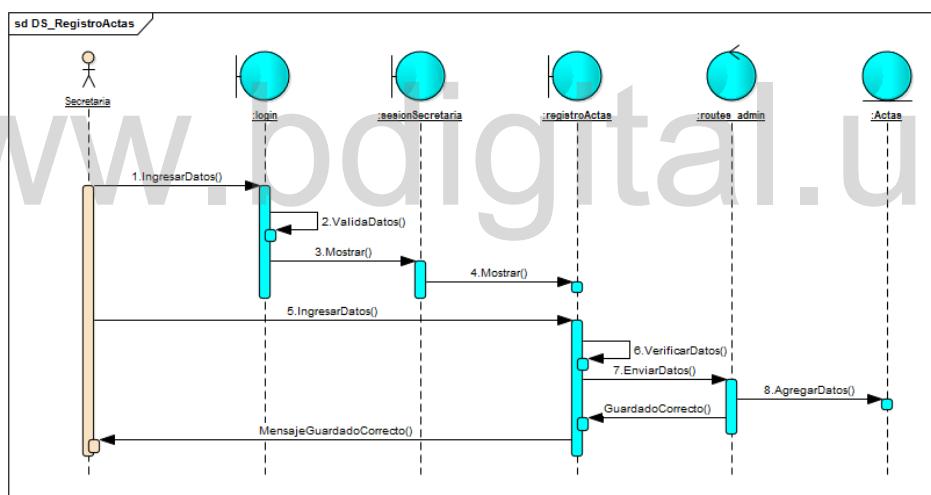


Figura 4.4: Diagrama de Secuencia Registro de Actas.

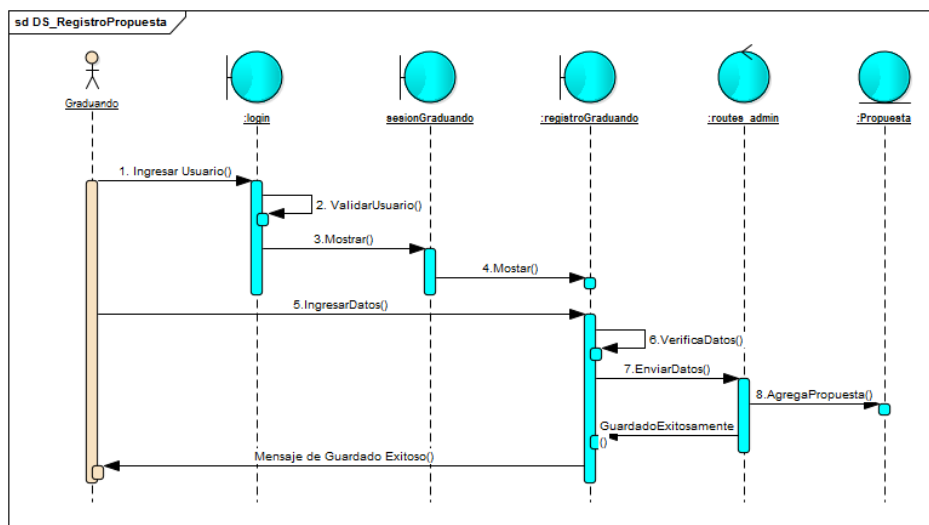


Figura 4.5: Diagrama de Secuencia de Registro de Propuesta de Proyecto de Grado.

4.5. Vista de Comportamiento o Despliegue.

Está relacionada con la topología de componentes de software en la capa física, así como las conexiones físicas entre estos componentes, véase la figura 4.6

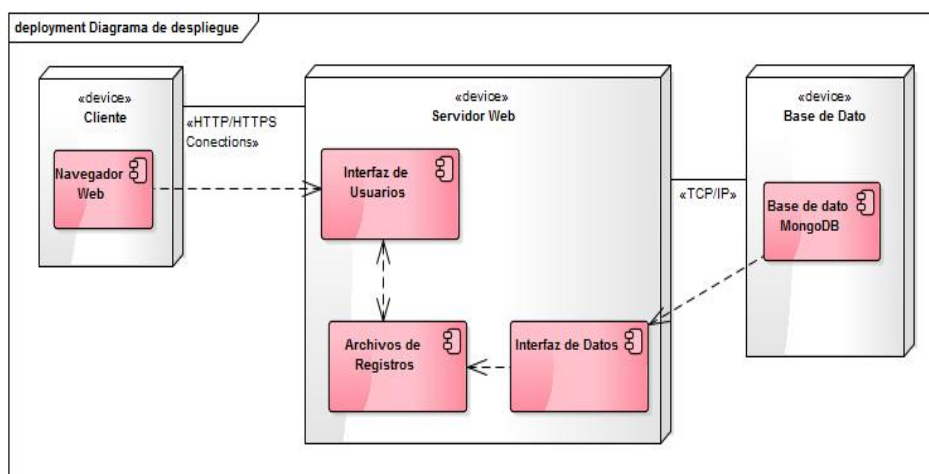


Figura 4.6: Diagrama de Despliegue.

4.6. Diseño de la Interfaz del Usuario.

4.6.1. Estructura de Navegación.

Esta basada en un grafo de navegación que representa gráficamente en forma de esquema la organización de la información de una estructura web, expresa todas las relaciones de jerarquía y secuencia entre los módulos o subpáginas de la aplicación, indicando los principales conceptos incluidos en el espacio de la información y las interrelaciones que existen entre ellos.

Las ventanas que forman parte de la interfaz de usuario son las siguientes:

Ventana principal: Es la vista principal con la que se encuentra el usuario a la hora de acceder a la dirección de la aplicación. Desde esta ventana se observan todos las subpáginas que posee el sistema.

Ventana de acceso: Es la ventana en la que se solicita el nombre y la contraseña del usuario que desea acceder al sistema.

Página emergente: Esta en una ventana de alertas que serán visibles a los usuarios al momento den que alguno de los datos o información ingresada no sean validos.

Páginas módulos: Son las vistas que contienen módulos en donde ciertos usuarios podrán tener acceso para realizar operaciones, esto dependerá de los permisos que posea cada uno de los usuarios.

Subpáginas: Son los enlaces que contienen los submódulos que redireccionan a cada una de las acciones que se pueden consultar.

Tablas desplegables: Esta vista permite visualizar el listado de los registro de usuarios o información que se encuentre almacenada en la base de datos, se utilizan a la hora de hacer consultas.

Páginas resultados: Es la vista que permite al usuario obtener información correspondiente a consultas q haya realizado.

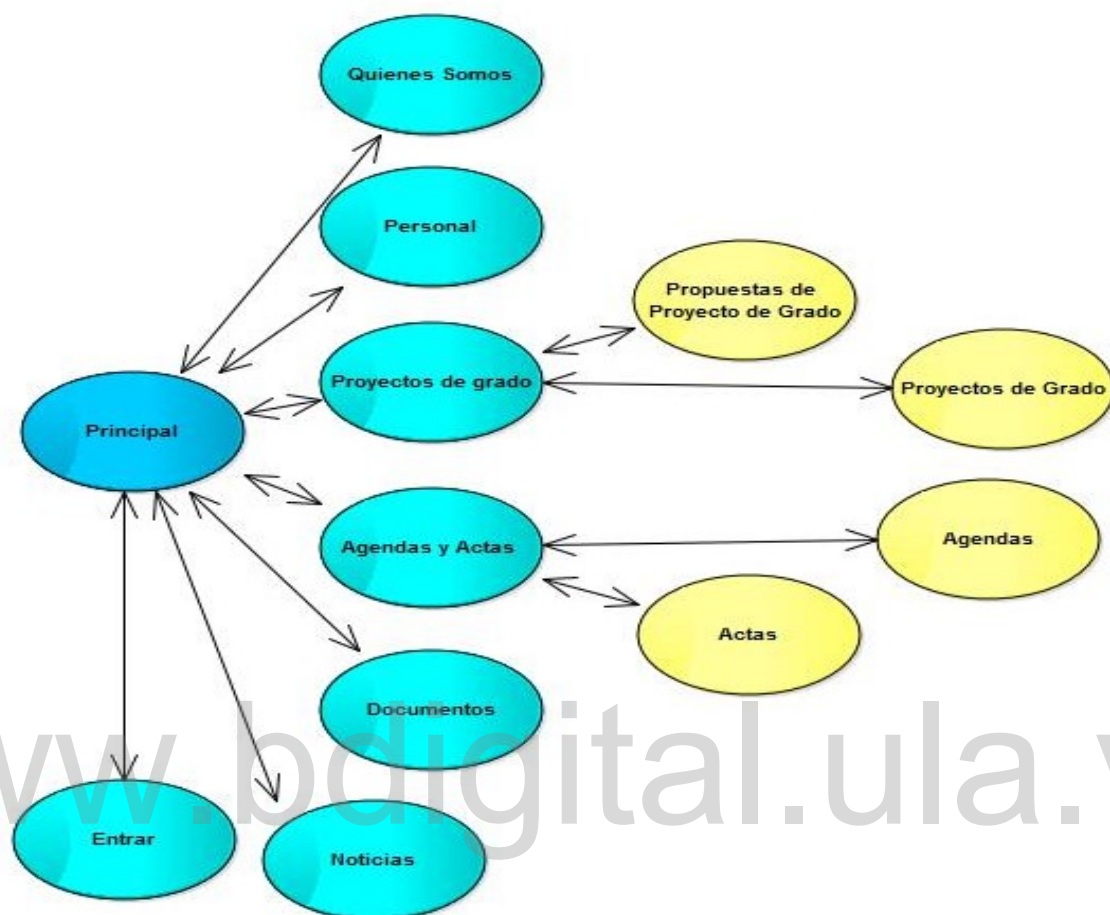


Figura 4.7: Grafo Navegación Usuario General.

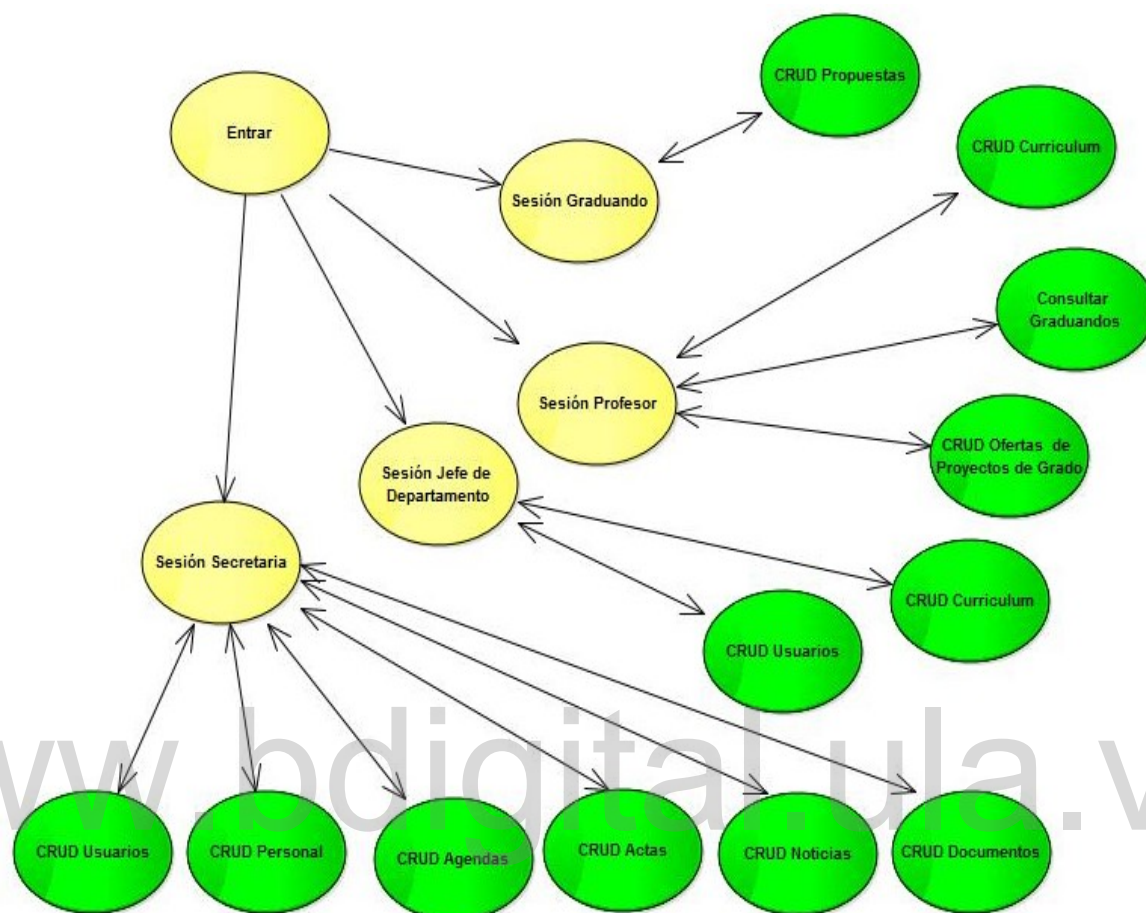


Figura 4.8: Grafo de Navegación Usuario Administradores.

4.6.1.1. Estilo visual de las páginas.

La aplicación web ofrece un estilo de fácil manejo y acceso para los usuarios, de interacción rápida y de búsquedas efectivas, los colores utilizados están relacionados con los colores alegóricos a la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes, además de ofrecer al usuario una presentación y recolección de información por medio de selección de menús, llenado de formularios y el uso de palabras comunes, lo que favorece a que el sistema sea de fácil comprensión a la hora de trabajar con él. A continuación se describen las partes que conforman la aplicación

Encabezado: Esta compuesto por: el logo del Departamento de Computación de la Universidad de Los Andes, también contiene la dirección de correo electrónico

institucional, número telefónico de contacto.

Barra de módulos: Contiene todas las pestañas correspondientes a los submódulos que posee el sistema.

- **Quienes Somos:** Es un menú desplegable que contiene las pestañas que redireccionan a: misión, visión, objetivos e historia referente al Departamento de Computación.
- **Personal:** Es el módulo que permite acceder a la interfaz donde se encuentra la información del personal del Departamento.
- **Proyectos de Grado:** Es un módulo con un menú desplegable que le permitirá al usuario entrar a la gestión de Propuestas de Proyectos de Grado o a la gestión de Proyectos de Grado.
- **Agendas y Actas:** Es un menú desplegable que permite acceder a la información de Actas o Agendas pertenecientes al departamento dependiendo del caso que desee consultar.
- **Documentos:** Permite acceder a la vista donde están todo lo referente a: Reglamentos o leyes en las cuales el departamento basa sus normas.
- **Noticias:** Permite acceder a la interfaz donde están todas las noticias relevantes tanto del Departamento como de la Escuela de Sistemas.

Botones: Son quienes tienen la tarea de hacer llamados a procesos que realicen acciones determinadas según el tipo de botón.

- **Ingresar al sistema:** Es el que realiza la acción de redireccionar al usuario a la página de acceso al sistema.
- **Guardar:** Invoca al proceso que permite almacenar en la base de datos alguna información que haya sido plasmada en los formularios que deben llenar los usuarios.
- **Consultar:** Permite redireccionar a la vista que contiene la información que haya sido consultada.

- **Editar:** Hace el llamado al proceso que redirecciona al usuario a una vista donde podrá realizar modificaciones a algún tipo de dato o información que ya este almacenada en la base de datos.
- **Eliminar:** Llama al proceso que realiza la acción de remover de la base de datos la información que haya sido seleccionada.
- **Inicio:** Regresa al usuario a la vista principal de la aplicación.

4.7. Estructura de la base de datos.

La base de datos del sistema de gestión del Departamento de Computación esta conformado por ocho colecciones, que abarcan: los usuarios, las actas, las agendas, documentos, noticias, preparadores, profesor y propuesta.

El lenguaje de MongoDB esta basado en una estructura de datos en documentos tipo JSON con un esquema dinámico conocido como BSON. Los datos almacenados son documentos que se guardan en colecciones, permitiendo así que haya un número indeterminado de documentos, estos se podrán manipular realizando operaciones como: registro, eliminación y edición de campos en cualquier momento ya que no existen esquemas predefinidos. La base de datos genera un `_id` que es único para cada uno de los documentos que haya sido almacenado en las colecciones, este es el campo conocido como clave principal para cada documento.



Figura 4.9: Colecciones de Base de Datos.

www.bdigital.ula.ve

Capítulo 5

Implementación y Pruebas Del Sistema

Esta etapa consiste en traducir el diseño arquitectónico de la aplicación realizado en el capítulo anterior en introducciones escritas en un lenguaje de programación que permita y facilite el cumplimiento de los requerimientos y necesidades planteadas en la subsección 3.3.2. El sistema de gestión administrativa y académica del Departamento de Computación, estará conformado por una base de datos que hace uso de un lenguaje de consulta basado en objetos *JSON* como parámetros y para el desarrollo del sistema programado se utilizó el lenguaje *JavaScript*. Luego de realizar la implementación del sistema es necesario que realizar una serie de pruebas que permiten corroborar el correcto funcionamiento del mismo, por medio de la puesta en práctica del sistema con la finalidad de encontrar errores de interfaces, base de datos o errores de programación.

5.1. Interfaz Gráfica

En esta fase se acopla la capa de presentación con los componentes asociados a la interfaz Usuario/Sistema, se muestra el flujo de ventanas que tendrá cada uno de los usuarios que forme parte del sistema.

1. El código del lado del cliente, contiene los elementos visuales (*HTML*, controles de servidor y texto estático) y el código que se debe ejecutar en el navegador del cliente,

en la forma de funciones *"JavaScript"*.

2. El código del lado del servidor web, contiene la lógica de programación de las páginas, esta capa es completamente manejada por el servidor nginx, en conjunto con el *"Framework"* *"Express"* donde los componentes se almacenan en archivos independientes. El código del lado del cliente se crea en un archivo .jade, es un motor de plantillas que simplifica la sintaxis de HTML, agiliza y facilita el proceso de desarrollo de plantillas HTML, además de permitir usar los controles web de HTML como son: botones de acción, cajas de texto, etiquetas, textarea, hipervínculos y controles sencillos como las tablas, las imágenes, la lógica de la página en archivos de clase independiente .js. Además, se utilizaron controles personalizados para presentar la información, entre estos se encuentran controles para emplearse en los filtros de búsqueda de usuarios, noticias, documentos, entre otros.

En la figura 5.1 se observa la pantalla principal del sistema, donde los usuarios tendrán acceso a los módulos y subpáginas de la aplicación de acuerdo a los permisos que tenga cada uno de los roles.

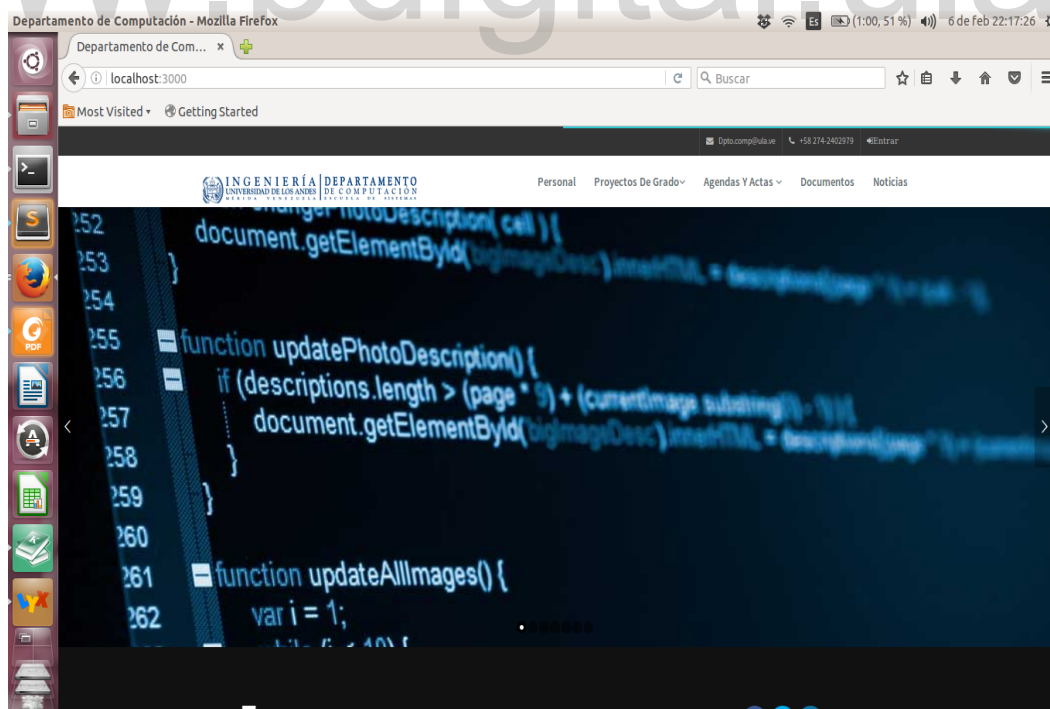


Figura 5.1: Vista Principal del Sistema.

En la figura 5.2 podemos observar la pagina de ingreso al sistema si el usuario es un administrador, donde deberá introducir los siguientes datos:

Nombre de Usuario: Es el nombre que utiliza el usuario para identificarse en el sistema.

Contraseña: Es una clave secreta que permitirá el acceso al sistema.

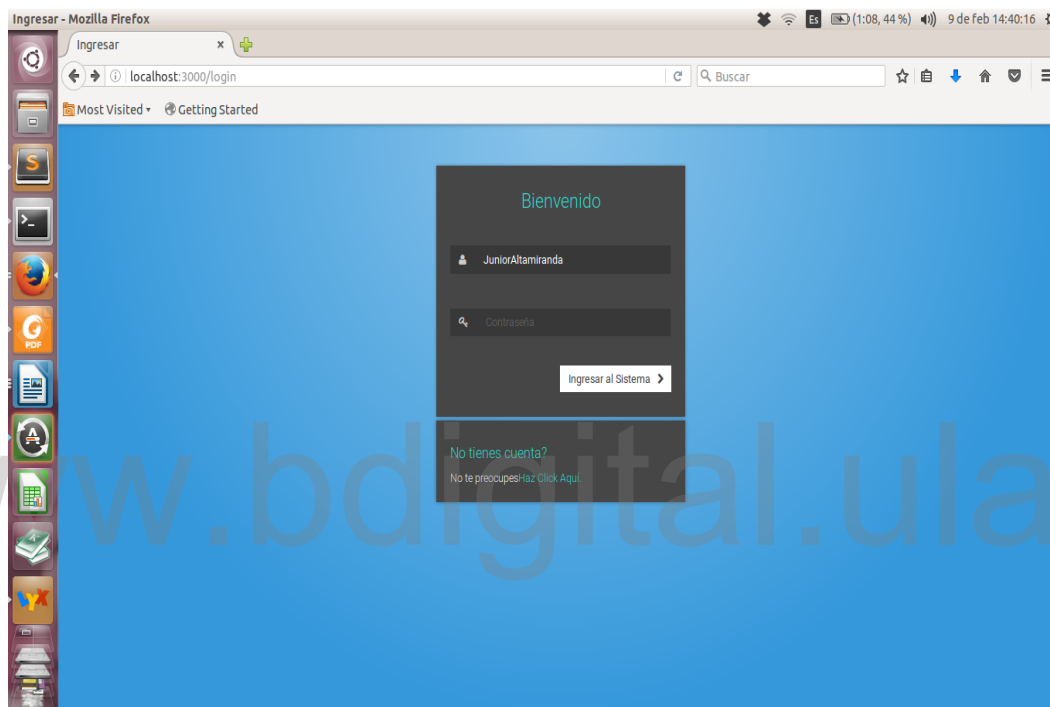


Figura 5.2: Vista de Acceso al Sistema

Esta información será validada por el sistema, lo que permitirá el acceso correcto a la aplicación de acuerdo a la autenticación que arroje como se muestra en la figura 5.3

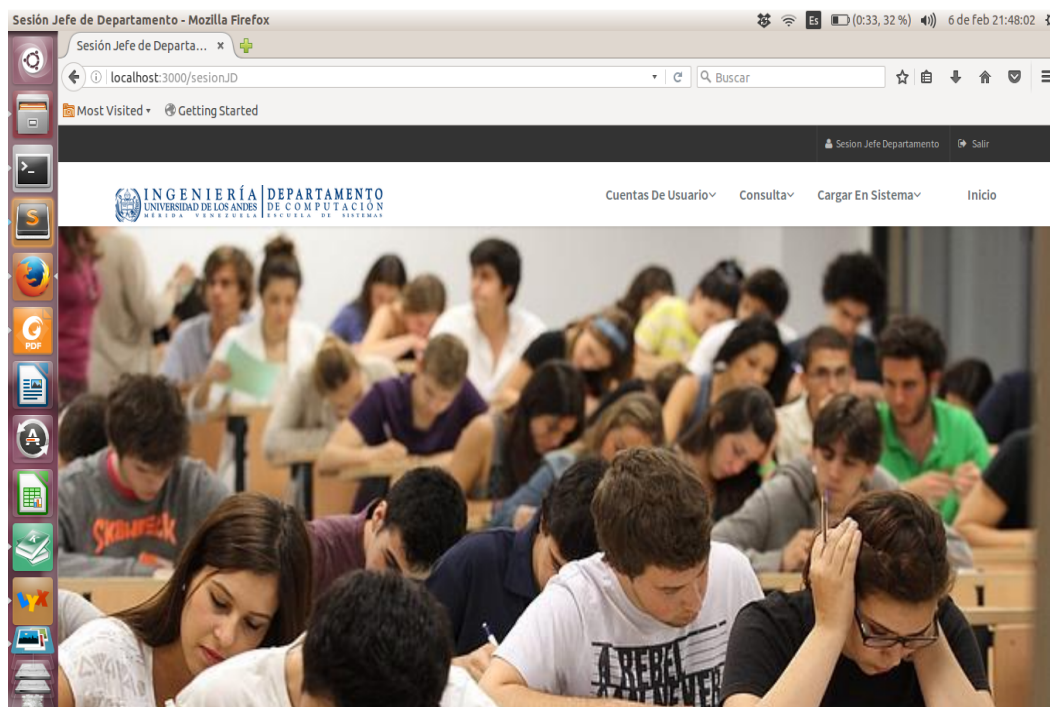


Figura 5.3: Acceso Correcto al Sistema como Jefe de Departamento.

En el caso contrario arrojará un mensaje de alerta notificando que no se pudo acceder al sistema por un error en alguno de los datos introducidos como se muestra en la figura 5.4.



Figura 5.4: Mensaje de Alerta de Contraseña Incorrecta.

Cuando se accede al sistema de manera correcta este lo redireccionara a una de las

siguientes sesiones de usuarios:

Jefe de Departamento: si el usuario accede al sistema bajo este rol de administrador la interfaz esta compuesta por la siguiente barra de menú:

- **Cuentas de usuarios:** Este módulo despliega un menú que ofrece las opciones de registrar o verificar si un usuario ya está registrado en sistema.
 - **Consulta:** Este módulo posee un menú desplegable donde el jefe de departamento podrá acceder al sistema de consulta de: actas, agendas, documentos, graduando, noticias y personal.
 - **Cargar en Sistema:** En este módulo despliega un submenú donde el jefe de departamento puede seleccionar entre las opciones de cargar su currículum vitae o los temas de proyectos de grado que desee ofrecer a la comunidad estudiantil.
- Inicio: Es un botón que permite redireccionar a la página principal del sistema.

Secretaria: Si el usuario que accede a esta sesión encontrará una barra de menú como se muestra en la figura 5.5 y los módulos disponibles para este usuario son:

- **Cuentas de usuarios:** Este módulo despliega un menú que ofrece las opciones de registrar o verificar si un usuario ya está registrado en sistema.
- **Consulta:** Este módulo posee un menú desplegable donde la secretaria podrá acceder al sistema de consulta de: actas, agendas, documentos, graduando, noticias y personal, obteniendo información reservada para usuarios generales, luego de realizar las consultas esta interfaz le permitirá editar, o eliminar la información o los datos que desee.
- **Cargar en sistema:** Este módulo posee un menú desplegable que le permite a la secretaria registrar en el sistema toda la información referente a actas, agendas, documentos, graduando, noticias y personal.
- **Inicio:** Es un botón que permite redireccionar a la página principal del sistema.

Profesor: Cuando se acceda al sistema bajo el rol profesor este tendrá acceso a dos menús:

- **Cargar en sistema:** Le permitirá al usuario registrar en el sistema el currículum vitae personal del profesor, así como también los proyectos de grado que el profesor desee ofertar a la comunidad estudiantil en general.
- **Consulta:** Le permitirá obtener información que este almacenada en el sistema sobre, actas, agendas, preparadores, graduando y personal.

Graduando: Cuando el usuario graduando accede al sistema podrá observar una barra de menús compuesto por:

- **Cargar en sistema:** Este menú desplegable le permitirá al graduando registrar en sistema datos de información personal, así como también de los datos relacionados a su propuesta de proyecto de grado, también podrá hacer un seguimiento de los documentos faltantes a lo largo de su proceso como futuro graduando.

5.2. Ensamblaje de Componentes

En esta sección se describe la implementación de toda la lógica de los datos de acuerdo al entorno de la aplicación web. En la capa de lógica de negocios se efectúa toda la lógica encargada de obtener la información relacionada con: los usuarios, actas, agendas, noticias, documentos, propuestas y proyectos de grado, personal. En esta capa es donde se lleva a cabo la lógica de negocios que se encarga de: insertar, consultar, editar o eliminar los datos antes mencionados. Toda la lógica de esta capa se encuentra implementada mediante procedimientos almacenados en el servidor y la base de datos, estos son:

1. Procedimientos para el manejo de datos relacionados con el registro de usuarios:
 - a) usuario/registroUsuario.
 - b) usuario/verificarUsuario.
 - c) usuario/mostrarUsuario.
 - d) usuario/editarUsuario.

INGENIERÍA DEPARTAMENTO
UNIVERSIDAD DE LOS ANDES DE COMPUTACIÓN
VENUELA VENEZUELA ESCUELA DE SISTEMAS

Sesión Secretaría Salir

Cuentas De Usuario Consultar Cargar En Sistema Inicio

REGISTRO DE USUARIO Volver

REGISTRO USUARIO

Tipo de Usuario Nombre de Usuario

Contraseña Confirmar Contraseña

GUARDAR

Enlaces De Interés

- Universidad De Los Andes
- Facultad de Ingeniería
- Escuela de Sistemas

©2016, Departamento de Computación. Todos los Derechos Reservados.

Figura 5.5: Registro de Usuario.

INGENIERÍA DEPARTAMENTO
UNIVERSIDAD DE LOS ANDES DE COMPUTACIÓN
VENUELA VENEZUELA ESCUELA DE SISTEMAS

Sesión Secretaría Salir

Cuentas De Usuario Consultar Cargar En Sistema Inicio

INFORMACIÓN USUARIOS Volver

Nombre de Usuario	Tipo de Usuario	
ADMIN	Administrador	Editar
RHONA	Administrador	Editar
ROSMARY	Secretaría	Editar
MIGUEL GOODY	Profesor	Editar
DOMINGO	Profesor	Editar
ALEJANDRO	Profesor	Editar
MIGUELGODOY	Graduando	Editar
JUNIORALTAMIRANDA	Administrador	Editar
RHONAREQUADRI	Graduando	Editar
FABIANA	Graduando	Editar
MARCOS	Profesor	Editar

Figura 5.6: Verificar Usuario.

2. Procedimientos para el manejo de datos relacionados con graduando:

- a) graduando/sesionGraduando
- b) graduando/registroGraduando
- c) graduando/mostrarGraduando
- d) graduando/verGraduando

The screenshot shows a web application interface for 'INGENIERÍA DEPARTAMENTO DE INGENIERÍA DE SISTEMAS'. The page title is 'REGISTRO GRADUANDO Y PROPUESTA DE PROYECTO DE GRADO'. The form is divided into two main sections: 'DATOS PERSONALES' and 'INFORMACIÓN DE PROPUESTA DE PROYECTO DE GRADO'. The 'DATOS PERSONALES' section includes fields for 'Nombres Graduando', 'Apellidos Graduando', 'Cédula de Identidad', 'Correo Electrónico', 'Número de Teléfono', 'Nombres Familiar', 'Apellidos Familiar', 'Número de Teléfono de Familiar', 'Período', 'Año Semestre de Inscripción', 'Período', and 'Año Semestre de Finalización'. The 'INFORMACIÓN DE PROPUESTA DE PROYECTO DE GRADO' section includes fields for 'Nombres Tutor', 'Apellidos Tutor', 'Título de Propuesta', 'Palabras Claves de Propuesta', 'Imagen Carta Compromiso', 'PDF Propuesta', and 'Imagen de Carta de Presentación de Avance'. There are 'Buscar' buttons next to the 'Imagen Carta Compromiso', 'PDF Propuesta', and 'Imagen de Carta de Presentación de Avance' fields. A 'Guardar' button is at the bottom of the form.

Figura 5.7: Registro de Graduando.

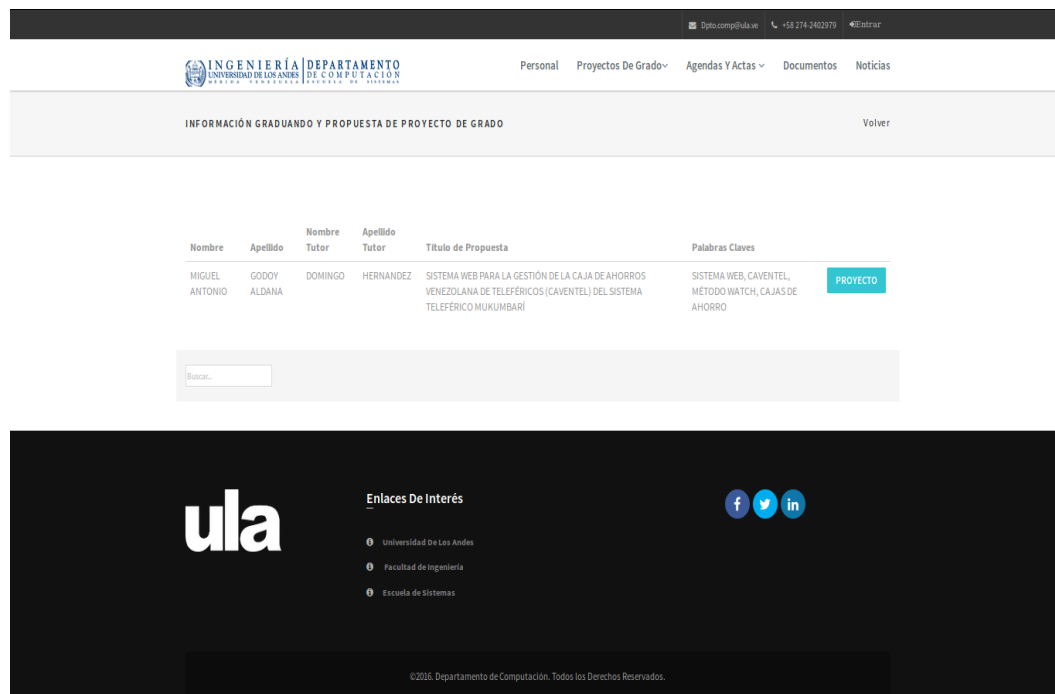


Figura 5.8: Vista de Información de Graduando.

3. Procedimientos para el manejo de datos relacionados con profesores:

- a) profesor/sesionProfesor
- b) profesor/registroProfesor
- c) profesor/mostrarProfesor
- d) profesor/editarProfesor

4. Procedimientos para el manejo de datos relacionados con preparadores:

- a) preparador/registroPreparador
- b) preparador/mostrarPreparador
- c) preparador/editarPreparador

5. Procedimientos para el manejo de datos relacionados con noticias:

- a) noticias/registroNoticias
- b) noticias/noticias
- c) noticias/mostrarNoticias
- d) noticias/editarNoticias

Session Secretaria Sair

INGENIERÍA DEPARTAMENTO
UNIVERSIDAD DE LOS ANDES DE COMPUTACION
FACULTAD DE INGENIERIA ESCUELA DE SISTEMAS

Cuentas De Usuario Consultar Cargar En Sistema Inicio

REGISTRO DE NOTICIAS Volver

INFORMACIÓN NOTICIAS

Titulo de la Noticia PDF del Documento de la Noticia Buscar

Autor de la Noticia Imagen de la Noticia Buscar

Fecha de Publicación de la Noticia DD/MM/YYYY

Descripción de la Noticia

GUARDAR

ula Enlaces De Interés

Universidad De Los Andes Facultad de Ingeniería

f t in

Figura 5.9: Registro Noticias.

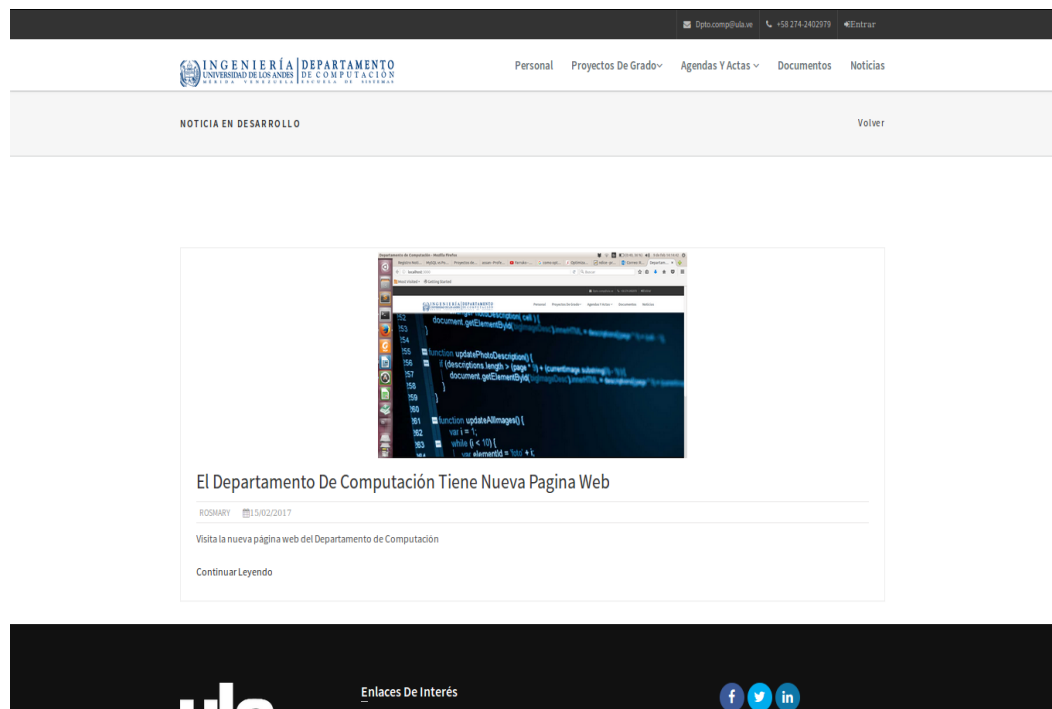


Figura 5.10: Consulta de Noticias.

6. Procedimientos para el manejo de datos relacionados con documentos:

- a) documentos/registroDocumentos
- b) documentos/documentos
- c) documentos/mostrarDocumentos
- d) documentos/editarDocumentos

7. Procedimientos para el manejo de datos relacionados con agendas:

- a) agendas/registroAgendas
- b) agendas/mostrarAgendas
- c) agendas/editarAgendas

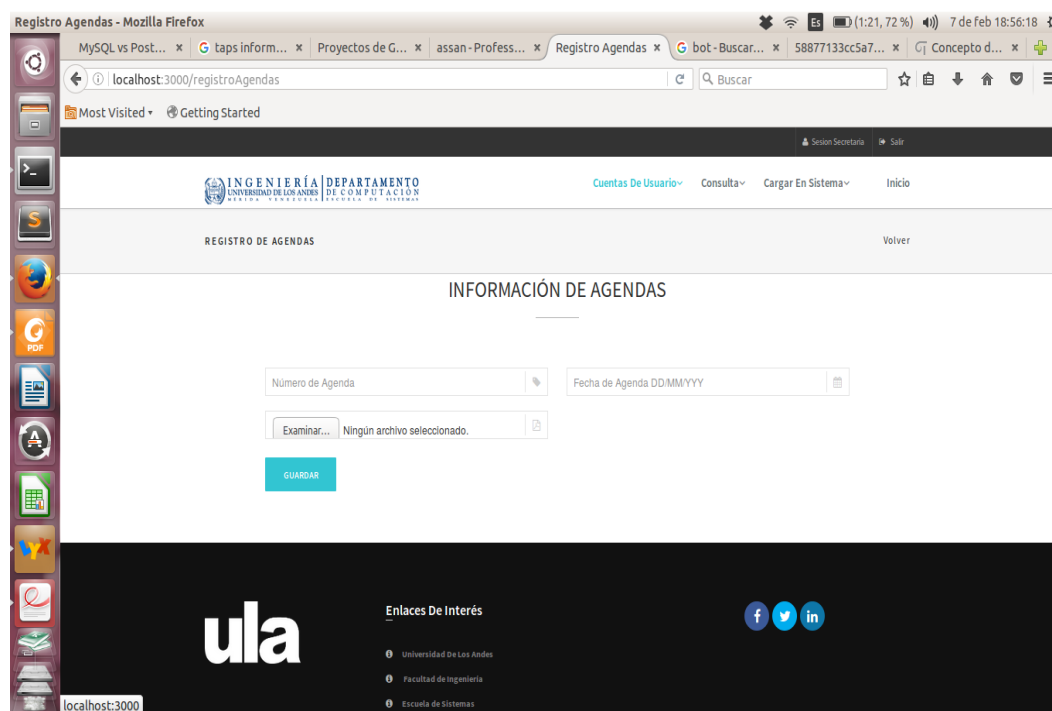


Figura 5.11: Registro de Agendas.

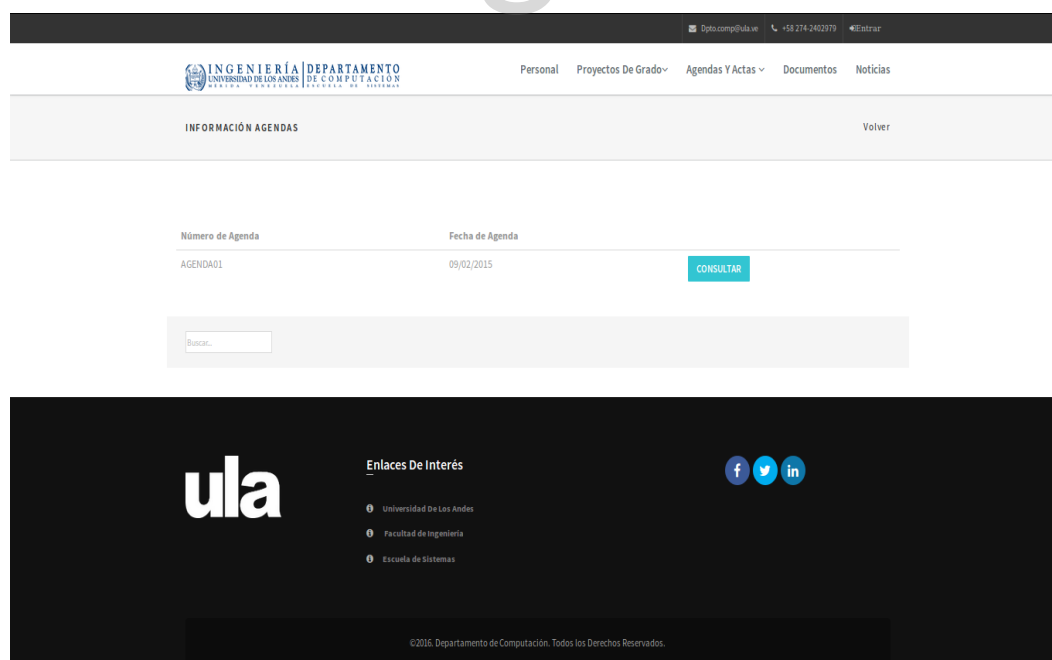


Figura 5.12: Consulta de Agendas.

8. Procedimientos para el manejo de datos relacionados con actas:

- a) actas/registroActas
- b) actas/mostrarActas
- c) actas/editarActas

Por medio de la figura 5.13 podemos observar como se lleva a ejecución el modelo MCV por medio de la integración de sus componentes:

1. Comunicación vista controlador

```
router.get("/RegistroGraduando", Authenticated, function (
    req, res) {
```

2. Comunicación del Controlador con el modelo Propuesta

```
Propuesta.findOne({userId: req.session.userId}
```

3. Respuesta del Modelo al controlador

```
function(err, propuesta){
```

4. Comunicación del Controlador a la Vista

```
res.render("graduando/registroGraduando",
```

```

router.route("/registroGraduando/:id") 1
.get(Authenticated, function (req, res) {
  if (req.session.roll == 'profe' || req.session.roll == 'office' || req.session.roll == 'admin') {
    2 Propuesta.findById(req.params.id, function (err, propuesta) {
      if (err) {
        console.log(String(err));
        switch (req.session.roll) {
        }
        return res.redirect(redirect);
      }
      console.log(propuesta.anioIns);
      console.log(req.session.roll)
      3 res.render("graduando/verGraduando", {propuesta: propuesta, roll: req.session.roll, message: req.flash('mes
    })
  } else if (req.session.roll == 'gradu') {
    Propuesta.findOne({userId:req.params.id}, function (err, propuesta) {
      if (err) {
      }
      if (propuesta) {
        var fecha = new Date();
        var anio = fecha.getFullYear();
        console.log(fecha.getMonth()+1);
        if (anio > propuesta.anioFin) {
          req.flash('message', 'Propuesta Fuera de Tiempo Reglamentario');
        }
        if (anio == propuesta.anioFin && propuesta.semestres == 'B' || propuesta.semestres == 'U' && (fecha.getMont
          req.flash('message', 'Propuesta Fuera de Tiempo Reglamentario');
        }
        res.render("graduando/verGraduando", {propuesta: propuesta, roll: req.session.roll, userId:req.session.us
      } else {
        req.flash('message', 'Debe Registrar Propuesta');
        res.redirect("/RegistroGraduando");
      }
    }
  }
})

```

Figura 5.13: Integración Componentes del Sistema

5.3. Pruebas del Sistema

La finalidad de esta etapa es comprobar la correcta funcionalidad del sistema de tal manera se verificará el apropiado funcionamiento de los módulos del sistema haciendo uso de las entradas realizadas por el usuario, estas pruebas son conocidas como cajas negras en este caso se utilizarán para comprobar los módulos de inserción, modificación y eliminación, además por medio de funciones *"JavaScripts"* se verificó que los datos que hayan sido ingresados sean correctos y así evitar que se almacenen datos erróneos en la base de datos.

5.3.1. Pruebas de caja blanca

Es un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo, las pruebas de caja blanca se llevan a cabo sobre un módulo realizando un seguimiento del código fuente según va ejecutando los casos de prueba, de manera que

se determinan de manera concreta las instrucciones, bloques, entre otros determinando los errores que existan.

Para verificar el comportamiento de los módulos se hizo pruebas sobre:

Registro de Usuarios: Se realizaron nuevos registros de usuarios del sistema, comprobando que se almacenaran de forma satisfactoria luego de pasar por las validaciones necesarias, la función usada para almacenar dichos datos fue:

```
exports.registroUsuario = function(req, res) {
  findOrCreateUser = function(){
    // find a user in Mongo with provided username
    Usuario.findOne({ 'username' : req.body.username.
      toUpperCase() }, function(err, user) {
      // In case of any error, return using the done method
      if (err){
        console.log('Error in SignUp: '+err);
        return done(err);
      }
      // already exists
      if (user) {
        console.log('User already exists with username: ' + req
          .body.username);
        res.render('usuario/
          registroUsuario', {message:
            'Usuario Existente'})
      } else if (req.body.password != req.body.
        password_confirmacion) {
        res.render('usuario/registroUsuario', {message: 'Las
          contraseñas NO Coinciden'})
      } else{
        // if there is no user with that email
        create the user
        var _usuario = new Usuario
        ();
        // set the user's local credentials
```

```

        _usuario.username = req.body.username.
            toUpperCase();
        _usuario.password = createHash(req.body.password);
        _usuario.password_confirmacion= createHash(req.body.
            password_confirmacion);
        _usuario.roll      = req.body.roll;
    // save the user
    _usuario.save(function(err) {
        if (err){
            console.log('Error in Saving user: '+err);
            //throw err;                res.render('usuario/
                registroUsuario ');
        }
        console.log('User Registration succesful');
        req.flash('message', 'Usuario Registrado
            Correctamente');          res.redirect('/
                informacionUsuarios ');
    });
}
});
};

var createHash = function(password){
    return bcrypt.hashSync(password, bcrypt.
        genSaltSync(10), null);    } }

```

Registro de Propuestas: Se realizaron nuevos registros de propuestas de proyectos de grados, comprobando que se almacenaran de forma satisfactoria luego de pasar por las validaciones necesarias, la función usada para almacenar dichos datos fue:

```

.post(Authenticated, function (req, res) {
    var extensionPDFCarta = req.body.pdfCarta.name.split(".").
        pop();

```



```
var extensionPDF = req.body.pdfP.name.split(".").pop();
var extensionimg = req.body.imgC.name.split(".").pop();
var _Propuesta = new Propuesta({
    apellidoG: req.body.apellidoG.toUpperCase(),
    nombreG: req.body.nombreG.toUpperCase(),
    cedulaG: req.body.cedulaG.toUpperCase(),
    telefonoG: req.body.telefonoG,
    correoG: req.body.correoG.toUpperCase(),
    nombreF: req.body.nombreF.toUpperCase(),
    apellidoF: req.body.apellidoF.toUpperCase(),
    telefonoFG: req.body.telefonoFG,
    semesIns: req.body.semesIns.toUpperCase(),
    anioIns: req.body.anioIns.toUpperCase(),
    semesFin: req.body.semesFin.toUpperCase(),
    anioFin: req.body.anioFin.toUpperCase(),
    nombreT: req.body.nombreT.toUpperCase(),
    apellidoT: req.body.apellidoT.toUpperCase(),
    tituloP: req.body.tituloP.toUpperCase(),
    palabrasC: req.body.palabrasC.toUpperCase(),
    pdfCarta: extensionPDFCarta,
    pdfP: extensionPDF,
    imgC: extensionimg,
    userId: req.session.userId,    });
_Propuesta.save(function (err) {
    if (!err) {
        if (extensionPDF) {
            var readPdf = fs.createReadStream(req.body.pdfP.
                path)
            var writePdf = fs.createWriteStream("public/pdf/
                propuesta/" + _Propuesta._id + "." +
                extensionPDF);
```

```
        readPdf.pipe(writePdf, function () {
            fs.unlinkSync(req.body.pdfP.path);
        });
    }
    if (extensionPDFCarta) {
        var readPdfCarta = fs.createReadStream(req.body.pdfCarta.path)
        var writePdfCarta = fs.createWriteStream("public/imagenes/cartaCompromiso/" + _Propuesta._id + "." + extensionPDFCarta);
        readPdfCarta.pipe(writePdfCarta,
function () {
            fs.unlinkSync(req.body.pdfCarta.path);
        });
    }
    if (extensionimg) {
        var readImg = fs.createReadStream(req.body.imgC.path)
        var writeImg = fs.createWriteStream("public/imagenes/actaAvance/" + _Propuesta._id + "." + extensionimg);
        readImg.pipe(writeImg, function () {
            fs.unlinkSync(req.body.imgC.path);
        });
    }
    req.flash('message', 'Guardado Exitoso');
    return res.redirect("/registroGraduando/" + req.session.userId);
} else {
    console.log(String(err));
    res.render("graduando/registroGraduando",
```

```
{roll: req.session.roll , message: 'Error Guardando'});  
    }  
  })  
});
```

5.3.2. Pruebas caja negra

Se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. Se parte de los requisitos funcionales, a muy alto nivel, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer como está construido por dentro. Las pruebas se aplican empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba.

Los datos de entrada se suministran, sin conocer o sin considerar el funcionamiento interno del sistema, se evaluara la respuesta obtenida para verificar si se amolda a la esperada.

Las pruebas que se hicieron con el sistema se ejecutaron en el servidor de desarrollo con la finalidad de que los posibles errores no afecten el ambiente de producción. A continuación se describen algunas de las pruebas realizadas.

5.3.2.1. Prueba de Registro de Propuesta de Proyecto de Grado

Inicialmente el graduando debe ingresar al sistema y dirigirse a la opción "Propuesta de Proyecto de Grado" del menú "Cargar en Sistema", debe ingresar todos los datos solicitados y guardar, luego de realizar esta operación el sistema debe mostrar los detalles del registro que se ha realizado, como se observa en la figura 5.14

Guardado Exitoso

APELLIDO Y NOMBRE DEL GRADUANDO RODRIGUEZ PEREZ FABIANA ANTONELLA	CEDULA 23456789	CONTACTO FABIANA@ULA.VE / 04164564534
APELLIDO Y NOMBRE DEL FAMILIAR RODRIGUEZ SILVIA	TELEFONO FAMILIAR 04266345678	
TITULO PROPUESTA MODELADO DE NEGOCIO PARA REDULA		PALABRAS CLAVES RED,MODELADO,ENRUTAMINETO
SEMESTRE INSCRIPCIÓN 2016-A	SEMESTRE CULMINACIÓN 2017-B	TUTOR RIVAS RAFAEL

Buttons: **EDITAR**, **CARTA COMPROMISO**, **CARTA AVANCE**

Figura 5.14: Prueba de Registro de Propuesta de Proyecto de Grado.

5.3.2.2. Prueba de Registro de Currículum de Profesores

El profesor debe acceder al sistema y dirigirse a la opción "Currículum Vítae/Temas de Proyectos de Grado" del menú "Cargar en Sistema", debe ingresar todos los datos y archivos solicitados y guardar, luego de realizar esta operación el sistema debe mostrar los detalles del registro que se ha realizado, como se observa en la figura 5.15

Guardado Exitoso

Nombres	Apellidos	Correo Electrónico	Cubículo	Estatus				
ALEJANDRO	MUJICA	ALEJANDROMUJICA@ULA.VE	104	Activo	CURRICULUM	TEMAS PROYECTO	EDITAR	ELIMINAR
MARCOS	PARRA	MARCOS@ULA.VE	103	Activo	CURRICULUM	TEMAS PROYECTO	EDITAR	ELIMINAR

Search bar:

Footer: ula, Enlaces De Interés, Universidad De Los Andes, Facultad de Ingeniería, Escuela de Sistemas

Figura 5.15: Prueba de Registro de Currículum de Profesores.

5.3.2.3. Prueba de Registro de Noticias

La secretaria debe acceder al sistema y dirigirse a la opción "Noticias" del menú "Cargar en Sistema", debe ingresar todos los datos y archivos solicitados y hacer clic en el botón guardar, luego de realizar esta operación el sistema debe mostrar la información detallada del registro que se ha realizado, como se observa en la figura 5.16



Figura 5.16: Prueba de Registro de Noticias.

5.3.2.4. Prueba de Registro de Proyecto de Grado

Inicialmente el graduando debe ingresar al sistema y dirigirse a la opción "Proyecto de Grado" del menú "Cargar en Sistema", debe ingresar todos los datos solicitados y guardar, luego de realizar esta operación el sistema debe mostrar los detalles del registro que se ha realizado, como se observa en la figura 5.17

Guardado Exitoso

APELLIDO Y NOMBRE DEL GRADUANDO RODRIGUEZ PEREZ FABIANA ANTONELLA	CEDULA 23456789	CONTACTO FABIANA@ULA.VE / 04164564534
APELLIDO Y NOMBRE DEL FAMILIAR RODRIGUEZ SILVIA	TELEFONO FAMILIAR 04266345678	
TITULO PROPUESTA MODELADO DE NEGOCIO PARA REDULA	PALABRAS CLAVES RED, MODELADO, ENRUTAMINETO	
SEMESTRE INSCRIPCIÓN 2016-A	SEMESTRE CULMINACIÓN 2017-B	TUTOR RIVAS RAFAEL

[PROPUESTA](#)
[CARTA COMPROMISO](#)
[CARTA AVANCE](#)

[PROYECTO](#)
[ACTA ENTREGA](#)
[EDITAR](#)

ula Enlaces De Interés [f](#) [t](#) [in](#)
 Universidad De Los Andes

Figura 5.17: Prueba de Registro de Proyecto de Grado.

5.3.3. Pruebas de Navegación

Se garantiza que todos los elementos del sistema web sean funcionales y permitan al usuario desplazarse a través de todas las páginas que componen un sitio web.

Módulo	Condición de Ejecución	Resultado
Acceso al login	Pulsar "Entrar"	Funciona Correctamente
Acceso al Sistema	Autenticar el usuario ingresando: Nombre de usuario, contraseña y pulsa Ingresar	Funciona Correctamente
Registro Usuario	Selecciona la pestaña "Registrar Usuario" del menú Cuentas de Usuario	Funciona Correctamente
Guardar Usuario	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente
Registro Graduando	Selecciona la pestaña "Propuesta de Proyectos de Grado" del menú "Cargar en Sistema"	Funciona Correctamente

Guardar Graduando	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente
Registro Profesor	Selecciona la pestaña "Curriculum Vitae" del menú "Cargar en Sistema"	Funciona Correctamente
Guardar Profesor	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente
Registro Preparador	Selecciona la pestaña "Preparador" del menú "Cargar en Sistema"	Funciona Correctamente
Guardar Preparador	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente
Registro Actas	Selecciona la pestaña "Actas" del menú "Cargar en Sistema"	Funciona Correctamente
Guardar Actas	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente
Registro Agendas	Selecciona la pestaña "Agendas" del menú "Cargar en Sistema"	Funciona Correctamente
Guardar Agendas	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente
Registro Noticias	Selecciona la pestaña "Noticias" del menú "Cargar en Sistema"	Funciona Correctamente
Guardar Noticias	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente
Registro Documentos	Selecciona la pestaña "Documentos" del menú "Cargar en Sistema"	Funciona Correctamente
Guardar Documentos	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente
Registro Personal	Selecciona la pestaña "Personal" del menú "Cargar en Sistema"	Funciona Correctamente
Guardar Personal	Ingresar los datos solicitados y pulsar el botón "Guardar"	Funciona Correctamente

Consulta Graduando	Selecciona la pestaña "Propuestas de Proyectos de Grado" del menú "Proyectos de Grado"	Funciona Correctamente
Consulta Personal	Selecciona la pestaña "Personal"	Funciona Correctamente
Consulta Noticias	Selecciona la pestaña "Noticias"	Funciona Correctamente
Consulta Documentos	Selecciona la pestaña "Documentos"	Funciona Correctamente
Consulta Agendas	Selecciona la pestaña "Agendas" del menú "Actas y Agendas"	Funciona Correctamente

Tabla 5.1: Pruebas de Navegación.

www.bdigital.ula.ve

Capítulo 6

Conclusiones Y Recomendaciones

6.1. Conclusiones

- Con el presente trabajo se logró cumplir con el objetivo general que es: Desarrollar un Sistema Web que permita la automatización de los procesos referidos al manejo de las propuestas de Proyectos de Grado y administrativo que se llevan a cabo por el Departamento de Computación de la Escuela de Ingeniería de Sistemas de la Universidad de Los Andes, de igual manera se lograron alcanzar los objetivos específicos.
- Por medio de la construcción de los diagramas de casos de uso y cadena de valor se alcanzó modelar los procesos que se llevan a cabo dentro del departamento, además de concretar los requerimientos planteados por los usuarios para el desarrollo del sistema.
- Se realizó la búsqueda de estudios aplicados a bases de datos SQL y NoSQL con la finalidad de conocer cual manejador de datos otorgaría un mejor rendimiento al sistema desarrollado a la hora de manejar una gran cantidad de datos.
- La arquitectura y organización del sistema se definió mediante el uso del método *Watch*, lo que hizo posible el diseño de cada uno de los componentes y módulos que conforman la aplicación, facilitando así la construcción del mismo.
- La implementación del sistema de gestión permite que los diferentes usuarios

accedan a él de forma rápida y fácil desde cualquier máquina con acceso a internet con lo que se permite obtener información vía on line.

- El desarrollo del sistema cumple con los requerimientos planteados, debido a que automatiza los procesos de: registro de propuestas y proyectos de grado, personal, actas , agendas, noticias y documentos, lo que permite tener mayor rendimiento y eficiencia en los servicios y actividades ejecutadas por el departamento.

6.2. Recomendaciones

- Ampliar e implementar el sistema a los demás Departamentos de EISULA o la Escuela en General.
- Establecer un plan de mantenimiento para el sistema de gestión que incluya actividades de limpieza de documentos y registros antiguos en la base de datos con el objetivo de optimizar su funcionamiento.
- Desarrollar proyectos que permitan la actualización tecnológica de la EISULA.
- Hacer uso del sistema con la finalidad de automatizar y actualizar los procesos llevados a cabo en el Departamento.
- Incluir los temas de bases de datos no relacionales (NoSQL) en los programas de la asignatura de Base de Datos.

Bibliografía

Barrios, J. (2013). Sistemas de Información. [http://webdelprofesor.ula.ve/ingenieria/ijudith/Archivos/clasesSIpdf/Programa_SI_09%20\(1\).pdf](http://webdelprofesor.ula.ve/ingenieria/ijudith/Archivos/clasesSIpdf/Programa_SI_09%20(1).pdf).

Correa, M. A. (2013). MySQL vs PostgreSQL vs MongoDB (velocidad). <http://macool.me/mysql-vs-postgresql-vs-mongodb-velocidad/04>.

Enciclopedia, V. (2016). Tipos de sistemas de información. http://www.eumed.net/libros-gratis/2012a/1169/tipos_de_sistema_de_informacion.html.

Enríquez, O. Y. et al. (2012). NoSql. <http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/74/74>.

Européens, C. d. S. (1977). Diseño de Bases de Datos. https://informaticosst.files.wordpress.com/2010/08/disenio_de_bases_de_datos_unidad_1_parte_1.pdf.

Guariman (2009). Diseño de una Aplicación Web para la Gestión en Línea de los Servicios Académicos de una Institución de Educación Superior. <http://ri.bib.udo.edu.ve/handle/123456789/1005>.

Guillen, P. R. (2003). Análisis y Diseño Orientado a Objetos. <http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%F1o%20orientado%20a%20objetos/rumbaugh.pdf>.

json.org (2013). JSON. <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.

Linares, F. (2016). Sistema Web para la Gestión de Proyectos de Grado en la EISULA.

- Márquez, M. (2016). Desarrollo de un Sistema de Recuperación Multimedia Basado en Manejadores de Bases de Datos NoSQL. <http://lyapunov.pg.ing.ula.ve/tutorPU>.
- MongoDB (2016a). Bases de Datos no Relacionales. <https://www.mongodb.com/es>.
- MongoDB (2016b). BSON. <https://www.mongodb.com/json-and-bson>.
- Montilva, J. (2004). Método WATCH. <https://www.scribd.com/doc/57229693/Metodo-WATCH-de-Prof-Jonas-Montilva>.
- Muñoz (2003). Sistemas de Información en las Empresas. https://www.upf.edu/hipertextnet/numero-1/sistem_infor.html.
- Rodriguez, H. (2008). Sistema de Gestión Académica para el Personal Docente y de Investigación de La Universidad de Los Andes.
- Segnini, A. (2009). Desarrollo de un Sistema Automatizado Bajo Entorno Web Para el Control de la Programación Académica en la Universidad de Oriente núcleo de Anzoátegui. <http://ri.bib.udo.edu.ve/bitstream/123456789/1850/1/21-TESIS.IDC009A30.pdf>.
- StrongLoop (2016). Express. <http://expressjs.com/es/>.
- UML (2015). Lenguaje Unificado de Modelado. http://en.wikipedia.org/wiki/Unified_Modeling_Language.

ANEXOS

www.bdigital.ula.ve

Anexo A

Método WATCH

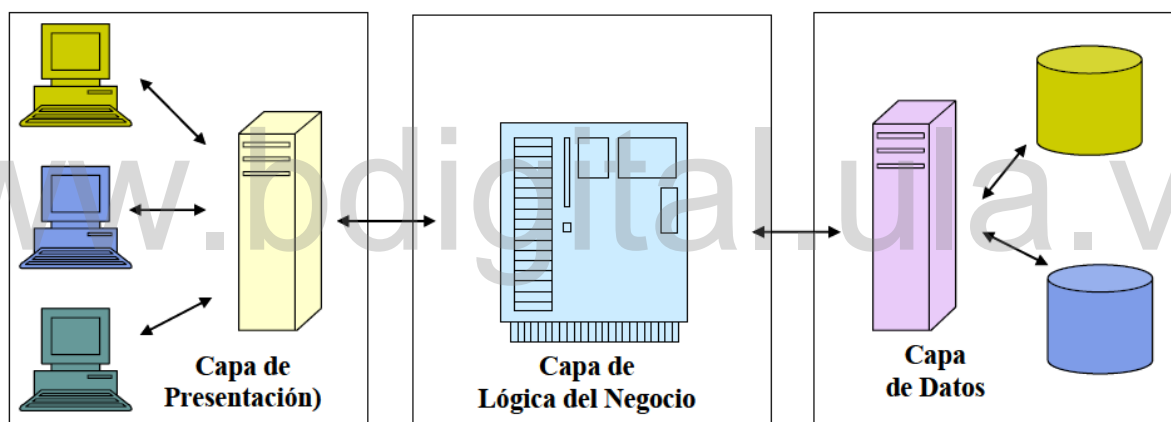


Figura A.1: Modelo 3 Capas

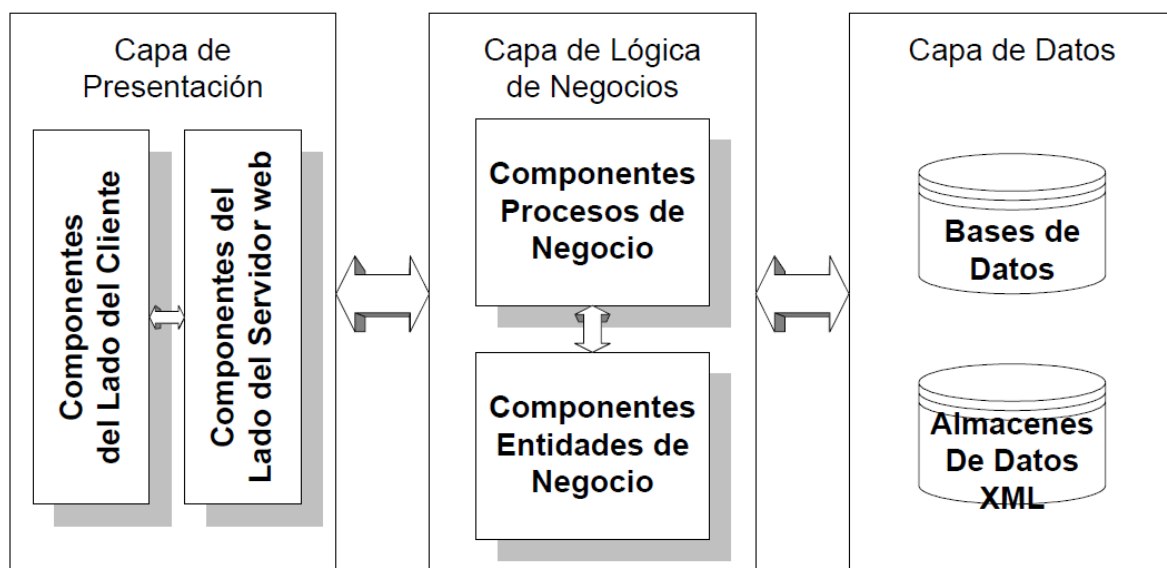


Figura A.2: Modelo de una aplicación empresarial basada en componentes

www.bdigital.ula.ve

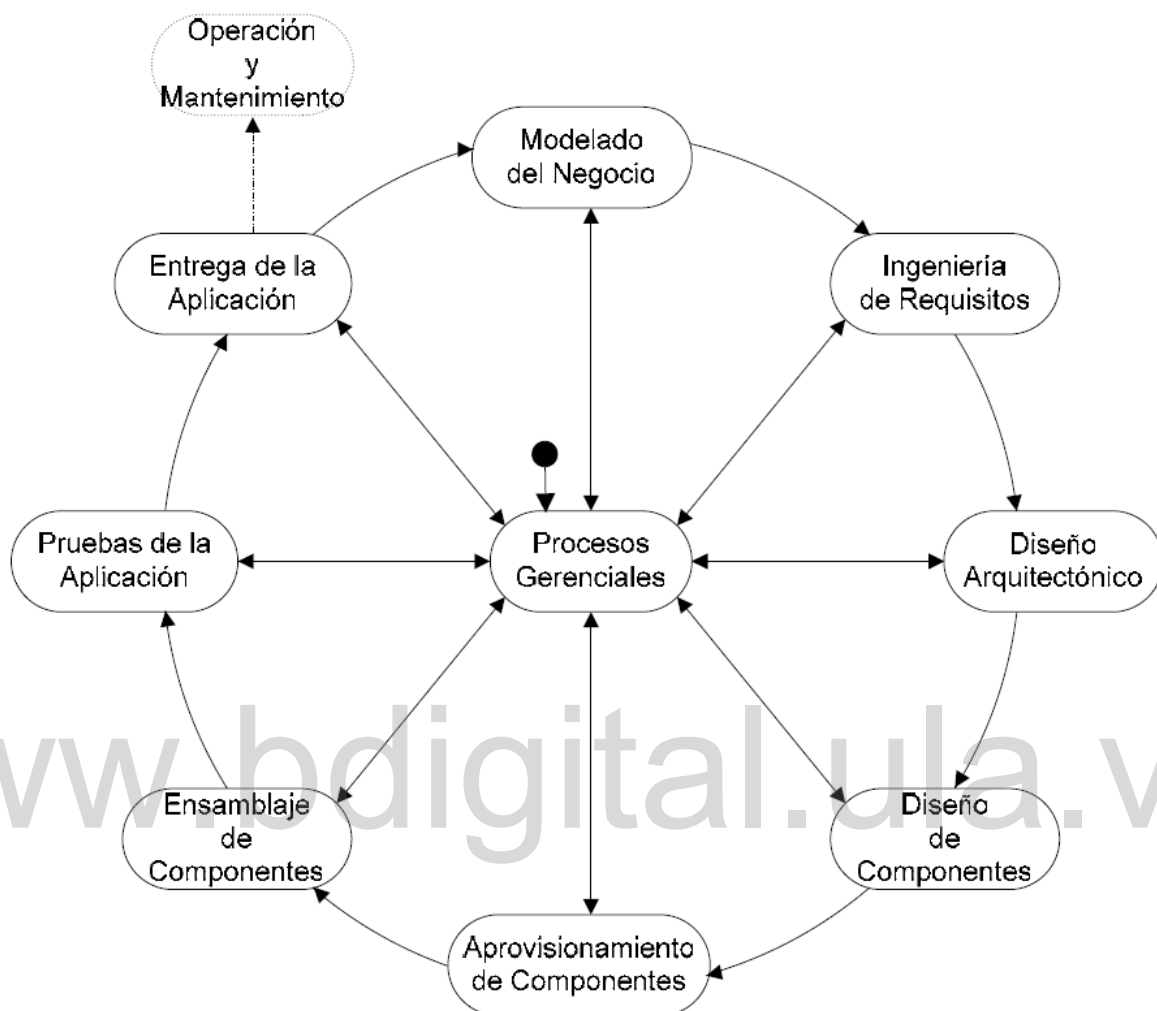


Figura A.3: Modelo de procesos WATCH

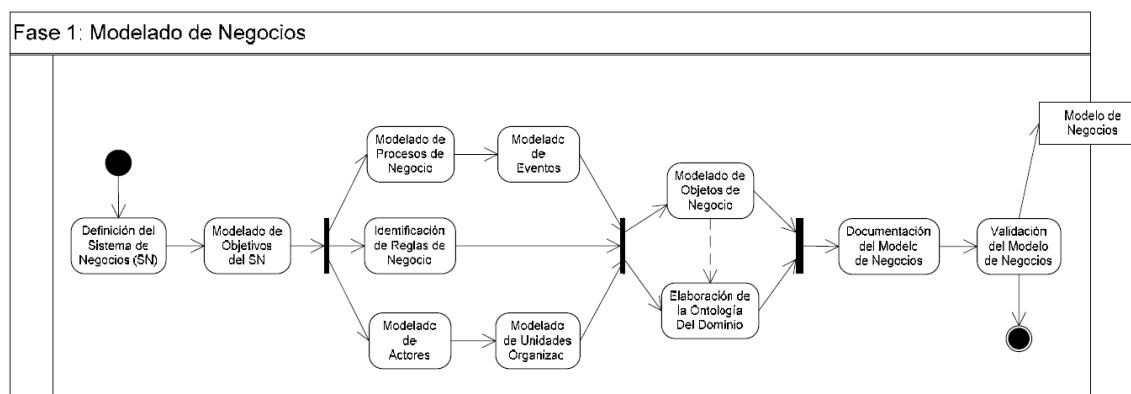


Figura A.4: El flujo de trabajo del Modelado del Negocio

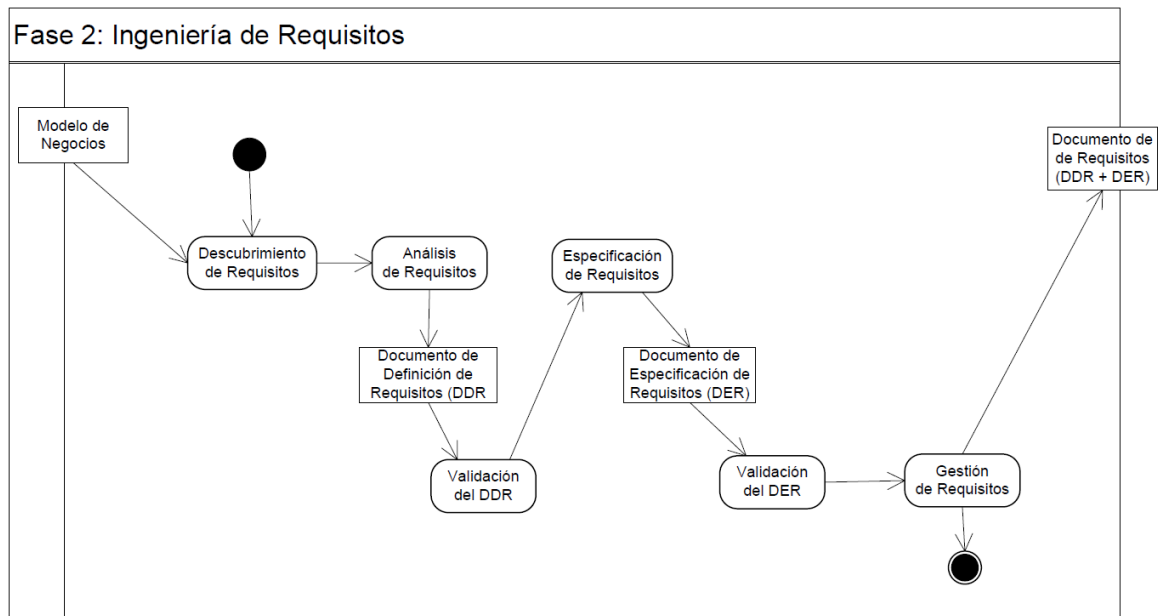


Figura A.5: Procesos de definición y especificación de requisitos

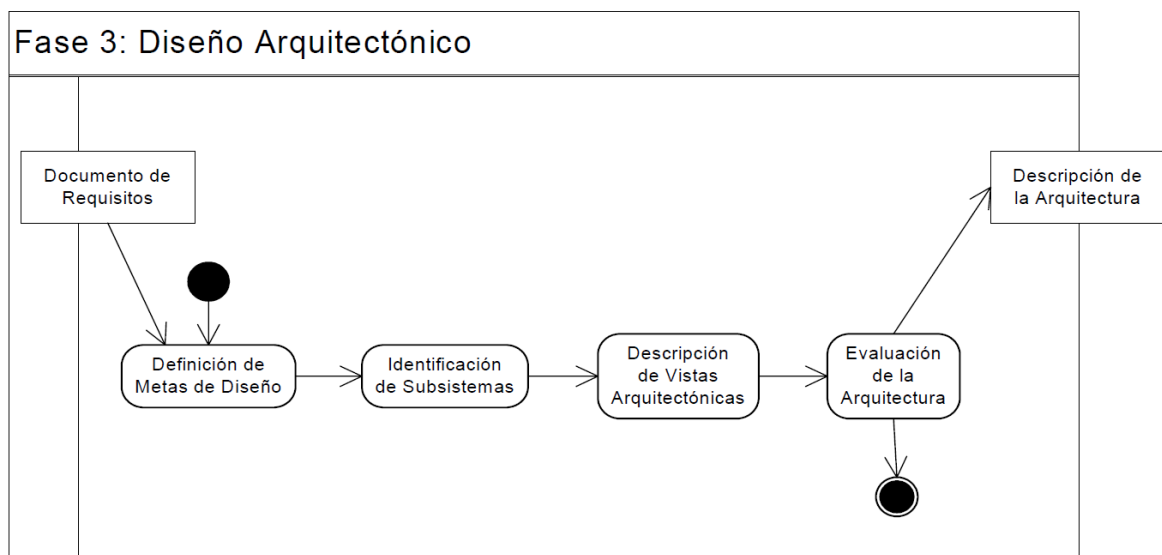


Figura A.6: Procesos del Diseño Arquitectónico

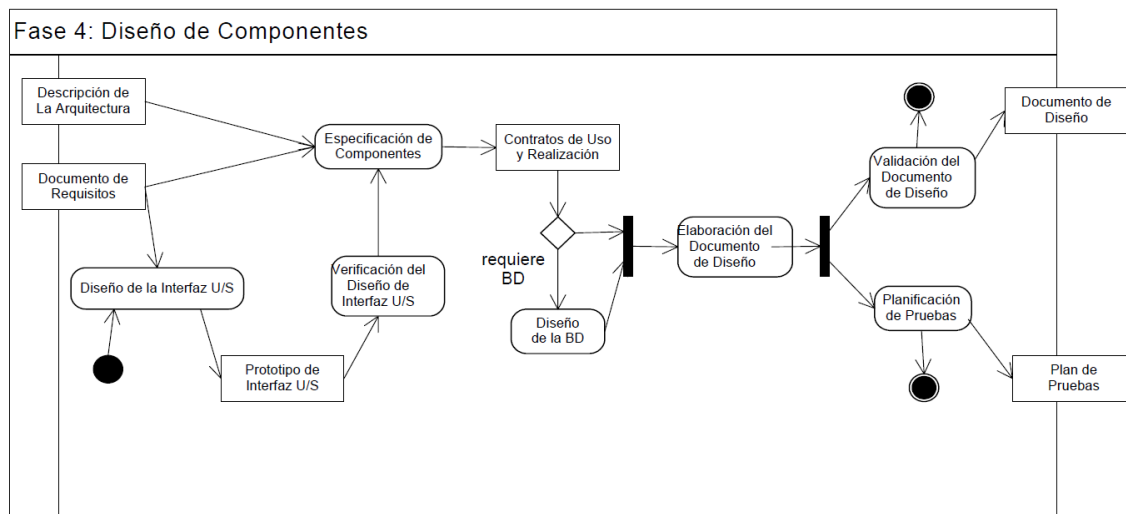


Figura A.7: Procesos del Diseño de Componentes

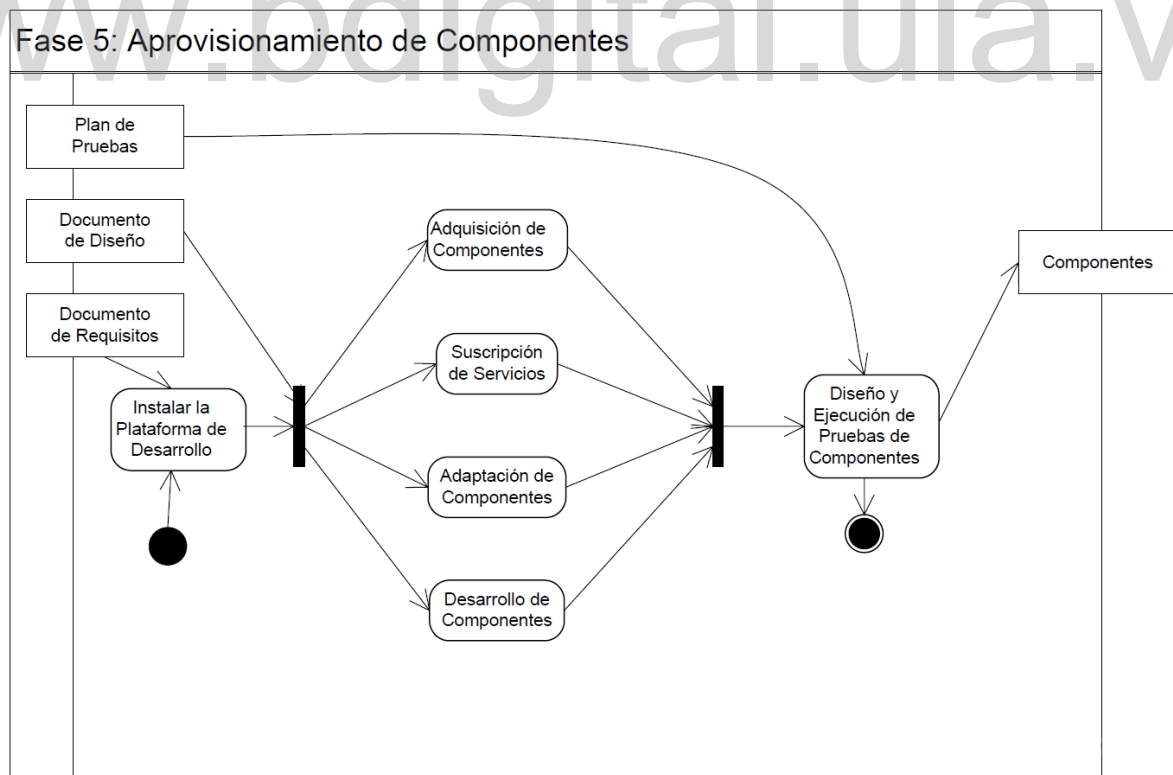


Figura A.8: Procesos del Aprovisionamiento de Componentes

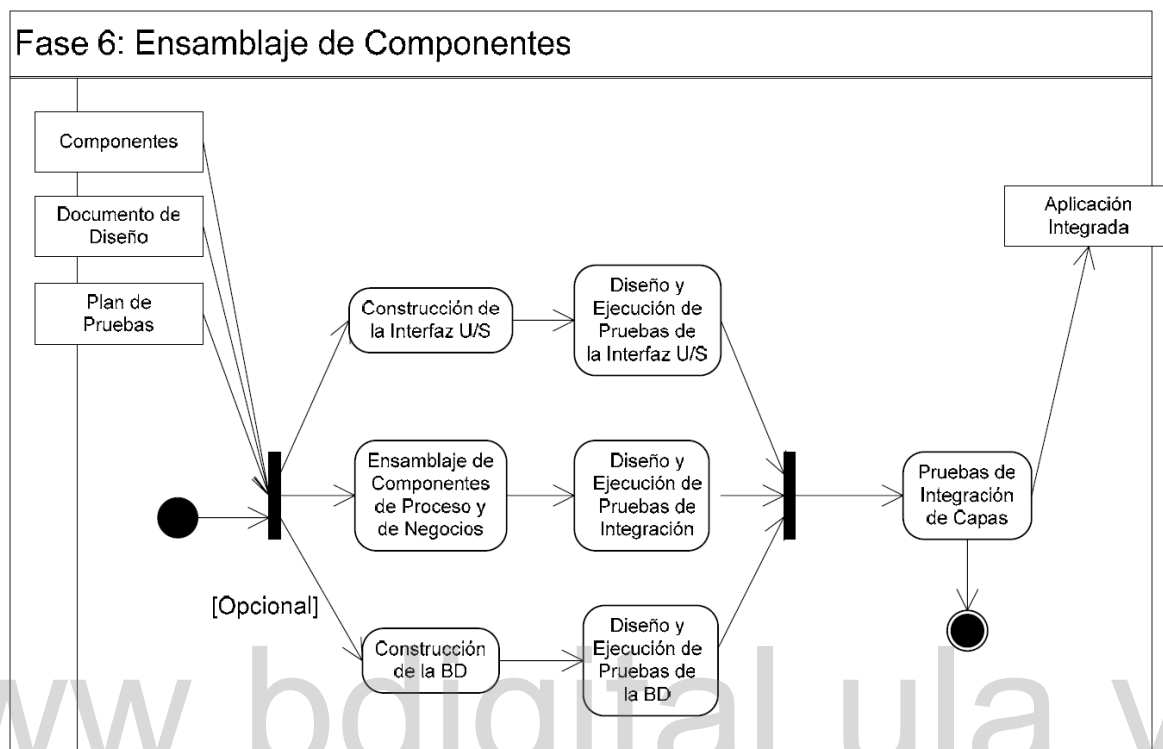


Figura A.9: Procesos del Ensamblaje de Componentes

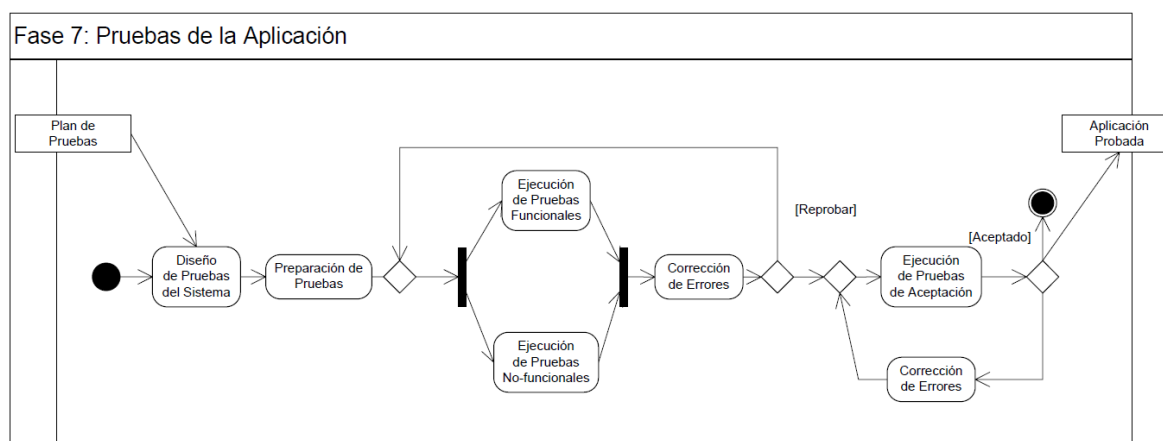


Figura A.10: Procesos de pruebas del subsistema

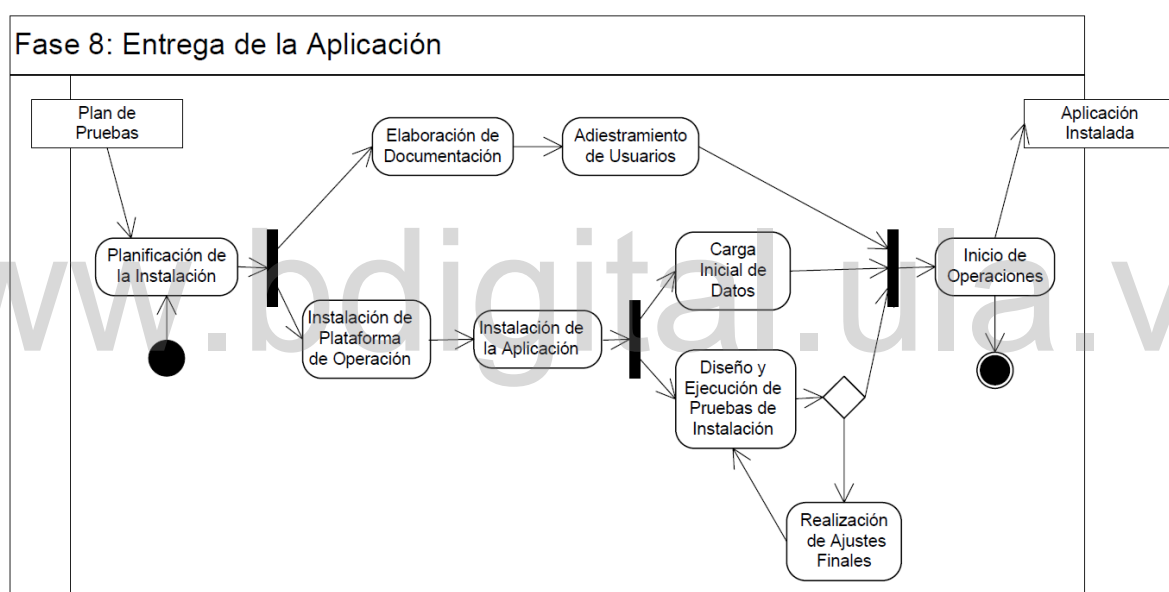


Figura A.11: Procesos de la Entrega de la Aplicación

Anexo B

Diagramas de Caso de Uso y Especificación Textual

B.1. Diagramas de Caso de Uso

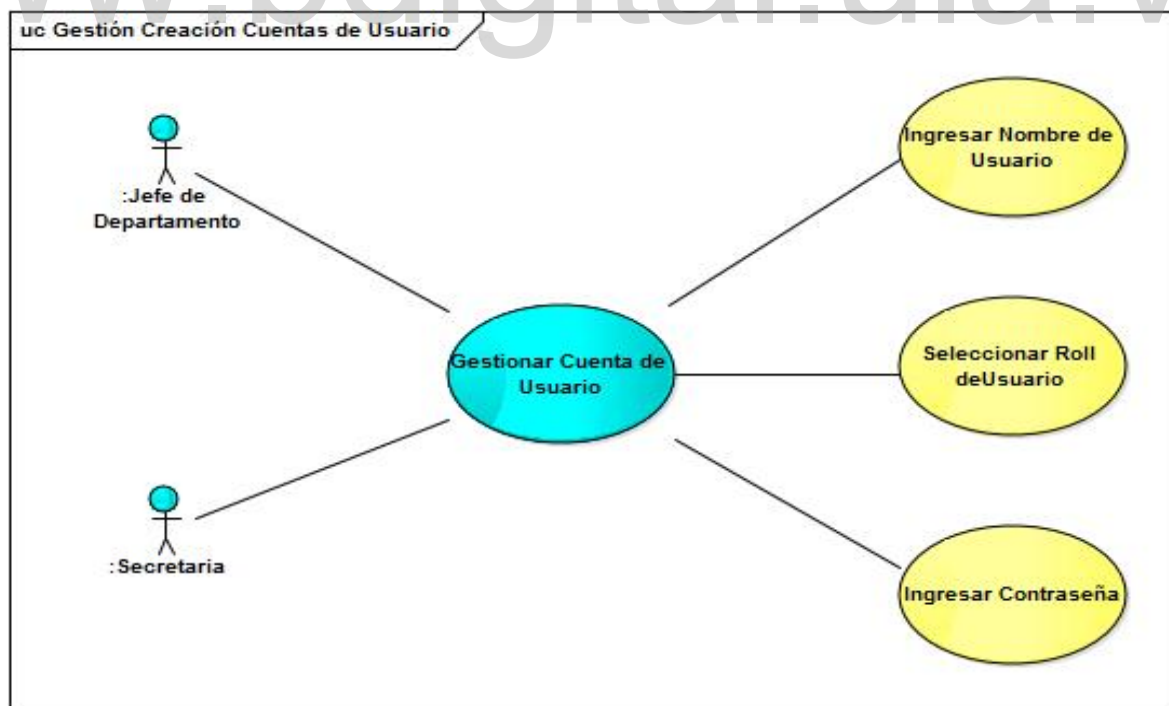


Figura B.1: Creación de Cuentas de Usuarios.

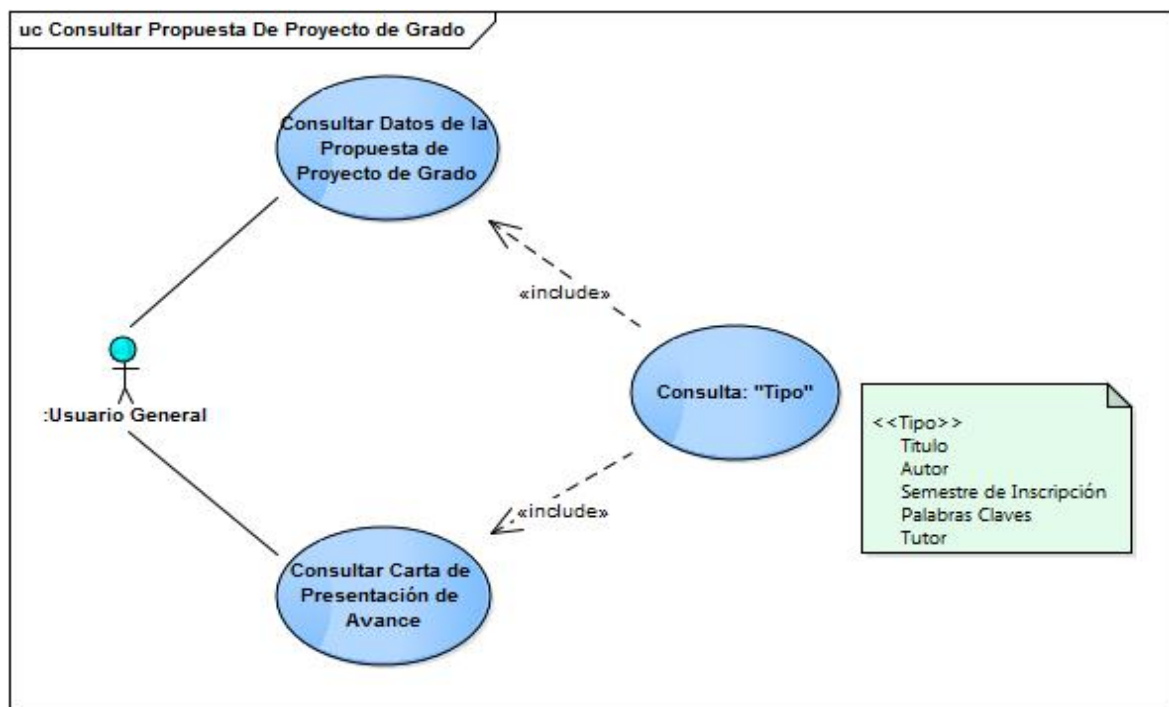


Figura B.2: Consulta de Propuesta de Proyecto de Grado.

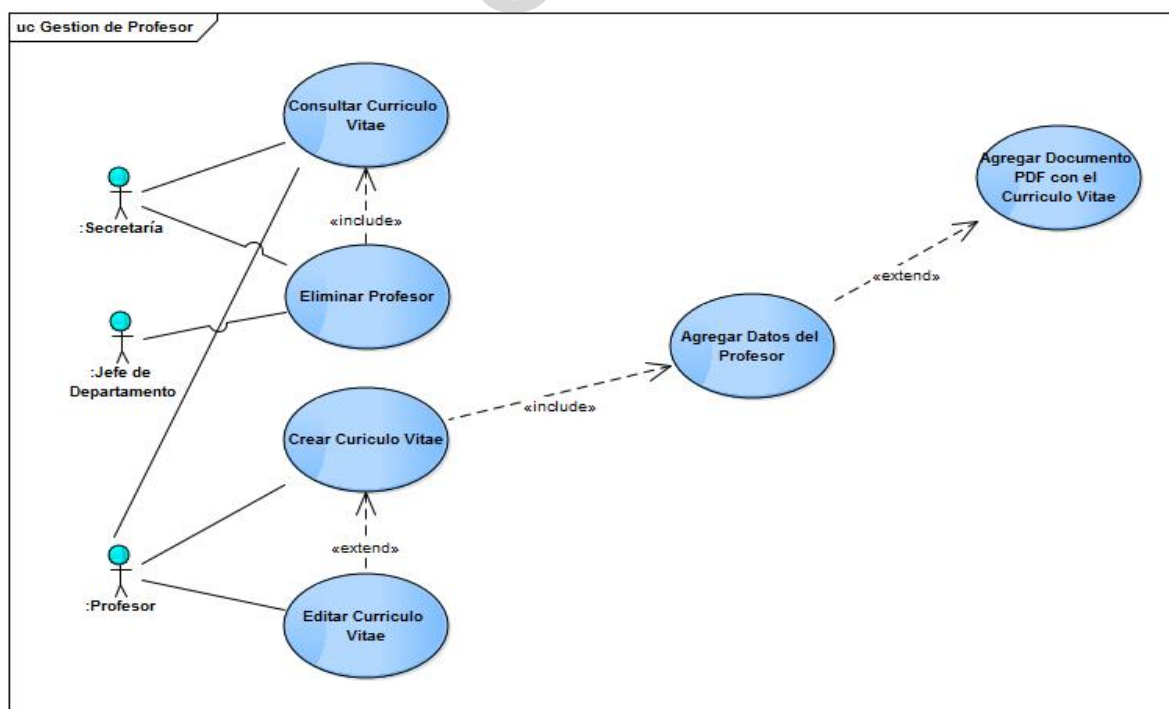


Figura B.3: Gestión de Profesores.

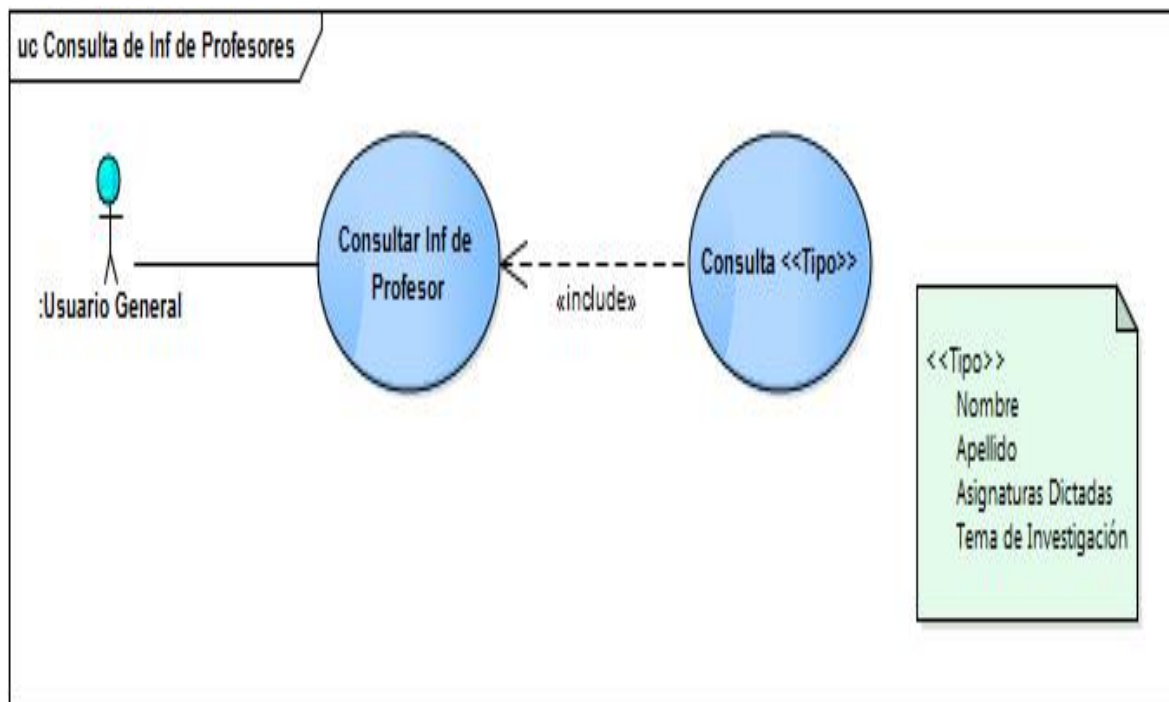


Figura B.4: Consulta Información de Profesores.

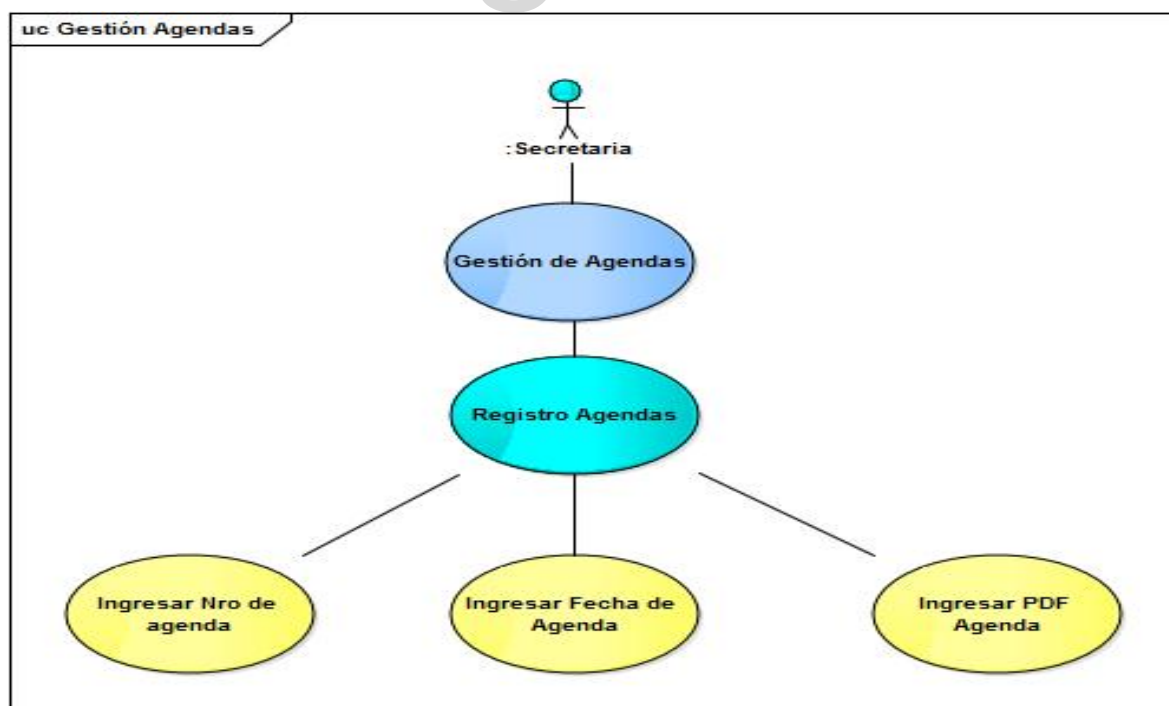


Figura B.5: Gestión Agendas.

B.2. Descripción textual de Diagramas de Caso de Uso

Caso de Uso:	CU13. Creación de Cuentas de Usuarios
Actores Principales:	Jefe de Departamento, Secretaria
Condiciones de Entrada:	El jefe de Departamento y la Secretaria deben haber ingresado al sistema.
Condiciones de Salida:	Creación correcta de la nueva cuenta de usuario.
Flujo de Eventos:	1.-El sistema permite visualizar una ventana donde solicita los datos: Tipo de Usuario, nombre de usuario, contraseña, y confirmación de contraseña . 2.- Introducir los datos solicitados 3.-Validación de datos en el sistema. 4.- Si los datos son válidos el sistema debe mostrar una página con los usuarios registrados en el sistema.
Flujos Alternativos:	4a.- En caso de que el nombre de usuario introducido no sea válidos el sistema debe arrojar una ventana de mensaje de “Error el Usuario no Existe o Es Incorrecto”. 4b.- En caso de que la contraseña introducida no sea válida el sistema debe arrojar una ventana de mensaje de “Error el Contraseña incorrecta”.

Tabla B.1: CU13. Creación de Cuentas de Usuarios.

Caso de Uso:	CU14. Consultar Propuesta de Proyecto de Grado
--------------	--

Actores Principales:	Jefe de Departamento, Secretaria, Graduando, Profesor y Usuario General
Condiciones de Entrada:	Debe haber propuestas de proyectos de grado almacenadas en el sistema
Condiciones de Salida:	Encontrar de manera satisfactoria la información solicitada al sistema.
Flujo de Eventos:	1.-El sistema mostrara una ventana donde se encontraran todas la propuestas de proyectos de grado, donde los usuarios podrán hacer consultas por nombre, apellido, titulo, tutor o palabras claves.
Flujos Alternativos:	

Tabla B.1: CU14. Consultar Propuesta de Proyecto de Grado.

Caso de Uso:	CU15. Gestión de Profesores
Actores Principales:	Profesores
Condiciones de Entrada:	El usuario profesor debe haber iniciado sesión e ir a la opción currículum vitae/Temas de Proyectos.
Condiciones de Salida:	Carga satisfactoria de los datos personales, documento PDF de su currículum vitae y los temas de proyectos de grado disponibles

Flujo de Eventos:	1.-El sistema mostrara una ventana donde se deberán introducir los datos del profesor. 2.- El usuario debe dar clic sobre el botón guardar, de esta manera el sistema almacenara los datos introducidos. 3.- El sistema validara los datos, cargados al sistema. 4.- Si los datos son válidos el sistema mostrara una ventana con los datos introducidos, y el archivo de Pdf del currículum
Flujos Alternativos:	3a.- Si algún campo obligatorio queda vacío; el sistema envía un mensaje de “Por favor, llena este campo” 3b.- Si alguno de los documentos y/o datos ingresados no son correctos el sistema envía un mensaje “Error los documentos y/o datos deben tener el formato correcto”

Tabla B.1: CU15. Gestión de Profesores.

Caso de Uso:	CU16. Consultar Agendas o Actas
Actores Principales:	Jefe de Departamento, Secretaria, Graduando, Profesor y Usuario General
Condiciones de Entrada:	Debe haber registro de agendas o actas en la base de datos
Condiciones de Salida:	Encontrar la información requerida.

Flujo de Eventos:	1.-El sistema mostrara una ventana donde se deberá introducir la información solicitada. 2.- El sistema validara los datos introducidos. 3.- Si los datos son válidos el sistema arrojara la información antes solicitada.
Flujos Alternativos:	2a.- Si algún campo obligatorio queda vacío; el sistema en envía un mensaje de “Por favor, lene este campo” 2b.- Si alguno de los documentos y/o datos ingresados no son correctos el sistema envía un mensaje “Error en el dato introducido”

Tabla B.1: CU16. Consulta de Agendas o Actas.

Caso de Uso:	CU17. Gestión de Personal
Actores Principales:	Secretaria
Condiciones de Entrada:	El usuario secretaria debe haber iniciado sesión e ir a la opción "Personal" del menú cargar en sistema.
Condiciones de Salida:	Carga satisfactoria de los datos solicitados en el formulario

Flujo de Eventos:	1.-El sistema mostrara una ventana donde se deberán introducir los datos del personal a registrar. 2.- El usuario debe dar clic sobre el botón guardar, de esta manera el sistema almacenara los datos introducidos. 3.- El sistema validara los datos, cargados al sistema. 4.- Si los datos son válidos el sistema mostrara una ventana con los datos introducidos
Flujos Alternativos:	3a.- Si algún campo obligatorio queda vacío; el sistema en envía un mensaje de “Por favor, llena este campo” 3b.- Si alguno de los documentos y/o datos ingresados no son correctos el sistema envía un mensaje “Error los documentos y/o datos deben tener el formato correcto”

Tabla B.1: CU17. Gestión de Personal.

Anexo C

Diagramas de Secuencia

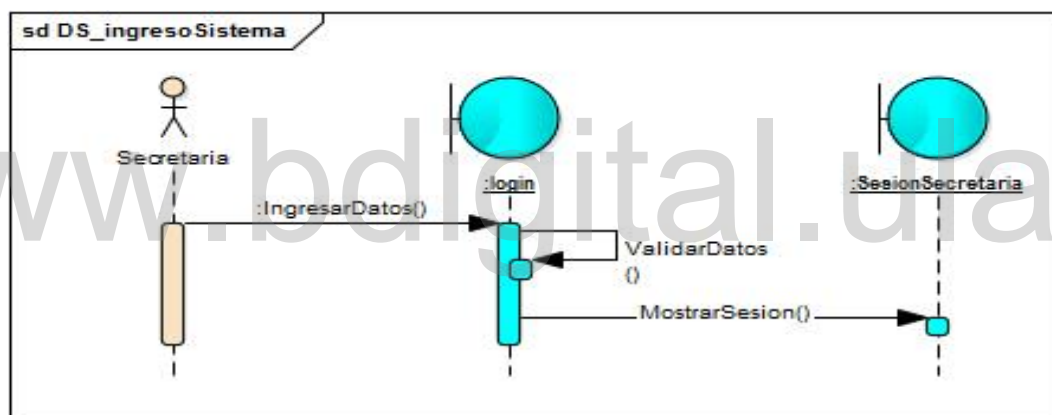


Figura C.1: Diagrama de Secuencia de Ingreso al Sistema.

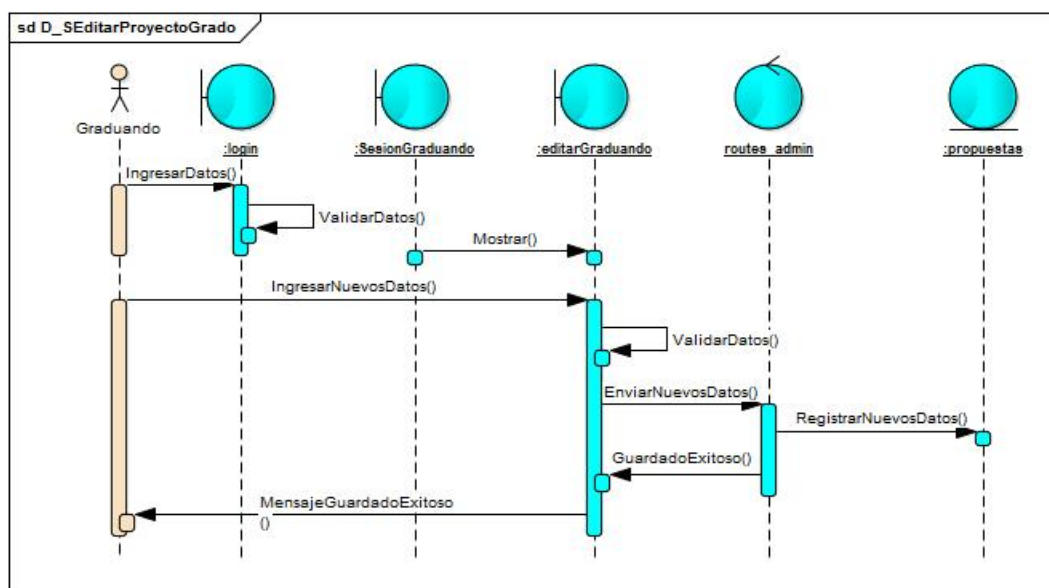


Figura C.2: Diagrama de Secuencia Edición de Propuesta de Proyecto de Grado.

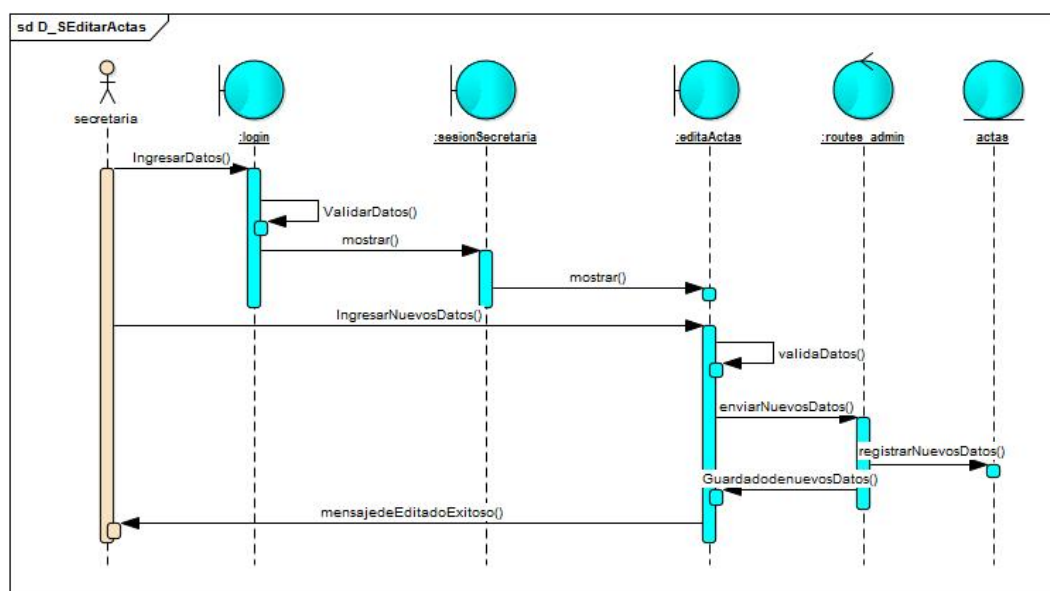


Figura C.3: Diagrama de Secuencia Edición de Actas.

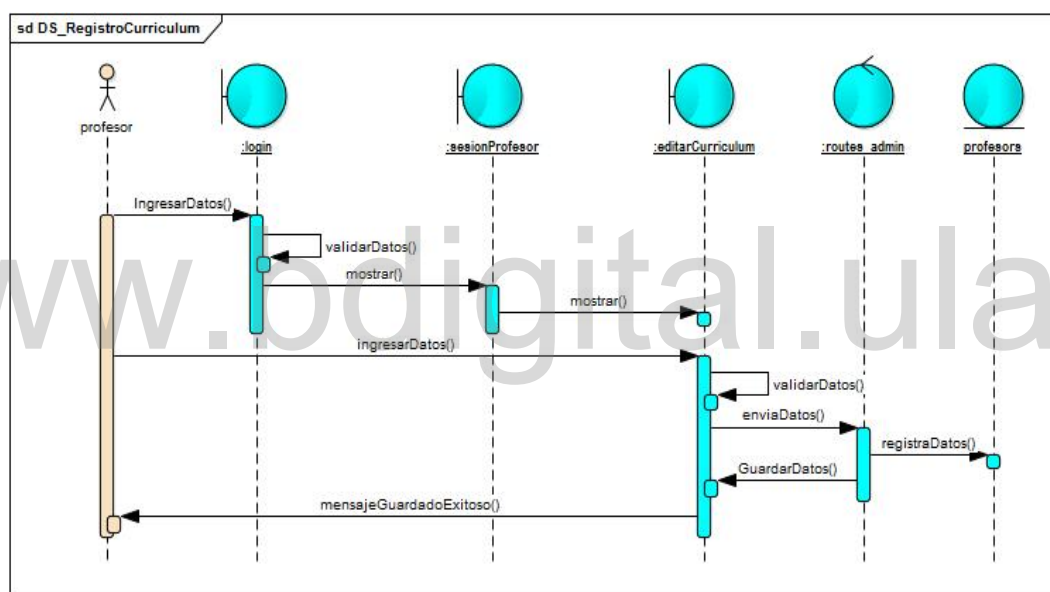


Figura C.4: Diagrama de Secuencia Registro Currículum Profesor.

Anexo D

Ensamblaje de Componentes

D.1. Procedimientos para el manejo de datos relacionados con profesores:

The screenshot displays a web application interface for the 'INGENIERÍA DEPARTAMENTO' of the 'UNIVERSIDAD DE LOS ANDES'. The page is titled 'REGISTRO DE PROFESOR' and includes a 'Volver' link. The main section is 'DATOS PERSONALES', which contains several input fields: 'Nombres', 'Apellidos', 'Correo Electrónico', 'Número de Cubículo', 'Estatus Profesor', and 'PDF Curriculum Vitae'. A 'Buscar' button is located next to the 'PDF Curriculum Vitae' field. Below these fields is a 'Proyectos de Grado' section with a text area. A 'GUARDAR' button is positioned at the bottom of the form. The footer of the page features the 'ula' logo, a list of 'Enlaces De Interés' (Universidad De Los Andes, Facultad de Ingeniería, Escuela de Sistemas), social media icons for Facebook, Twitter, and LinkedIn, and a copyright notice: '©2016 Departamento de Computación. Todos los Derechos Reservados.'

Figura D.1: Registro de Profesor.

The screenshot displays a web interface for the Department of Computing at the University of Los Andes. The header includes the department's name and contact information. The main content area is titled 'INFORMACIÓN PROFESOR' and contains a table with the following data:

Nombres	Apellidos	Correo Electrónico	Cubiculo	Estatus
ALEJANDRO	MUJICA	ALEJANDROMUJICA@ULA.VE	s04	Activo

Below the table, there are two buttons: 'CURRICULUM' and 'TEMAS PROYECTO'. A search bar is located below the buttons. The footer section includes the ULA logo, a list of links of interest (Universidad De Los Andes, Facultad de Ingeniería, Escuela de Sistemas), and social media icons for Facebook, Twitter, and LinkedIn. The copyright notice at the bottom reads: '©2016. Departamento de Computación. Todos los Derechos Reservados.'

Figura D.2: Información Profesor.

D.2. Procedimientos para el manejo de datos relacionados con preparadores:

The screenshot displays a web interface for the 'REGISTRO PREPARADOR' (Preparator Registration) form. The header includes the ULA logo and navigation links: 'Cuentas De Usuario', 'Consulta', 'Cargar En Sistema', and 'Inicio'. The form is titled 'DATOS PERSONALES' and contains several input fields: 'Nombres', 'Apellidos', 'Cédula de Identidad', 'Correo Electrónico', 'Número de Teléfono', and 'Fecha de Contrato DD/MM/YYYY'. A 'GUARDAR' button is located below the 'Número de Teléfono' field. The footer features the ULA logo, 'Enlaces De Interés' (University of Los Andes, Faculty of Engineering, School of Systems), social media icons, and a copyright notice: '©2016, Departamento de Computación. Todos los Derechos Reservados.'

Figura D.3: Registro de Preparador.

INGENIERÍA DEPARTAMENTO
UNIVERSIDAD DE LOS ANDES DE COMPUTACIÓN
FACULTAD DE INGENIERÍA ESCUELA DE SISTEMAS

Personal Proyectos De Grado Agendas Y Actas Documentos Noticias

INFORMACIÓN PREPARADOR [Volver](#)

Nombres	Apellidos	Correo Electrónico
NERIO	ALBARRAN	NERIO@ULA.VE

Buscar...

ula

Enlaces De Interés

- Universidad De Los Andes
- Facultad de Ingeniería
- Escuela de Sistemas

©2016, Departamento de Computación. Todos los Derechos Reservados.

Figura D.4: Vista de Información de Preparadores

D.3. Procedimientos para el manejo de datos relacionados con documentos:

INGENIERÍA DEPARTAMENTO
UNIVERSIDAD DE LOS ANDES DE COMPUTACIÓN
FACULTAD DE INGENIERÍA ESCUELA DE SISTEMAS

Cuentas De Usuario Consulta Cargar En Sistema Inicio

REGISTRO DE DOCUMENTOS [Volver](#)

INFORMACIÓN DOCUMENTOS

Nombre del Documento a Registrar Archivo PDF del Documento a Registrar [Buscar](#)

[GUARDAR](#)

ula

Enlaces De Interés

- Universidad De Los Andes
- Facultad de Ingeniería
- Escuela de Sistemas

©2016, Departamento de Computación. Todos los Derechos Reservados.

Figura D.5: Registro de Documentos.

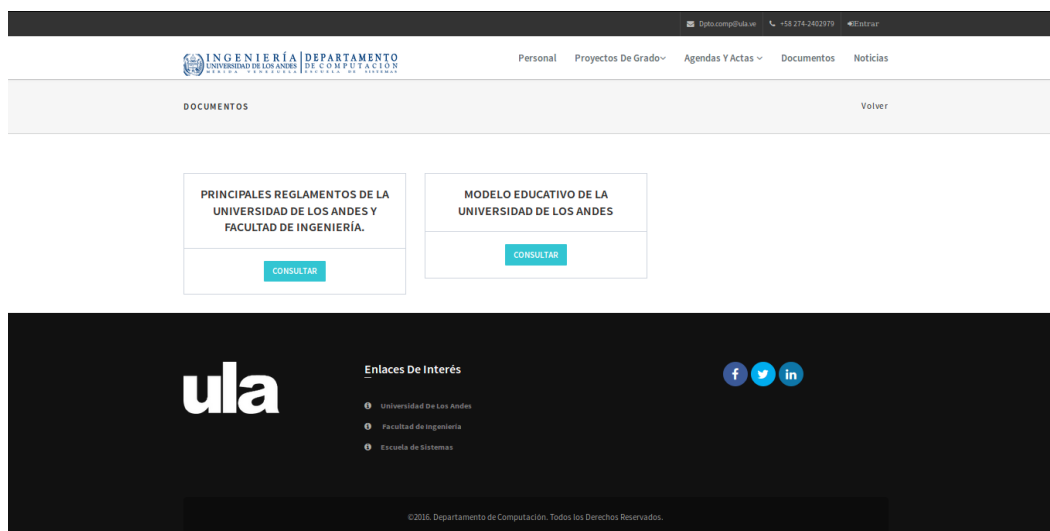


Figura D.6: Consulta de Documentos.

D.4. Procedimientos para el manejo de datos relacionados con actas:



Figura D.7: Registro de Actas.

INFORMACIÓN ACTAS

Volver

Número de Acta	Fecha de Acta
ACTA01	09/02/2017

CONSULTAR

Buscar...

1 2 3 4 5 6

ula

Enlaces De Interés

- Universidad De Los Andes
- Facultad de Ingeniería
- Escuela de Sistemas

© 2015 Departamento de Computación, Telecomunicaciones y Redes de Computación

Figura D.8: Consulta de Actas.

Anexo E

Instalación y Creación de una Base de Datos en MongoDB

Haciendo Referencia al proyecto realizado por Márquez Marielb en el año 2016. *Para hacer uso de la misma, fue necesario seguir una serie de pasos para la instalación tanto en el computador donde se trabajo localmente como en el servidor donde actualmente está funcionando el sistema de software. Los paquetes que componen una distribución de mongo son:*

- ***mongodb-org*** Este paquete instalará automáticamente los paquetes restantes.
- ***mongodb-org-server*** Contiene el demonio mongod, la configuración asociada y scripts de inicialización. Es el proceso que se encarga del funcionamiento de las tareas propias de la base de datos. Estará presente en cada máquina en la que corra la base de datos, ya sea en una máquina aislada o como parte de una Replica Set.
- ***mongodb-org-mongos*** Contiene el demonio mongos. Este proceso será necesario en el caso de que en nuestra instalación tengamos un Sharded Cluster. Es el responsable de enrutar las peticiones del cliente a la máquina que contiene los datos solicitados y también, de devolverlos. Es capaz, incluso, de realizar algunas operaciones sobre ellos antes de hacerlos llegar al cliente.

- ***mongodb-org-shell*** Paquete que contiene la mongo shell con la que nos conectamos a la base de datos, para tareas administrativas fundamentalmente
- ***mongodb-org-tools*** Conjunto de utilidades que permiten: exportar pequeñas cantidades de datos a chero (*mongoexport*), importar datos desde chero (*mongoimport*), hacer backups (*mongodump*), restaurar backups (*mongorestore*), comprobar el estado de una instancia (*mongod* o *mongos*), etcétera.

Para la instalación Desde la terminal se hace:

1. *Importamos la clave pública de MongoDB: Esta clave es necesaria para que el gestor de paquetes de Debian compruebe la consistencia y autenticidad requerida a los paquete.*

```
sudo apt-key adv keyserver hkp://keyserver.ubuntu.com:80 recv EA312927
```

2. *Generamos un chero que contendrá la url del repositorio de MongoDB:*

```
echo "deb http://repo.mongodb.org/apt/debian wheezy/mongodb-org /3.2 main"
/ sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list
```

3. *Refrescamos la base de datos local de paquetes:*

```
sudo apt-get update
```

4. *Instalamos la última versión estable de MongoDB, y todos los paquetes necesarios, en nuestro sistema:*

```
sudo apt-get install -y mongodb-org
```

5. *Iniciar el servidor de mongo:*

```
mongod
```

6. *Abrimos una shell de MongoDB:*

```
mongo
```

Para la creación de una base de datos dentro de mongodb solo es necesario insertar un dato en una colección y ella automáticamente creara la base de datos y la colección a la que se le inserta el dato:

use nombreBaseDatos

db.nombreColeccion.insert({campoNuevo: "informacionCampo"})

www.bdigital.ula.ve

Anexo F

Implementación en Servidor

Implementación realizada por Márquez Marielb en el año 2016 para el Proyecto de Grado "Desarrollo de un sistema de recuperación multimedia basado en manejadores de bases de datos NoSQL"

F.1. *Instalación de Node.js*

Hay varias maneras de trabajar con mongo, con php, por ejemplo, pero la mejor forma de trabajar con el mismo es haciendo uso de Node.js, el cual se explicó en el capítulo anterior. Node.js aprovecha al máximo las funciones de MongoDB, se comunican a través de la librería mongoose y se instaló de la siguiente manera:

F.2. *Instalando NVM*

NVM significa Node Version Manager o Gestor de Versiones de Node, permite instalar múltiples versiones de Node.js y npm en un mismo ambiente, dejando cada una completamente aislada de las otras. Puedes tener Node v0.4 y v0.11 instaladas en tu computador y ejecutar una u otra según lo necesites. Lo primero que se necesita es:

- *curl: necesario para descargar el script de instalación.*
- *build-essential: necesario en caso de que vayamos a compilar Node.js desde el código fuente.*

- *libssl-dev: ídem build-essential.*
- *git: necesario porque el script de instalación descarga NVM desde un repositorio en Github* **`sudo apt-get install -y curl build-essential libssl-dev git`**

Ir al repositorio de NVM en Github y copiar la línea del instalador.
`https://github.com/creationix/nvm.git`

- *Luego se hace curl* **`https://raw.githubusercontent.com/creationix/nvm/v0.10.0/install.sh | sh`**

Una vez terminada la instalación se cierra la terminal y se vuelve a abrir.

F.3. Instalando Node.js

Una vez instalado NVM hacemos: `nvm install 4` save Esto instalará la versión 4 de Node.js y la última versión de npm. Existe otra forma de instalar Node.js, la cual también fue probada y también funciona es:

- *Actualizar paquetes: Para iniciar la instalación de cualquier paquete debemos siempre actualizar nuestros repositorios, esto lo hacemos con el siguiente comando: **`sudo apt-get update sudo apt-get upgrade`***
- *Instalar Curl: Durante el proceso de instalación necesitaremos este binario. Para instalarlo debemos ejecutar el siguiente comando: **`sudo apt-get install curl`***
- *Añadir los repositorios de NodeSource: Debian Wheezy mediante los backports provee una versión de Node.js sin embargo no existe un binario oficial para instalar NPM, lo cual hace que debamos recurrir a otros repositorios para solventar esta herramienta sumamente importante. Para añadir estos repositorios debemos ejecutar el siguiente comando en nuestra terminal como usuario root: **`sudo su curl -sL https://deb.nodesource.com/setup`***

F.4. Instalando Express

sudo npm install -g express

En la terminal se ejecuta express y se crean una serie de carpetas y archivos donde será desarrollado del sistema. Se procede a abrir la carpeta en la cual fue ejecutado el mismo, se verá que ha sido creado un conjunto de carpetas y archivos, donde ha de ser desarrollado el sistemaNode.js.

La estructura de carpetas y archivos es la siguiente. Tenemos 5 carpetas: bin, node_modules, public, routes y views.

- *bin: Es una carpeta de asuntos generales. Aquí puedes guardar los archivos que contengan funciones y cosas de uso general.*
- *node_modules: Aquí es donde el NPM guarda todos los módulos de los que depende el proyecto de Express Generator. Mejor que no toques nada, eso es territorio del NPM.*
- *public: Esta carpeta estará accesible públicamente, por lo que es el sitio ideal para los CSS, las imágenes, los JS (los que se usen en el navegador) o archivos descargables.*
- *routes: Aquí se describen todas las posibles rutas accesibles de la web. Es una carpeta porque pueden organizarse en distintos archivos para mejor organización.*
- *views: Este es el sitio para las plantillas de las páginas y serán usadas por el motor de plantillas elegido, que en nuestro caso ha sido Jade.*