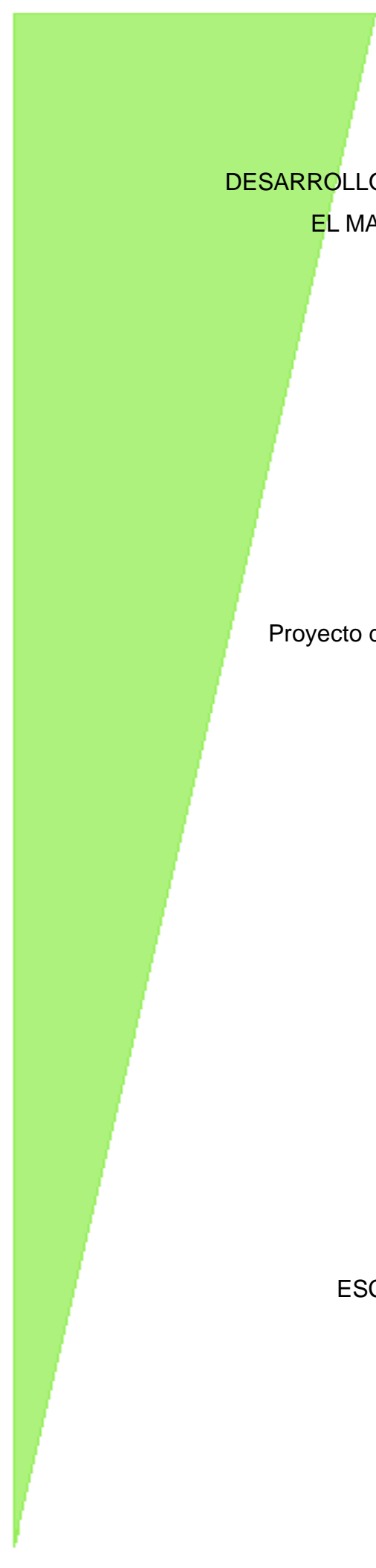


DESARROLLO DE LA NUEVA VERSIÓN DEL SISTEMA DE INFORMACION  
PARA EL MANEJO DE LA INFRAESTRUCTURA DE LA RED DE DATOS  
INSTITUCIONAL

LEIDY JOHANNA LIZARAZÚ BASTO  
MARÍA JEIMY MESA GÓMEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2011



DESARROLLO DE LA NUEVA VERSIÓN DEL SISTEMA DE INFORMACION PARA  
EL MANEJO DE LA INFRAESTRUCTURA DE LA RED DE DATOS  
INSTITUCIONAL

LEIDY JOHANNA LIZARAZÚ BASTO  
MARÍA JEIMY MESA GÓMEZ

Proyecto de grado presentado como requisito parcial para optar el título de  
Ingeniería de Sistemas

Modalidad de trabajo de Investigación

Director

EMILIO JUSTINIANO CÁRCAMO TORCONIS

Codirector

ENRIQUE TORRES LÓPEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2011

Todo sacrificio que hacemos para lograr nuestros propios objetivos y retos,  
nos fortalece en nuestra vida, para luchar y nunca desfallecer.

Agradezco a Dios por permitirme culminar este proyecto, aunque  
con varios tropiezos y caídas, pero eso hace parte de nuestra vida.

A mis padres, hermanas y novio que estuvieron ahí en las buenas y en las malas,  
apoyándome, dándome fuerzas y voces de aliento  
cuando más lo necesite, los amo y les dedico este proyecto.

A mis compañeros que estuvieron apoyándome en la causa, mil gracias!

Leidy Johanna Lizarazú Basto

Las palabras no alcanzan para describir lo que siento,  
por lo cual haré mi mejor esfuerzo.

A todos aquellos que me han formado.  
Especialmente a mis padres quienes han  
sido el motivo especial para esforzarme  
y lograr todo lo que me he propuesto.

A mis profesores, familiares, compañeros y amigos,  
de los cuales aprendí cada día que compartimos.

A mis hermanos que junto a mis padres  
han sido de gran apoyo durante toda mi vida,  
y quienes siempre han creído en mí.

A mi compañera de trabajo con la cual compartí gratos momentos,  
por su alegría e infinita paciencia.

A mis compañeros universitarios con los cuales compartimos años de estudio.

A Diosito,  
quien ha sido el guía durante toda mi vida y quien hace que esto sea hoy posible.

Gracias... Totales!

María Jeimy Mesa Gómez

## AGRADECIMIENTOS

A LA DIVISIÓN DE SERVICIOS DE INFORMACION DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER, que nos dio la oportunidad de desempeñarnos en un ambiente de investigación y desarrollo de alta calidad, relacionarnos con diferentes profesionales involucrados en el proyecto y por la experiencia en el campo laboral.

Al Ingeniero Emilio Cárcamo, director de proyecto, quien nos ayudo, guió y enseñó muchos de sus conocimientos en el área, para finalizar con éxito este proyecto.

Al Ingeniero Enrique Torres co-director del proyecto, por la experiencia transmitida y quien confió en nosotros para el desarrollo de este proyecto.

Al Ingeniero Humberto Díaz, por su colaboración y experiencias vividas en el desarrollo de software.

A los profesores de la Escuela de Ingeniería de Sistemas e Informática de la Universidad por todos los conocimientos transmitidos a lo largo de toda nuestra vida universitaria.

A los usuarios de SIMANRED por el apoyo y colaboración en la obtención de los requisitos del desarrollo del sistema de información y para los cuales fue diseñado esté.

***A Diosito por la vida y por la oportunidad de terminar con éxito este proyecto.***

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	18
TÍTULO .....	20
1 CONTEXTO GENERAL .....	21
1.1 DEFINICIÓN DEL PROBLEMA .....	21
1.2 OBJETIVOS.....	21
1.2.1 Objetivo General .....	21
1.2.2 Objetivos Específicos .....	22
1.3 JUSTIFICACIÓN.....	23
2 MARCO TEÓRICO .....	24
2.1 ESTADO DEL ARTE .....	24
2.2 GENERALIDADES DE RED DE DATOS .....	24
2.2.1 ¿Qué es una Red de Datos?.....	24
2.3 GENERALIDADES DEL ENTORNO DESARROLLO .....	27
2.3.1 Aplicativos Orientados a la web .....	27
2.3.2 Diseño UML.....	29
2.3.2.1 Diagramas de Casos de uso.....	30
2.3.2.2 Diagrama de Clases .....	33
2.3.2.3 Diagrama de Secuencia .....	36
2.3.3 JAVA ENTERPRISE EDITION (JAVA EE) .....	38
2.3.3.1 Seguridad .....	39
2.3.3.2 Soporte para servicios web.....	39
2.3.3.3 Ensamble y despliegue de una aplicación Java EE.....	40

2.3.3.4	Empaquetado de aplicaciones.....	40
2.3.3.5	Componentes Java EE .....	41
2.3.3.6	Contenedores JAVA EE .....	42
2.3.4	TECNOLOGÍA SERVLET JAVA .....	43
2.3.4.1	Servlets.....	43
2.3.5	JAVASERVER FACES .....	45
2.3.5.1	Beneficios de la Tecnología JavaServer Faces .....	46
2.3.6	SEAM.....	47
2.3.7	JBOSS ENTERPRISE APPLICATION PLATAFORM .....	50
2.3.7.1	Características y beneficios.....	50
2.3.8	ENTERPRISE ARCHITECT .....	51
2.3.9	EJB 3.0.....	52
2.3.10	JPA.....	52
2.3.11	JPQL.....	53
2.3.12	CONSTRUCCIÓN DE PROTOTIPOS .....	53
2.3.12.1	¿Qué es un prototipo? .....	53
2.3.12.2	Prototipo No funcional .....	55
2.3.13	Axure RP Pro 5.6 .....	55
3	METODOLOGÍA DE DESARROLLO .....	57
3.1	CICLO DE VIDA DEL PROYECTO.....	57
3.1.1	Análisis de Requerimientos.....	57
3.1.2	Diseño .....	58
3.1.3	Implementación de la Aplicación .....	59
3.1.4	Pruebas del software.....	59

3.1.5	Ajustes .....	59
3.2	METODOLOGIA DE DESARROLLO DEL PROYECTO .....	60
3.2.1	Modelo de construcción por prototipos.....	60
3.2.2	Estructura del Modelo de Construcción de Prototipos.....	61
3.2.3	Procedimiento a seguir para la metodología planteada .....	63
3.3	APLICACIÓN DE LA METODOLOGÍA .....	64
3.3.1	DIAGRAMAS UML .....	64
3.3.1.1	Diagrama de Casos de Uso.....	64
3.3.2.1	Diagramas de Clases.....	70
3.3.2.2	Diagrama de Secuencias .....	77
4	PROTOTIPOS .....	80
4.1	PROTOTIPO NO FUNCIONAL.....	80
4.1.1	Primer prototipo .....	80
4.1.2	Segundo prototipo.....	84
4.2	PROTOTIPO FUNCIONAL.....	86
4.3	ESQUEMA DE SEGURIDAD UNIVERSIDAD INDUSTRIAL DE SANTANDER .....	93
4.3.1	Estructura de la Base de Datos soporte.....	94
4.3.2	Entorno de Navegación.....	97
4.3.3	Entorno de Control de Datos .....	97
4.3.4	Auditoría.....	98
4.3.5	Estándares de la DSI.....	98
4.3.6	Documentación básica del programa Fuente de un EJB de Sesión.....	100
5.	CONCLUSIONES.....	103

6. RECOMENDACIONES .....	105
7. BIBLIOGRAFIA .....	106

## LISTA DE FIGURAS

<i>Figura 1. Red de Área Local .....</i>	<i>25</i>
<i>Figura 2. Red de Área Amplia o Extensa.....</i>	<i>26</i>
<i>Figura 3. Arquitectura cliente_servidor.....</i>	<i>28</i>
<i>Figura 4. Actor .....</i>	<i>31</i>
<i>Figura 5. Caso de Uso.....</i>	<i>31</i>
<i>Figura 6. Tipo de relaciones Caso de Uso .....</i>	<i>33</i>
<i>Figura 7. Representación de una Clase .....</i>	<i>34</i>
<i>Figura 8. Tipos de Relaciones entre Clases .....</i>	<i>36</i>
<i>Figura 9. Elementos del diagrama de secuencia.....</i>	<i>37</i>
<i>Figura 10. Estructura de un módulo web .....</i>	<i>41</i>
<i>Figura 11. Modelo de construcción de prototipos.....</i>	<i>61</i>
<i>Figura 12. Estructura del modelo.....</i>	<i>62</i>
<i>Figura 13. Casos de Uso SIMANRED.....</i>	<i>65</i>
<i>Figura 14. Diagrama de Clases SIMANRED (Fragmento).....</i>	<i>71</i>
<i>Figura 15. Diagrama de Secuencia – Caso de Uso Registrar Centro de Cableado.....</i>	<i>78</i>
<i>Figura 16. Diagrama de Secuencia – Caso de Uso Registrar Elementos del Centro de Cableado .....</i>	<i>79</i>
<i>Figura 17. Crear sede (Primer Prototipo).....</i>	<i>81</i>
<i>Figura 18. Maestro – Detalle (Primer Prototipo).....</i>	<i>81</i>
<i>Figura 19. Formulario detalle sede (Primer Prototipo).....</i>	<i>82</i>
<i>Figura 20. Formulario de editar sede (Primer Prototipo).....</i>	<i>83</i>
<i>Figura 21. Formulario de eliminar sede (Primer Prototipo) .....</i>	<i>84</i>
<i>Figura 22. Formulario Crear Equipo Comunicación (Primer Prototipo) .....</i>	<i>85</i>
<i>Figura 23. Listado Equipo Comunicación (Primer Prototipo) .....</i>	<i>86</i>
<i>Figura 24. Resumen de las acciones (Primer Prototipo) .....</i>	<i>86</i>
<i>Figura 25. Menú Estructura Física (Prototipo Final).....</i>	<i>88</i>
<i>Figura 26. Formulario crear sede (Prototipo Final) .....</i>	<i>88</i>
<i>Figura 27. Menú Estructura de Telecomunicaciones (Prototipo Final).....</i>	<i>89</i>
<i>Figura 28. Formulario Centro Telecomunicaciones (Prototipo Final) .....</i>	<i>90</i>
<i>Figura 29. Formulario de Crear Centro de telecomunicación (Prototipo Final).....</i>	<i>91</i>
<i>Figura 30. Menú Centro Telecomunicaciones (Prototipo Final) .....</i>	<i>91</i>
<i>Figura 31. Formulario de Modificar Centro (Prototipo Final) .....</i>	<i>92</i>
<i>Figura 32. Formulario Agregar centro Derivado (Prototipo Final) .....</i>	<i>93</i>
<i>Figura 33. Código Fuente.....</i>	<i>102</i>

## LISTA DE TABLAS

<i>Tabla 1. Detalle del caso de uso - Registrar Dispositivo .....</i>	<i>66</i>
<i>Tabla 2. Detalle del caso de uso - Registrar Equipo de Comunicación .....</i>	<i>67</i>
<i>Tabla 3. Detalle del caso de uso – Administrar Vlan .....</i>	<i>69</i>
<i>Tabla 4. Atributos clase Dispositivo .....</i>	<i>72</i>
<i>Tabla 5. Atributos clase Dispositivo .....</i>	<i>73</i>
<i>Tabla 6. Atributos clase Equipo Comunicación .....</i>	<i>74</i>
<i>Tabla 7. Atributos clase Ip Asociada .....</i>	<i>76</i>

## TERMINOS Y DEFINICIONES

**JAVA** lenguaje de Programación que se caracteriza por tener una arquitectura que permite que el código escrito funcione en multitud de sistemas operativos sin ser modificado

**SISTEMA DE INFORMACIÓN** aplicación comercial para el computador. Está constituida por la base de datos, los programas de aplicación, los procedimientos manuales y automatizados, e incluye los sistemas computacionales que realizan procesamiento.

**MÉTODO** es una operación que define como se comporta un objeto.

**OBJETO** se define como la unidad que en tiempo de ejecución realiza las tareas de un programa. También a un nivel más básico se define como la instancia de una clase.

**CLASE** una clase de objetos, describe un grupo de objetos con propiedades similares, con relaciones comunes entre otros y con una semántica común.

**SIMANRED** sistema de Información para el manejo de la infraestructura de la red de datos institucional.

**UAA** unidad Académica y Administrativa de la Universidad Industrial de Santander.

## RESUMEN

### Titulo:

DESARROLLO DE LA NUEVA VERSIÓN DEL SISTEMA DE INFORMACION PARA EL MANEJO DE LA INFRAESTRUCTURA DE LA RED DE DATOS INSTITUCIONAL \*

### Autores

LIZARAZÚ BASTO, Leidy Johanna\*\*

MESA GÓMEZ, María Jeimy\*\*

### Palabras clave

Simanred, Redes, JavaEE5, Enterprise, Ingeniería del software.

### Resumen

El presente trabajo de grado es presentado a la Universidad Industrial de Santander (UIS), con el fin de sustituir el antiguo Sistema de Información (SIMANRED) que es muy limitado en cuanto a funcionalidades y no cumple con las necesidades actuales que la UIS presenta en la gestión de su infraestructura de red de datos, siendo poco funcional y con un periodo de usabilidad caducado (culminación de su ciclo de vida); debido a esto, se desarrollo una nueva versión del Sistema de Información para el manejo de la infraestructura de la red de datos Institucional orientado a la Web y con calidad a nivel empresarial, siendo este el objetivo general de este proyecto, buscando así manejar, manipular, consultar, eliminar, registrar y modificar de forma adecuada la información que suministre el sistema supliendo las necesidades actuales de la UIS en cuanto a la administración de la infraestructura de la red de datos, para esto es necesario realizar el análisis de requerimientos, diseño e implementación de la nueva versión del sistema de la red de datos institucional.

Se muestra el diseño y desarrollo de una herramienta software robusta y moderna, que cuenta con toda la documentación del código fuente y diagramas UML. Se ha realizado bajo la asesoría de la División de Servicios de Información (DSI), brindando el material de soporte para el desarrollo (Estándares) y las capacitaciones necesarias.

---

\* Proyecto de Grado en la modalidad de Investigación.

\*\* Facultad de Ingenierías Físico-Mecánicas. Ingeniería de Sistemas e Informática. Director: Emilio Justiniano Cárcamo Troconis. Codirector: Enrique Torres López

## **ABSTRACT**

### **Title:**

DEVELOPMENT OF THE NEW VERSION OF THE INFORMATION SYSTEM FOR THE MANAGEMENT OF THE INFRASTRUCTURE OF THE INSTITUTIONAL DATA NETWORK.\*

### **Authors**

LIZARAZÚ BASTO, Leidy Johanna\*\*  
MESA GÓMEZ, María Jeimy\*\*

### **Keywords**

Simanred, Networks, JavaEE5, Enterprise, Software engineering.

### **Abstract**

This graduate work is presented to the Industrial University of Santander (UIS) to replace the information system for managing the infrastructure of data network (SIMANRED) previous, because It had very functionality limited and doesn't meet the current needs in the UIS for management the infrastructure of data network, It's little functional and Its the usability period has expired ( culmination of their life cycle), because of this, we developed a new version of the information system for managing the infrastructure of the Institutional Data Network with oriented Web and business quality level, this being the overall objective of this project, thus seeking to manage, manipulate, query, delete, record and appropriately modify the information that provide the system, satisfying the current needs of the UIS in the administration of network infrastructure data, for this is necessary to perform the requirements analysis, design and implementation of the new version of Information system of network of data institutional.

It shows the design and development of a modern and robust software tool that has all the source code documentation and UML diagrams. It's was performed under the guidance of the Information Services Division (DSI) providing material support for development (Standards) and the necessary training.

---

\* Degree Project in the modality Research.

\*\* Physical-Mechanic Engineering Faculty. Systems and Informatics Engineering. Director: Emilio Justiniano Cárcamo Troconis. Codirector: Enrique Torres López

## INTRODUCCIÓN

---

La era actual se caracteriza por fenómenos como la globalización, aumento de la competitividad organizacional y por el auge de las tecnologías de la información y telecomunicaciones; donde la información juega un papel importante en las actividades sociales, culturales y económicas, hasta tal punto de ser llamada “sociedad de la información”<sup>1</sup>. Luego, el conocimiento se ha convertido en el factor fundamental de creación de riqueza en la sociedad. Las organizaciones generan grandes volúmenes de datos, producto de actividades y procesos, luego, se hace necesario el uso de sistemas de información que los procesen, con el objetivo de proporcionar información útil, rápida y veraz que ayude a la optimización de las labores y a la toma de decisiones.

En ese orden de ideas, las instituciones de educación superior cuentan con sistemas de información para la administración de sus actividades académicas, financieras, manejo de personal, infraestructura de datos, entre otras.

La División de Servicios de Información de la Universidad Industrial de Santander, encargada de la administración y el desarrollo de las TI en los ámbitos académico y administrativo, ha venido desarrollando diferentes proyectos con el fin de mejorar los sistemas de información institucionales, actualizándolos con tecnologías líderes en el mercado, y de esta forma mejorar los procesos internos de la Universidad.

En este trabajo de grado se plantea el diseño, desarrollo e implementación de la nueva versión del sistema de información para el manejo de la red de datos institucional SIMANRED, con el objetivo de optimizar el control y manejo de la

---

<sup>1</sup> Mattelart, Armand (2002b), Historia de la sociedad de la información, Piados, Barcelona, 193 p.

infraestructura de red de la universidad, de acuerdo a las nuevas necesidades tecnológicas y de funcionamiento que demanda la institución.

Este documento se encuentra dividido en siete (7) capítulos. En el segundo capítulo se describe el marco teórico (generalidades de la red de datos y generalidades del entorno de desarrollo). En el capítulo tres se explica la metodología tenida en cuenta para el desarrollo de este proyecto, en el capítulo cuatro se muestran los diferentes prototipos elaborados, además se especifica el esquema de seguridad manejado para los sistemas de información de la UIS, en el capítulo cinco se encuentran depositadas las conclusiones y en el capítulo 6 las recomendaciones, y en último capítulo la bibliografía.

## TÍTULO

DESARROLLO DE LA NUEVA VERSIÓN DEL SISTEMA DE INFORMACION  
PARA EL MANEJO DE LA INFRAESTRUCTURA DE LA RED DE DATOS  
INSTITUCIONAL

*NOMBRE:* Ing. Emilio Cárcamo Troconis

*INSTITUCIÓN:* Universidad Industrial de Santander

*CARGO:* Profesional adscrito a la División de Servicios de Información.

*NOMBRE:* Ing. Enrique Torres López

*INSTITUCIÓN:* Universidad Industrial de Santander

*CARGO:* Profesional adscrito a la División de Servicios de Información.

## AUTORES

Nombre: Leidy Johanna Lizarazú Basto

Código: 2051446

Carrera: Ingeniería de Sistemas e Informática

Nombre: Maria Jeimy Mesa Gómez

Código: 2053258

Carrera: Ingeniería de Sistemas e Informática

Entidades interesadas en el proyecto:

División de Servicios de Información

Universidad Industrial de Santander

Simanred

Universidad Industrial de Santander

# **1 CONTEXTO GENERAL**

## **1.1 DEFINICIÓN DEL PROBLEMA**

La información asociada a la infraestructura de la red de datos de la UIS se encuentra registrada en un sistema de información llamado SIMANRED, el cual ya ha cumplido su ciclo de vida. Debido a los cambios significativos en la infraestructura de datos, a la falta de mantenimiento y al avance de la tecnología, en dicho sistema se han encontrado limitaciones funcionales y de usabilidad que repercuten en la velocidad y eficiencia con que se realizan los procesos de soporte a usuario y registro de la información.

Estos inconvenientes se traducen en la desactualización de los datos que residen en el sistema, dificultando así las labores cotidianas de cualquier institución o empresa que maneja los sistemas de información. Por ende se propone una herramienta que sea agradable a los usuarios, sea útil y que la información suministrada se asemeje a la realidad del entorno para la cual fue desarrollado.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo General**

Desarrollar la nueva versión del sistema de información para el manejo de la infraestructura de la red de datos institucional, SIMANRED, aplicando las últimas tecnologías de programación web utilizadas por la universidad.

### 1.2.2 Objetivos Específicos

Realizar el análisis, diseño, implementación y puesta en funcionamiento de la nueva versión del sistema de información para el manejo de la infraestructura de la red de datos institucional, SIMANRED, que permita:

- El registro y actualización permanente de la infraestructura de la red de datos institucional asociada con los centros de comunicaciones, equipos de comunicaciones, cableado estructurado, VLANs y dispositivos conectados, información necesaria para su adecuada administración y control. Adicionalmente, el sistema propuesto soporta los futuros cambios a nivel de infraestructura y tecnología proyectados.
- Desarrollar el módulo de consultas e informes que proporcione las características de la red como:
  - ▲ Equipos de comunicación instalados.
  - ▲ IP's disponibles por subredes.
  - ▲ Centros de cableado principales y secundarios.
  - ▲ Elementos del centro de cableado.
  - ▲ Dispositivos conectados.

El contenido de estas consultas e informes será definido de acuerdo con la división de Servicios de Información.

- Establecer un esquema de seguridad basado en roles y usuarios, asociados a los roles según las actividades que realicen, que permita regular el acceso a la información y la navegación en el sistema.

### 1.3 JUSTIFICACIÓN

La Universidad Industrial de Santander, es una entidad de educación superior en continuo cambio: el aumento del número de estudiantes y de la cobertura en las diferentes sedes, la creación permanente de nuevas instalaciones y oficinas, los avances y actualización del entorno de telecomunicaciones, entre otros aspectos, hace que el trabajo de administración de la infraestructura tecnológica sea más complejo y difícil de gestionar.

La aplicación para la administración de la red, que utiliza actualmente la UIS, ha cumplido su ciclo software, y aunque funciona, se ve limitada en eficiencia y velocidad en los procesos de soporte a usuarios, dado que los tiempos de respuesta a las solicitudes no es el mejor, y no se adapta a la naturaleza dinámica de la institución. En ese orden de ideas, el nuevo Sistema de Información para el Mantenimiento de la Red (SIMANRED), surge como alternativa para gestionar de manera adecuada la red de datos de la universidad y los procesos derivados de ella, conforme a los estándares de desarrollo definidos por la División de Servicios de Información.

Los beneficios en los que podría contribuir esta investigación son los siguientes:

- Aportar información coherente, para valorar los resultados de las inversiones realizadas en la red de datos, versus los avances proyectados en los planes de desarrollo de la Universidad.
- Facilitar el mantenimiento y soporte de la red de datos institucional y la experiencia de usuario en el manejo del sistema.
- Generar informes sobre los componentes de la red de La Universidad y su ubicación en el campus.

## **2 MARCO TEÓRICO**

### **2.1 ESTADO DEL ARTE**

A comienzos del año 2000, se creó un sistema de información llamado SIMANRED para la administración de la red de datos Institucional UIS, teniendo como base el registro y mantenimiento de la información de equipos, tomas, Ips disponibles, Switches, Centros de Cableado, Racks, entre otros, los cuales hacían parte de la infraestructura de red de esa época.

La Universidad Industrial de Santander con el objetivo de actualizar los sistemas de información institucionales, se apoya en la División de Servicios de Información DSI, quien ha decidido usar las últimas tecnologías de desarrollo web para estar a la vanguardia de los sistemas de clase mundial y así satisfacer necesidades que ayuden en las labores institucionales. Es por esta razón que la DSI planteó la realización de una nueva versión de este sistema de información que brinde a los usuarios una herramienta más dinámica, moderna y de alta usabilidad.

### **2.2 GENERALIDADES DE RED DE DATOS**

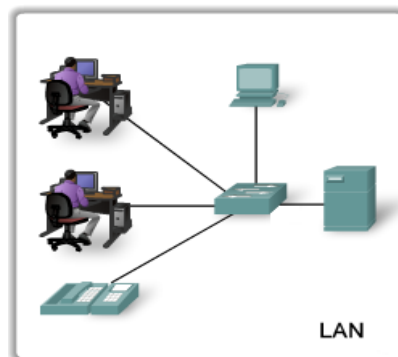
#### **2.2.1 ¿Qué es una Red de Datos?**

La sociedad actual del conocimiento ha creado la necesidad de utilizar nuevas herramientas tecnológicas de información, que impacten en todo tipo de organización. A medida que se avanza, surgen diferencias en la forma de enviar, recibir, almacenar, capturar y procesar la información; antes se utilizaba un entorno de trabajo independiente, lo cual resultaba ineficiente y lento en cuanto a las expectativas y necesidades de los usuarios de hoy día, dificultándose un poco el intercambio de información. Ahora, sin importar la localización física recurso y

del usuario, se puede llevar a cabo de una manera muy sencilla, conectando o enlazando dos o más puntos (terminales), por un medio físico, el cual sirve para enviar o recibir información, a esto se le conoce como red de datos<sup>2</sup>.

Las redes se pueden clasificar según su tamaño. Una red individual generalmente cubre una única área geográfica y proporciona servicios y aplicaciones a personas dentro de una estructura organizacional común, como una empresa, un campus o una región. Este tipo de red se denomina Red de área local (LAN). Una LAN por lo general está administrada por una organización única. El control administrativo que rige las políticas de seguridad y control de acceso está implementado en el nivel de red<sup>3</sup>.

**Figura 1. Red de Área Local**



Las zonas o ubicaciones separadas por grandes distancias geográficas se conocen como Redes de área amplia (WAN), es posible que deba utilizar un proveedor de servicio de telecomunicaciones (TSP) para interconectar las LAN en las distintas ubicaciones. Los proveedores de servicios de telecomunicaciones operan grandes redes regionales que pueden abarcar largas distancias.

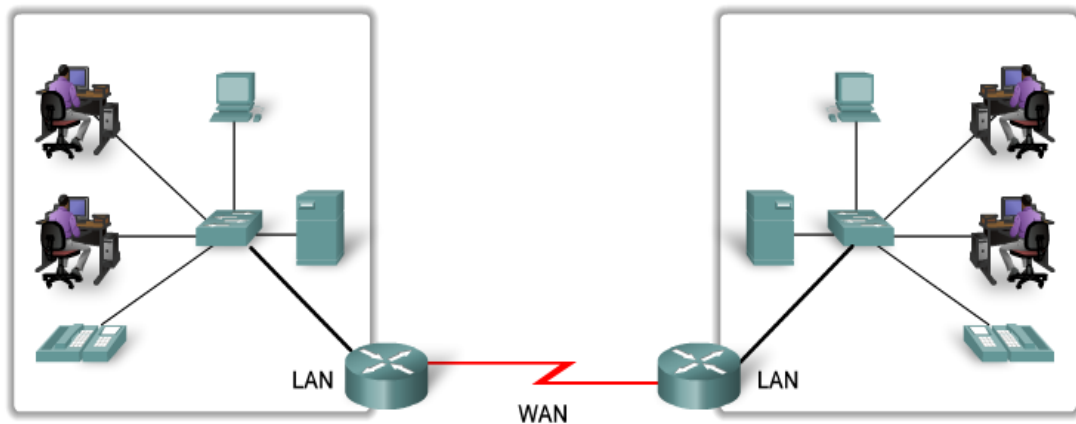
---

<sup>2</sup> Tomado de REDES DE DATOS LAN, Disponible en internet:  
[http://www.uazuay.edu.ec/estudios/electronica/proyectos/redes\\_de\\_datos\\_lan.pdf](http://www.uazuay.edu.ec/estudios/electronica/proyectos/redes_de_datos_lan.pdf)

<sup>3</sup> Tomado de CCNA Exploration, Network Fundamentals, versión 4, Capítulo 2 Comunicación a través de la red, sección 2.2.1.1 LAN (Red de Área Local).

Las WAN utilizan dispositivos de red diseñados específicamente para realizar las interconexiones entre las LAN<sup>4</sup>

Figura 2. Red de Área Amplia o Extensa



La estructura básica en una red de datos está integrada de diferentes partes:

Los **Racks** o bastidores están destinados para alojar equipos electrónicos como; servidores, conmutadores, sistemas de audio y video, enrutadores de comunicaciones o Router, cortafuegos, PachPanels, etc. Sus medidas están normalizadas para que sea compatible con equipamiento de cualquier fabricante. Los racks son útiles en los centro de cableado, donde el espacio por lo general es pequeño y se necesita alojar varios equipos.

Los **servidores** Es software y hardware que realiza tareas en nombre de varios usuarios, una maquina especializada que tiene como propósito proveer información para que otros equipos puedan utilizarla (administrador del sistema). Los Hubs, hacen la función de amplificador de señales, y en los que se encuentran conectados los nodos. Los PachPanel's, son organizadores de cables. Los Routers conectan una red con otra red. Por lo tanto, el router es responsable

---

<sup>4</sup> Tomado de CCNA Exploration, Network Fundamentals, versión 4, Capítulo 2 Comunicación a través de la red, sección 2.2.2.1 WAN (Red de Área Amplia).

de la entrega de paquetes a través de diferentes redes. La efectividad de las comunicaciones de Internet depende, en gran medida, de la capacidad de los routers de reenviar paquetes de la manera más eficiente posible<sup>5</sup>. Los Switch son dispositivo que permite la interconexión de redes sólo cuando esta conexión es necesaria, opera en la capa 2 del modelo OSI, que es el nivel de enlace de datos, y tienen la particularidad de aprender y almacenar las direcciones de dicho nivel, por lo que siempre irán desde el puerto de origen a puerto destino<sup>6</sup>. Este dispositivo se utiliza para segmentar una red en múltiples dominios de colisiones. A diferencia del hub, un switch reduce las colisiones en una LAN. Cada puerto del switch crea un dominio de colisiones individual. Esto crea una topología lógica punto a punto en el dispositivo de cada puerto. Además, un switch proporciona ancho de banda dedicado en cada puerto y así aumenta el rendimiento de una LAN. El switch de una LAN también puede utilizarse para interconectar segmentos de red de diferentes velocidades<sup>7</sup>.

## **2.3 GENERALIDADES DEL ENTORNO DESARROLLO**

### **2.3.1 Aplicativos Orientados a la web**

En todo proceso de desarrollo de software orientados a la web, se debe tener en cuenta que tipo de tecnologías se van a usar, de tal forma que el desarrollo del proyecto sea más eficiente, teniendo en cuenta el alcance de la aplicación y la población objetivo, así como las tecnologías elegidas para realizar la implementación.

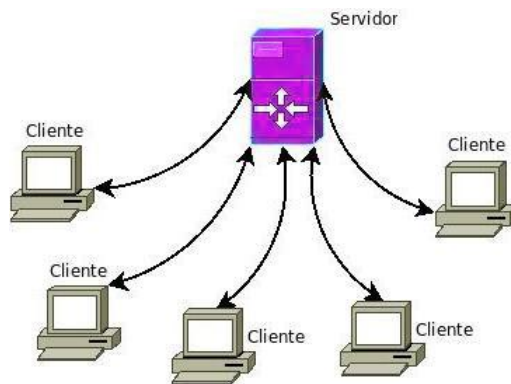
---

<sup>5</sup> Tomado de 1.0.1 INTRODUCCION DEL CAPITULO, capitulo Routingprotocols and concepts, Academyconnection Cisco.

<sup>6</sup> DEFINICION DE SWITCH, [Disponible en internet: http://www.mastermagazine.info/termino/6801.php](http://www.mastermagazine.info/termino/6801.php), <http://www.definicionabc.com/tecnologia/switch.php>.

<sup>7</sup> CCNA Exploration, Network Fundamentals, versión 4, Capítulo 10 Planificación y cableado de redes, sección 10.1.1.2 Selección de un dispositivo LAN adecuado.

Figura 3. Arquitectura cliente\_servidor



En el modelo cliente servidor, el cliente (equipos que forman parte de una red de datos) envía un mensaje solicitando un determinado servicio al servidor y este, envía uno o varios mensajes de respuesta, es decir provee un servicio. En un sistema distribuido, cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

En las aplicaciones web, el servidor generalmente provee páginas en formatos como HTML o XHTML, que son soportadas por los navegadores; el servidor envía al cliente las páginas dinámicas, provocando la sensación de una secuencia interactiva. Estos servidores que proveen aplicaciones web son denominados Servidores HTTP puesto que para realizar las interacciones entre cliente y servidor se utiliza el protocolo HTTP como puente de comunicación.

En el dominio de la redes de datos, y para los sitios Web de tercera generación, los lenguajes más utilizados para el desarrollo de páginas dinámicas son JAVA, ASP, PHP y PERL. La información generada en este tipo de sitios, está almacenada en grandes bases de datos, que permiten actualizarla, e incluso, cambiar completamente el aspecto con que se muestran. Las aplicaciones interactivas y la información en tiempo real caracterizan este tipo de desarrollos.

## **Web 2.0.**

Para entender en qué consiste la web 2.0, debemos contextualizarnos, en lo que se denomina la web 1.0, que es un sistema de distribución de información basado en hipertexto, paginas estáticas accesibles a través Internet, es decir, el usuario no interactúa con ellas, simplemente las paginas le muestran información.

La Web 2.0, a diferencia de la web 1.0, está orientada a la interacción de los usuarios con las aplicaciones, donde éste es el encargado de actualizar la información que se encuentra en internet, gracias al uso de blogs, wikis, redes sociales y servicios de alojamiento, que permiten el trabajo colaborativo y compartir la información.

Entre los ejemplos de la Web 2.0 están redes sociales como Facebook, Hi5, badoo; wikis:Wikipedia; blogs:blogspot,wordpress; y servicios de alojamiento de información como Flickr, rapidshare, hotfile,etc.

### **2.3.2 Diseño UML**

El lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*), permite diseñar, construir y documentar los elementos que forman un sistema software orientado a objetos. En el proceso de creación del UML han participado empresas como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores y se ha convertido en una herramienta de amplio uso en la industria de desarrollo software, no solo por el prestigio de sus creadores, si no porque incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE).

En la División de Servicios de Información (DSI), para documentar los sistemas de información se hace uso de 3 diagramas UML básicos:

- Diagrama de Clases
- Diagrama de Secuencias
- Diagrama de Casos de Uso

A continuación se profundiza en cada uno de ellos,

### **2.3.2.1 Diagramas de Casos de uso**

El conjunto de casos de uso al ser coherente y consistente promueve una imagen fácil del comportamiento del sistema, un entendimiento común entre el cliente/propietario/usuario y el equipo de desarrollo, de su análisis se puede concluir si el sistema cumple satisfactoriamente con los requisitos de los usuarios.

#### **Elementos del diagrama de casos de uso**

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

##### **✓ Actores**

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Los actores pueden modelar a personas, pero también se pueden corresponder con otros sistemas que se comunican con él.

Para acentuar la diferencia entre un actor y un usuario, podemos pensar que un actor es como una clase que se define por la descripción de su comportamiento. Por otro lado un usuario puede desempeñar varios papeles, es decir, puede actuar como actores diferentes. Se representa mediante una figura humana dibujada con

palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.).

Figura 4. Actor



### ✓ **Casos de Uso**

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

Figura 5. Caso de Uso



Un caso de uso especifica dos tipos de secuencias o comportamientos:

- a) Secuencia Básica o Comportamiento Normal: Son las acciones que ejecuta el actor sobre el sistema en el orden establecido en cada caso de uso.

b) *Secuencia o Comportamiento Alternativo*: Es el camino que el Actor invoca cuando este no lleva a cabo la secuencia básica en forma satisfactoria, es decir, cuando el Actor comete errores al momento de interactuar con el sistema.

### ✓ **Relaciones de Casos de Uso**

Existen relaciones entre los casos de uso, las cuales conectan los anteriores elementos en el diagrama (Actores y Casos de Uso).

- ❖ *Inclusión (include o use)*: Es una forma de interacción, un caso de uso dado puede "incluir" otro. Este uso se asemeja a una expansión de una macro, donde el comportamiento del caso incluido es colocado dentro del comportamiento del caso de uso base. No hay parámetros o valores de retorno.
- ❖ *Extensión (Extend)*: Esta relación indica que el comportamiento del caso de uso extensión puede ser insertado en el caso de uso extendido bajo ciertas condiciones."La extensión, es el conjunto de objetos a los que se aplica un concepto. Los objetos de la extensión son los ejemplos o instancias de los conceptos."<sup>8</sup>
- ❖ *Relación de Generalización entre Actores*: Se utiliza cuando un actor hereda ciertas características de otro más general.
- ❖ *Relación de Generalización entre casos de Uso*: Indica cuando un caso de uso hereda y adiciona características de otro más general.

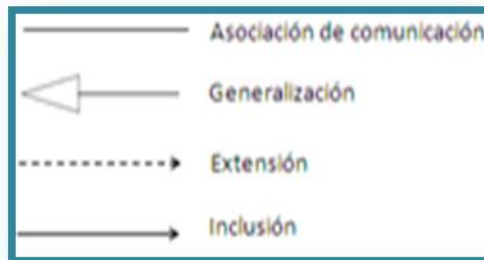
---

<sup>8</sup> Análisis y Diseño Estructurado y Orientado a Objetos de Sistemas Informáticos. Quinta Edición. Universidad Carlos III de Madrid, España. McGraw-Hill, 2003

- ❖ *Relación de Asociación:* Se utiliza para conectar un actor con un caso de uso e implica la existencia de una comunicación entre ellos.

En la siguiente figura se observa los diferentes tipos de relaciones que se presentan en un diagrama de casos de uso.

Figura 6. Tipo de relaciones Caso de Uso



### 2.3.2.2 Diagrama de Clases

Un diagrama de clases modela la vista estática del sistema, ya que no describe el comportamiento del sistema en función del tiempo. Un diagrama de clases recoge tanto los conceptos del dominio de la aplicación como aquellos que forman parte la implementación de la aplicación.

Los elementos principales son Clases y Relaciones.

#### ✓ **Clases**

Una clase es una descripción de un conjunto de objetos con las mismas propiedades (atributos), el mismo comportamiento (operaciones), las mismas

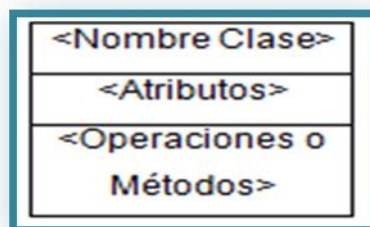
relaciones con otros objetos y la misma semántica. Una clase puede representar un concepto del mundo real (modelo de análisis) o un concepto de implantación (modelo de diseño) del sistema.

Los atributos y las operaciones de una clase pueden ser:

- Público, si todo elemento que pueda ver a la clase puede ver también el atributo u operación perteneciente a la clase. Estos se representa mediante el signo ( + ) situado delante del nombre del atributo u operación.
- Protegido, si solo pueden ver al atributo u operación indicados los elementos pertenecientes a su clase o un descendiente de su clase. Esto se representa mediante el signo ( # ) situado delante del atributo u operación.
- Privado, si solo pueden ver al atributo u operación indicados los elementos pertenecientes a esa clase. Esto se representa mediante el signo ( - ) situado delante del atributo u operación.

La siguiente figura corresponde a una clase,

Figura 7. Representación de una Clase



## ✓ **Relaciones**

Una relación es una abstracción de un conjunto de interrelaciones semánticas puntuales que se dan sistemáticamente entre diferentes tipos de objetos.

Las relaciones pueden ser de diferentes tipos:

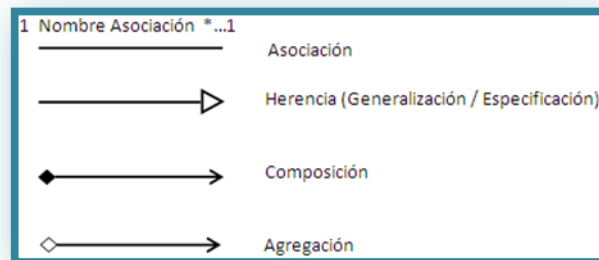
- Asociación, son las que representan relaciones entre objetos, sin un carácter especial. Constituyen una conexión semántica bidireccional entre las clases. Una asociación entre clases no es flujo de datos sino que hay un enlace entre los objetos de las clases asociadas; así pues, la asociación representa a un conjunto de enlaces.
- Agregación, es un tipo de asociación que recoge una semántica especial. Son relaciones en las que un objeto está compuesto por otros del mismo o de diferentes tipos; es decir, relaciones todo-parte (pertenece a, tiene un, es parte de). Por ejemplo, un Banco tiene clientes.
- Herencia, define una relación en una superclase y una o varias subclases en la que ambas, superclase y subclase recogen el mismo concepto, pero las subclases lo hacen con un nivel de detalle mayor que las superclases. Las subclases heredan los atributos y comportamiento de las superclases. Las subclases pueden tener más de una superclase, lo que se denomina herencia múltiple.

La herencia se puede encontrar mediante dos métodos:

- Generalización, proporciona el mecanismo para crear superclases que encapsulan los atributos y el comportamiento común de varias clases.

- Especialización, proporciona la capacidad para crear subclases que representan el mismo concepto con un mayor nivel de refinamiento y detalle de las superclases.<sup>9</sup>

Figura 8. Tipos de Relaciones entre Clases



### 2.3.2.3 Diagrama de Secuencia

Un diagrama de secuencia representa la interacción entre clases del modelo de estructuras estáticas, ordenadas temporalmente a través de tiempo y se modela para cada caso de uso.

Normalmente cada caso de uso tiene asociados varios diagramas de secuencia. Uno que representa el curso típico de funcionamiento del caso de uso y uno o más por cada posible ejecución alternativa del caso de uso, es decir, si se producen errores.

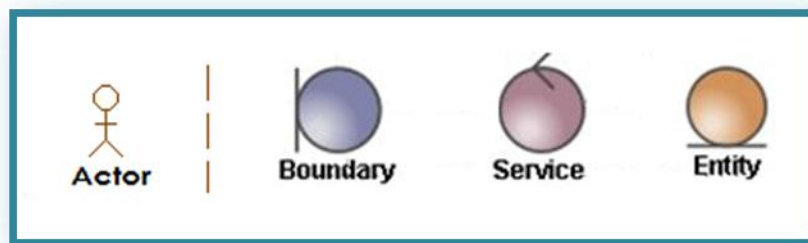
Los diagramas de secuencia son un tipo de diagrama de interacción. Se utilizan especialmente cuando se trata de sistemas en tiempo real.

<sup>9</sup> Análisis de Diseño estructurado y Orientado a Objetos de Sistemas Informáticos, Amescua, Cuadrado, Mc Graw Hill, Quinta Edición, Capítulo 2.

Una clase en el diagrama de secuencia se representa con un rectángulo en cuyo interior aparece el nombre de la clase.<sup>10</sup>

Los principales elementos que involucra un diagrama de secuencia se muestran en la siguiente figura:

Figura 9. Elementos del diagrama de secuencia



- **Actor**
- **Barra de sincronización temporal (Línea de tiempo)**
- **Boundary o Interfaz de usuario:** Son objetos “Que muestran o piden datos”. Representa una pantalla, ventana, página, reporte.
- **Service o Control:** Son objetos “Que hacen”. Son componentes que invocan, coordinan a otros objetos, realizan cálculos, consultan o modifican objetos de negocio.

---

<sup>10</sup> Análisis de Diseño estructurado y Orientado a Objetos de Sistemas Informáticos, Amescua, Cuadrado, Mc Graw Hill, Quinta Edición, Capito 4

- **Entity u Objeto de Negocio:** Son objetos “Que son”. Son los objetos detectados en el modelo de dominio, casi siempre son persistentes, es decir, se guardan en un repositorio de datos.<sup>11</sup>

### 2.3.3 JAVA ENTERPRISE EDITION (JAVA EE)

Java Enterprise Edition (Java EE) es una plataforma utilizada para desarrollar software en el lenguaje Java. La plataforma Java (Enterprise Edition) se diferencia de Java Standard Edition (Java SE) en que esta añade bibliotecas para desplegar, tolerancia a fallos, Java multi-nivel, etc. Java EE incluye varias especificaciones de API, tales como JDBC, RMI, email, JMS, servicios web, XML, etc., también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, Conectores, servlets, portlets, JavaServer, Pages y varias tecnologías de servicios web.<sup>12</sup>

Esta edición de Java hace más eficiente el desarrollo de aplicaciones Java empresariales, ya que brinda a los desarrolladores un grupo de APIs que ayudan a reducir el tiempo de desarrollo, la complejidad y el rendimiento de las aplicaciones. Introduce un modelo de programación simplificado, convirtiendo los descriptores de despliegue XML en opcionales.

La API de persistencia permite un mapeo de objetos a relaciones para manejo de datos relacionales en beans empresariales, componentes web y clientes de aplicación. También puede ser utilizado en aplicaciones Java SE, fuera del ambiente Java EE.

---

<sup>11</sup> ITURBIDE, José Luis. Técnicas de Análisis y Diseño Orientado a Objetos. Disponible en internet: [http://jiturbide.blogspot.com/2009\\_11\\_01\\_archive.html](http://jiturbide.blogspot.com/2009_11_01_archive.html)

<sup>12</sup> Enciclopedia ENCYDIA. Disponible en internet: [http://es.wikilingue.com/pt/Java\\_EE](http://es.wikilingue.com/pt/Java_EE)

### **2.3.3.1 Seguridad<sup>13</sup>**

La plataforma Java EE permite la creación de reglas estándar que son definidas por el desarrollador, para controlar el acceso en sus aplicaciones. Se implementan al desplegar la aplicación en el servidor. Java EE dispone de mecanismos de acceso predefinidos para que los desarrolladores no tengan que implementarlos. Una sola aplicación trabaja en diversos ambientes de desarrollo sin necesidad de modificar el código fuente.

### **2.3.3.2 Soporte para servicios web**

Los servicios web son aplicaciones empresariales que utilizan un estándar abierto basado en XML y protocolos de transporte para intercambiar datos entre clientes. La plataforma proporciona una API para XML y las herramientas que se necesita para diseñar, desarrollar, probar y desplegar rápidamente servicios web y clientes que operan totalmente con otros servicios web y clientes ejecutándose en plataformas basadas en Java o no.

Para escribir servicios web y clientes con las APIs de XML de Java EE todo lo que debe hacer es para los datos en los parámetros de la llamadas a los métodos y procesar los datos retornados; o para servicios web orientados a documentos, debe enviar documentos que contengan la comunicación del servicio en ambos sentidos. No se necesita programación a bajo nivel dado que la API de XML hace el trabajo de traducir los datos de la aplicación desde y hacia el flujo de datos basado en SML que es enviado a través de los protocolos estandarizados de transporte basados en XML. Estos estándares y protocolos basados en SML son introducidos en las siguientes secciones.

---

<sup>13</sup> Disponible en internet: <http://java.cabezudo.net/trabajos/JEE5/manual/jee5.v0.01.00/ababac.html#aaf>

La traducción de los datos a un flujo de datos estandarizado basado en XML es lo que hace que los servicios web y clientes escritos con las APIs de Java EE puedan operar entre ellos totalmente. Esto no necesariamente significa que los datos transportados incluyan etiquetas XML porque el transporte de los datos pueden ser texto plano, datos XML o cualquier tipo de dato binario como audio, vídeo, mapas, ficheros de programa, documentos CAD o lo que desee.<sup>14</sup>

### **2.3.3.3 Ensamble y despliegue de una aplicación Java EE**

Una aplicación Java EE es empaquetada en una o más unidades estándar para ser desplegada en cualquier sistema compatible con la plataforma Java EE. Cada unidad contiene:

- Un componente o componentes funcionales (como un bean empresarial, página JSP, servlet o Applet).
- Un descriptor de despliegue que describe su contenido.

Una vez que una unidad Java EE ha sido producida, está lista para ser desplegada. Su despliegue típicamente involucra el uso de una herramienta para especificar la información de ubicación específica, como una lista de usuarios locales que pueden acceder a esta y el nombre de la base de datos local. Una vez desplegado en una plataforma local, la aplicación está lista para ejecutarse.

### **2.3.3.4 Empaquetado de aplicaciones**

Una aplicación Java EE es distribuida en un fichero Archivo Empresarial (EAR) que es un Archivo Java estándar (JAR) con una extensión .ear. El uso de archivos

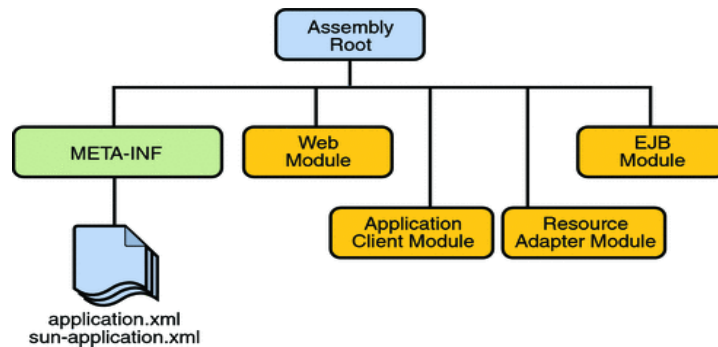
---

<sup>14,26</sup> Disponible en internet: <http://java.cabezudo.net/trabajos/JEE5/manual/jee5.v0.01.00/ababae.html>

EAR y módulos hace posible ensamblar una gran cantidad de aplicaciones, Java EE utilizando alguno de los mismos componentes. No se necesita codificación extra; es solo un tema de ensamble (o empaquetado) de varios módulos Java EE en un fichero EAR de Java EE.

Un fichero EAR contiene, como muestra la siguiente figura la estructura de un fichero EAR, módulos Java EE y descriptores de despliegue. Un descriptor de despliegue es un documento XML con una extensión .xml que describe la configuración de despliegue de una aplicación, un módulo o un componente. Dado que la información en el descriptor de despliegue es declarativa, esta puede ser cambiada sin la necesidad de modificar el código fuente. En tiempo de ejecución, el servidor Java EE lee el descriptor de despliegue y actúa sobre la aplicación, módulo o componente como corresponde.

Figura 10. Estructura de un módulo web



### 2.3.3.5 Componentes Java EE

Un componente es una unidad de software que está contenida y ensamblada en una aplicación Java EE, posee sus clases relacionadas e interactúa con los demás componentes que así lo especifiquen.

La plataforma Java EE define los siguientes componentes:

Aplicaciones cliente y Applets, estos componentes se ejecutan en el cliente.

Los componentes Web:Servlets, JavaServer Faces, y tecnología JavaServerPages™ (JSPTM); se ejecutan en el servidor.

Los componentes JavaBeans™ (EJB™) empresariales (beans empresariales), son llamados componentes de negocios y se ejecutan en el servidor.

Los componentes Java EE se desarrollan con el lenguaje de programación Java y su compilación es igual que cualquier programa en este lenguaje. A diferencia de las clases Java estándar, los componentes Java se ensamblan en una aplicación y se verifica que estén bien formados, de acuerdo a las especificaciones de Java EE, se despliegan en ejecución donde son manejados por el servidor.

En caso de que la aplicación Java EE haga uso de un navegador o una aplicación cliente pesada se debe tener cuidado en la decisión de cuál de los dos utilizar para equilibrar la funcionalidad en el cliente y acercarse al usuario para cargarle al servidor toda la funcionalidad que sea posible, esto facilita la distribución, despliegue y manejo de la aplicación.

#### **2.3.3.6 Contenedores JAVA EE <sup>15</sup>**

La arquitectura basada en componentes hace que las aplicaciones Java EE sean fáciles de escribir ya que la lógica de negocio está organizada en componentes reutilizables. Adicionalmente el servidor Java EE brinda servicios de capas bajas en forma de contenedores para cada tipo de componente.

---

<sup>15</sup> Disponible en internet: <http://java.cabezudo.net/trabajos/JEE5/manual/jee5.v0.01.00/ababac.html#aaf>

## 2.3.4 TECNOLOGÍA SERVLET JAVA

### 2.3.4.1 Servlets

Un servlet es una clase del lenguaje de programación Java que es utilizada para extender las habilidades de los servidores que guardan aplicaciones a las cuales se accede mediante el modelo petición-respuesta.

A pesar de que los servlets pueden devolver a cualquier tipo de respuesta, estos son comúnmente utilizados para extender las aplicaciones almacenadas en servidores web. Para estas aplicaciones, la tecnología Servlet Java define las clases servlets específicas para HTTP.

El ciclo de vida de un servlet está controlado por el contenedor en donde el servlet ha sido desplegado. Cuando una petición es mapeada a un servlet, el contenedor realiza los siguientes pasos:

1. Si una instancia del servlet no existe, el contenedor web:
  - Carga la clase servlet.
  - Crea una instancia de la clase servlet.
2. Inicializa la instancia del servlet llamando al método `init`.
3. Invoca el método `service`, pasando los objetos `request` y `response`.
4. Si el contenedor necesita quitar el servlet, este finaliza el servlet llamando el método `destroy`.

Luego de que el contenedor carga e instancia la clase servlet y antes que distribuya solicitudes de los clientes, el contenedor Web inicializa el servlet. Para construir a medida este proceso y que el servlet cargue datos de configuración persistente, se inicializan recursos y se realiza cualquier otra actividad a tiempo, usted debe sobrecargar el método `init` de la interface `Servlet`. Un servlet que no puede completar su proceso de inicialización debe lanzar una excepción `UnavailableException`.

Muchas aplicaciones necesitan que una serie de solicitudes de un cliente sean asociadas con otro. Las aplicaciones basadas en Web son responsables por mantener este estado, llamado sesión, ya que HTTP no tiene estado. Para soportar las aplicaciones que necesitan mantener el estado, la tecnología Servlet de Java proporciona una API para manejo de sesiones y permite varios mecanismos para implementar sesiones.

Las sesiones son representadas por un objeto HttpSession. Se accede a la sesión llamando el método getSession de un objeto request. Este método retorna la sesión actual asociada con esa solicitud, aunque si dicha solicitud no tiene una sesión entonces esta última es creada.

Cuando un contenedor de servlet determina que un servlet debe ser eliminado del servicio, el contenedor llama al método destroy de la interface Servlet. En este método, usted libera todos los recursos que el servlet está utilizando y guarda cualquier estado persistente.

Todos los métodos de un servicio dado por un servlet deben ser completados cuando un servlet es eliminado. El servidor trata de asegurarse de esto llamando el método destroy solo luego de que todas las solicitudes de servicio han retornado o después de un período de gracia específico del servidor, lo que suceda primero. Si su servlet tiene operaciones que toman un tiempo largo en ejecutar, las operaciones pueden seguir ejecutándose luego de que el método destroy es llamado. Es preciso asegurar que todos los hilos manejan las solicitudes de clientes de forma completa realizando los siguientes procesos:

- Mantener la cuenta de cuantos hilos siguen ejecutando el método service.
- Proveer un apagado limpio con el método destroy notificando a los hilos que tardan mucho tiempo en ejecutarse de la baja y esperar a que estos completen.

- Hacer un sondeo periódico de los métodos que tardan mucho en ejecutar y si es necesario, detener el trabajo, hacer limpieza y retornar.

### 2.3.5 JAVASERVER FACES <sup>16</sup>

La tecnología de JavaServer Faces es un marco de trabajo de interfaz de usuario para crear aplicaciones web. Los componentes principales de la tecnología de JavaServer Faces son los siguientes:

- Un marco de trabajo para componentes GUI.
- Un modelo flexible para dibujar componentes en diferentes tipos de HTML, lenguajes de marcas y tecnologías. Un objeto `Renderer` genera las marcas para dibujar el componente y convierte los datos almacenados en un objeto modelo en tipos que pueden ser representados en una vista.
- Un `RenderKit` estándar para generar marcas HTML/4.01.

Las siguientes características soportan los componentes GUI:

- Validación de entrada
- Manejo de eventos
- Conversión de datos entre objetos de modelo y componentes
- Creación administrada de objetos modelo.
- Configuración de navegación en la página.

Toda esta funcionalidad está disponible utilizando la APIs estándar y los ficheros de configuración basados en XML.

---

<sup>16</sup> Disponible en internet: <http://java.cabezudo.net/trabajos/JEE5/manual/jee5.v0.01.00/ababai.html#abl>

### **2.3.5.1 Beneficios de la Tecnología JavaServer Faces**

JSF es una especificación estándar lo que pueden encontrarse implementaciones de distintos fabricantes. Esto permite no vincularse exclusivamente con un proveedor concreto, y poder seleccionar el más adecuado según los requerimientos de la aplicación; según el número de componentes que suministra, el rendimiento de éstos, soporte proporcionado, precio, política de evolución, etc.

JSF trata la vista (la interfaz de usuario) de una forma algo diferente a lo que se está acostumbrado en las aplicaciones web, donde la programación de la interfaz se desarrolla a través de componentes y está basada en eventos (pulsación de un botón, cambio en el valor de un campo, etc.).

JSF es muy flexible ya que permite personalizar tanto los componentes como la recarga de la vista de las páginas, con el fin elaborar interfaces de usuario en la forma que más nos convenga.

La tecnología JavaServer Faces permite construir aplicaciones web que introducen realmente una separación entre el comportamiento y la presentación, separación sólo ofrecida tradicionalmente por arquitecturas UI del lado del cliente y parcialmente por la tecnología JSP.

Separar la lógica de negocio de la presentación también permite que cada miembro del equipo de desarrollo de la aplicación web se centre en su parte asignada del proceso diseño, y proporciona un modelo sencillo de programación para enlazar todas las piezas.

Otro objetivo importante de la tecnología JavaServer Faces es mejorar los conceptos asociados con componente-UI y capa-web sin limitarse a una

tecnología de *script* particular o un lenguaje de marcas. Aunque la tecnología JavaServer Faces incluye una librería de etiquetas JSP personalizadas para representar componentes en una página JSP, las APIs de JavaServerFaces se han creado directamente sobre el API *JavaServlet*. Esto permite, teóricamente, hacer algunas cosas avanzadas: usar otra tecnología de presentación junto a JSP, crear componentes propios directamente desde las clases de componentes, y generar salida para diferentes dispositivos cliente; entre otras.

### **2.3.6 SEAM**

Seam es un Framework de aplicaciones de Java Enterprise que tiene los siguientes principios:

- Un tipo de cosas:

Seam define un modelo de componentes uniformes para la lógica del negocio en su aplicación. Un componente Seam puede ser con estado, que se encuentra asociado a cualquiera de los diversos contextos bien definidos, incluyendo el de larga duración, el contexto de persistencia, de procesos de negocio y el contexto de la conversación, que se conserva en todas las solicitudes múltiples en una interacción con el usuario.

En Seam no hay distinción entre los componentes del nivel de presentación y componentes de la lógica del negocio. El desarrollador puede estratificar la aplicación de acuerdo con la arquitectura que se haya diseñado.

Los componentes Seam pueden simultáneamente tener acceso al estado asociado a la solicitud web y al estado de recursos transaccionales.

- Integrar JSF con EJB 3.0

EJB 3 es un modelo de componentes a nivel de lógica de negocio y de persistencia, del lado del servidor, mientras que JSF es un modelo de componentes de la capa de presentación.

JSF y EJB3 funcionan mejor juntos, a pesar de que Java EE5 no proporciona un estándar para integrar estos modelos de componentes. No obstante los creadores de ambos modelos (JSF y EJB3) proporcionan puntos de extensión estándar para permitir la integración con otros marcos.

Seam unifica los modelos de componentes de JSF y EJB3, dejando al desarrollador el único problema de diseñar la lógica del negocio.

- Integrar JSF con EJB 3.0

EJB 3 es un modelo de componentes a nivel de lógica de negocio y de persistencia, del lado del servidor, mientras que JSF es un modelo de componentes de la capa de presentación.

JSF y EJB3 funcionan mejor juntos, a pesar de que Java EE5 no proporciona un estándar para integrar estos modelos de componentes. No obstante los creadores de ambos modelos (JSF y EJB3) proporcionan puntos de extensión estándar para permitir la integración con otros marcos.

Seam unifica los modelos de componentes de JSF y EJB3, dejando al desarrollador el único problema de diseñar la lógica del negocio.

- La prueba de integración es fácil

Los componentes de Seam, siendo simples clases Java, son por naturaleza comprobables. Sin embargo, para aplicaciones complejas, las pruebas unitarias por sí solas son insuficientes. SEAM proporciona la capacidad de prueba de aplicaciones Seam como una característica central del framework. El usuario puede escribir las pruebas JUnit o TestNG que reproducen una interacción completa con un usuario. Estas pruebas pueden ser ejecutadas directamente en el IDE, donde Seam automáticamente implementa los componentes EJB con JBoss Embebido.

- Las especificaciones técnicas no son perfectas

Existen limitaciones en el ciclo de vida de JSF para las solicitudes GET que fija Seam. Los autores de Seam actualmente trabajan con expertos JCP para asegurarse de que las correcciones pertinentes sean realizadas para la próxima versión.

- Hay más de una aplicación web que sirve páginas HTML

Un Framework de aplicaciones web realmente completo debe abordar problemas como la persistencia, la concurrencia, a sincronía, administración del estado, seguridad, correo electrónico, mensajería, pdf, generación de gráficos, servicios web, caché, entre otros.

Seam integra JPA e Hibernate3 para persistencia, EJB TimerService y Quartz para ligeras a sincronías, jBPM para flujo de trabajo, JBoss rules para reglas del negocio, Meldware Mail para correo electrónico, HibernateSearch y Lucene para búsqueda de texto, JMS para mensajes y JBoss Caché para la página de almacenamiento en caché.

### **2.3.7 JBOSS ENTERPRISE APPLICATION PLATAFORM**

El servidor de aplicaciones JBoss es una herramienta certificada para el desarrollo de aplicaciones empresariales Java. Su madurez y el esfuerzo de muchos desarrolladores, e incluso las sugerencias que han realizado muchos usuarios, han permitido que JBoss AS (Application Server) se popularice ampliamente y sea común en los currículos de muchos desarrolladores. Encuestas recientes muestran que es el servidor de aplicaciones más popular actualmente.

Es reconocido por soportar los estándares más recientes. De hecho, es el primer servidor de aplicaciones en alcanzar la certificación J2EE 1.4 cuando salió su versión 4.0 (actualmente va en la versión 5, liberada a finales de 2008). Pero JBoss no sólo marca la pauta en la adopción de estándares con su servidor de aplicaciones, sino en la imposición de los mismos. Recientemente se le eligió para hacer parte del Java CommunityProcess (JCP). Además en los últimos años ha estado a la cabeza del desarrollo de Java Enterprise llegando a establecerse en todas las especificaciones de requerimientos de Java (Java SpecificationRequests, JSRs).

#### **2.3.7.1 Características y beneficios**

- Eclipse completo basado en Entorno de desarrollo integrado (JBossDeveloper Studio)
- Apoya una amplia gama de Java EE y estándares para servicios web.
- JBoss paquetes de EAP e hibernate, mapeo Objeto/Relación y persistencia.
- JBoss Seam Web Application Framework.
- Almacenamiento en caché, clustering y alta disponibilidad.
- Servicios de seguridad.

- Servicios Web e interoperabilidad.
- Integración de servicios de mensajería.<sup>17</sup>

### 2.3.8 ENTERPRISE ARCHITECT

Es una de las herramientas más completas dedicadas al análisis y diseño orientado a objetos, para el modelado de proyectos de software. Combina la capacidad de la versión 2.1 de UML (Lenguaje Unificado de Modelado) con una interfaz intuitiva, lo que hace que sea eficaz tanto para trabajo en un ordenador de sobremesa como para ser usada dentro de un equipo de desarrollo.

Algunas de las características que presenta esta herramienta son:

- Puede representar todo el ciclo de vida de un sistema: desde el análisis de los requisitos hasta el mantenimiento del mismo pasando por el diseño de los modelos, las pruebas del sistema y las etapas de análisis.
- La utilización de UML hace que de una manera fácil se puedan crear modelos de sistema consistentes y trazables.
- Como la herramienta se apoya en el lenguaje UML, se pueden generar a través de ella los 13 tipos de diagramas soportados por dicho lenguaje.
- Se pueden generar informes completos de una manera sencilla, ya que permite la creación de documentación del sistema en varios formatos.

---

<sup>17</sup> Tomado y traducido de JBoss Enterprise ApplicationPlatform. Disponible en internet: <http://www.jboss.com/products/platforms/application>

### 2.3.9 EJB 3.0

Enterprise JavaBeans (EJB) es una arquitectura de componentes para la construcción de aplicaciones empresariales ejecutadas en servidores. Tiene por propósito proveer una forma estándar de implementar este tipo de aplicaciones, haciéndose cargo de aspectos comunes y repetitivos como la persistencia, la integridad transaccional y la seguridad, permitiendo que el desarrollador pase a preocuparse exclusivamente por la lógica del negocio en sí.

JBoss AS fue de los primeros servidores de aplicaciones en adoptar las especificaciones de EJB 3.0. Este modelo de EJB simplifica el desarrollo eliminando la necesidad de una interfaz “Home” y descriptores de despliegue (reemplazándolos por anotaciones), y facilita la implementación de la persistencia de una nueva manera por medio de JPA.

### 2.3.10 JPA

Más conocida por su sigla **JPA**, es la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí pasaba con EJB2, y permitir usar objetos regulares.

Una característica fascinante de JPA, es que permite que se hagan cambios al diseño de la base de datos sin tener que reescribir enteramente las aplicaciones. Para esto JPA introduce JPQL.

### **2.3.11 JPQL**

Java Persistence Query Language es el lenguaje de consultas que se usará en el desarrollo de este trabajo. Con él, el desarrollador jamás se refiere a las entidades directamente al momento de hacer las consultas, sino a los objetos mismos que fueron mapeados. Si el diseño de la BD cambiara, sólo habría que modificar las anotaciones de las entidades. El resto del código quedaría intacto y seguiría funcionando tal y como antes. Esta facilidad permite optimizar las bases de datos cuando se considere necesario, migrar a bases diferentes, o sencillamente corregir diseños defectuosos con un esfuerzo mínimo.

### **2.3.12 CONSTRUCCIÓN DE PROTOTIPOS**

#### **2.3.12.1 ¿Qué es un prototipo?**

Un prototipo es un ejemplar o una representación limitada del diseño de un producto que permite a las partes responsables de su creación y a los interesados experimentar su diseño y utilidad al probarlo en situaciones reales. Pertenece a los modelos de desarrollo evolutivo y es frecuente que los usuarios interesados en el desarrollo del sistema, definan un conjunto de objetivos globales que debe realizar el software, pero no identifican los requerimientos detallados de lo que se espera que realice el sistema o estos no tienen claros los requisitos del nuevo software

Un prototipo de software apoya a dos actividades del proceso de ingeniería de requerimientos:

- Obtención de requerimientos: les permite adquirir nuevas ideas para los requerimientos y encontrar áreas fuertes y débiles del software.

- Validación de requerimientos. El prototipo puede revelar errores y omisiones en los requerimientos. La construcción de prototipos puede utilizarse como un análisis de riesgo y una técnica de reducción. Un modelo iterativo del proceso, como el desarrollo incremental, se utiliza junto con un lenguaje diseñado para el desarrollo rápido de aplicaciones. Por lo tanto, las técnicas utilizadas para desarrollar un prototipo para validar los requerimientos también se utilizan para desarrollar el sistema de software mismo.

#### Ventajas:

- Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- Al demostrar las funciones del sistema se identifican las discrepancias entre los desarrolladores del software y los usuarios.
- Durante el desarrollo del prototipo el personal del desarrollo de software puede darse cuenta de que los requerimientos son inconsistentes y/o están incompletos.
- Se dispone rápidamente de un sistema que funciona y demuestra la factibilidad y usabilidad de la aplicación a administrar.

En sistemas grandes y complejos una forma de resolver la dificultad de evaluación es utilizar un enfoque evolutivo para el desarrollo de sistemas. Esto significa proporcionar al usuario un sistema incompleto y después modificarlo y aumentarlo en el momento en que los requerimientos del usuario sean claros.

El enfoque de construcción de prototipos desechables es para ayudar a refinar y clasificar la especificación del sistema. El prototipo se escribe, evalúa y modifica. La evaluación del prototipo informa del desarrollo de la especificación detallada del sistema que se incluye en el documento de requerimientos de este. Una vez que se ha redactado la especificación, el prototipo ya no es útil y se desecha.

### **2.3.12.2 Prototipo No funcional <sup>18</sup>**

Es un modelo no funcional a escala configurado para probar ciertos aspectos de diseño. Un ejemplo de este enfoque es un modelo a escala completa de un automóvil que se usa para pruebas en un túnel de viento. El tamaño y forma del automóvil son precisos, pero el automóvil no es funcional. En este caso solo se incluyen las características del automóvil que son fundamentales para la prueba en el túnel de viento.

Un modelo no funcional a escala de un sistema de información podría producirse cuando la codificación requerida por las aplicaciones es demasiado extensa para incluirse en el prototipo, pero se puede conseguir una idea útil del sistema a través de la elaboración de un prototipo de la entrada y la salida. En este caso, el procesamiento, debido al excesivo costo y el tiempo requerido, no podría incluirse en el prototipo. Sin embargo, aún se podrían tomar algunas decisiones sobre la utilidad del sistema con base en la entrada y la salida incluidas en el prototipo.

### **2.3.13 Axure RP Pro 5.6**

Axure RP es una herramienta completa para la creación rápida de WireFrames, prototipos y especificaciones para aplicaciones y sitios Web. La presentación, pantallas interactivas, ayudan a comunicar el diseño claramente, y colabora en la retroalimentación con los clientes, identificando claramente los requisitos que por una u otra manera se pasaron por alto en las primeras entrevistas.

Con Axure RP se pudo crear un prototipo interactivo basado en el navegador, sin necesitar programación de código.

---

<sup>18</sup> KENDALL, Julie E. Análisis y Diseño de sistemas. Capítulo 6. Página 153

Axure muestra de manera gráfica el esquema de navegación del Sistema, brindando widgets y utilidades con las que el público en general está familiarizado, además de aportar librerías de elementos creados para rapidez en la elaboración del prototipo y facilitar la introducción del cliente al sistema.

Las características principales de Axure RP son: <sup>19</sup>

- Generar prototipos interactivos HTML con el clic de un botón, sin una sola línea de código.
- Muestra una amplia gama de interacciones en su diseño como base de unión, expansión y contracción de contenido.
- Comparte sus prototipos HTML con otros compañeros y clientes para ser observados en los navegadores más populares sin necesidad de instalar para observar el prototipo.

Es una herramienta distribuida por Axure Software Solutions, Inc, empresa privada dedicada al desarrollo de software, la versión piloto de Axure RP que es su producto líder, fue en enero de 2003.

---

<sup>19</sup> Disponible en internet: <http://www.axure.com/>

### **3 METODOLOGÍA DE DESARROLLO**

#### **3.1 CICLO DE VIDA DEL PROYECTO**

A continuación una descripción de las diferentes actividades que se llevaron a cabo durante el transcurso del proyecto, buscando conceptualizar la metodología que se aplicó en el desarrollo del nuevo sistema de información de la red de datos de la Universidad Industrial de Santander (SIMANRED).

##### **3.1.1 Análisis de Requerimientos.**

El análisis de requerimientos es el proceso en el cual se establecen los requisitos o servicios que el cliente requiere de un sistema y los límites bajo los cuales va a operar, dentro de este proceso es necesario establecer los requerimientos funcionales como también los no funcionales.

Los requerimientos funcionales definen las funciones o tareas que el sistema será capaz de realizar.

Los requerimientos no funcionales son las características que pueden limitar el sistema, ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad, mantenimiento, seguridad, estándares, etc.

La especificación de requerimientos suministra los medios para valorar la calidad de los programas, una vez que se haya construido.

Todo el proceso de análisis de requerimientos se especificó, junto con los funcionarios que trabajan con SIMANRED, las funcionalidades y comportamiento que debía tener los programas desarrollados en el transcurso del proyecto, se

definió la interfaz, el esquema de navegación entre páginas junto con otros elementos y se establecieron los estándares de diseño que el sistema debe mantener.

En esta etapa se hicieron reuniones periódicas con los usuarios de SIMANRED, para que fuesen ellos los que definan las características del software que se desea implantar, para que este se ajustara plenamente a las exigencias de la infraestructura de la red de datos institucional actualmente.

### **3.1.2 Diseño**

En la etapa de diseño se traducen los requerimientos funcionales y no funcionales en una representación del software, donde se pueda evaluar su calidad antes de que comience la codificación.<sup>20</sup>

El diseño se realizó, mediante modelos UML (Lenguaje de Modelado Unificado) y se incluyeron los diagramas que han sido seleccionados dentro de los estándares de desarrollo de software utilizados en la División de Servicios de Información, que son:

- Diagrama de casos de uso
- Diagrama de Clases
- Diagrama de Entidad- Relación
- Diagrama de Secuencia

Se utilizó la herramienta Enterprise Architect licenciada por la Universidad Industrial de Santander a través de División de Servicios de Información.

---

<sup>20</sup> Ingeniería del Software, Análisis y Diseño. Disponible en internet:  
[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/fuentes\\_k\\_jf/capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo2.pdf)

### **3.1.3 Implementación de la Aplicación**

En esta etapa se procede a generar el software que se ha diseñado. Se realizó teniendo en cuenta los parámetros establecidos por la División de Servicios de Información en cuanto a los estándares técnicos y de calidad que caracterizan las aplicaciones que son generadas para el servicio de la Universidad, teniendo como base el Lenguaje de programación JAVA 5, frameworks como: seam, java server faces (JSF), Enterprise Java Beans (EJB 3.0) e Informix como motor de base de datos.

### **3.1.4 Pruebas del software**

La fase de pruebas es una de las más costosas del ciclo de vida de software, permiten verificar y revelar la calidad de un producto software. Se realizan ya que las fallas de software ocasionan graves pérdidas económicas; estas son cien a mil veces más costosas de encontrar y de reparar después de la construcción.

Las pruebas se centran en los procesos lógicos internos del software, asegurando que todas las sentencias sean probadas y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos.

Estas pruebas se aplicaron de forma permanente a lo largo del desarrollo del proyecto por parte del equipo de trabajo, y se abrió un espacio dónde los usuarios finales interactuaron con la aplicación con el objetivo que detectaran posibles fallos que hayan sido omitidos en el momento del desarrollo.

### **3.1.5 Ajustes**

Es el proceso de mejora del sistema tomando como base las sugerencias y observaciones que se plantearon en el periodo de pruebas, aquí también se

incluyen las correcciones a los posibles fallos detectados, mejoras en la interfaz y el diseño de la misma.

Este proceso está incluido dentro del proceso iterativo de refinamiento que se le dio al sistema, para adaptarse plenamente a las necesidades del cliente que en el presente caso es la División de Servicios de Información.

## **3.2 METODOLOGIA DE DESARROLLO DEL PROYECTO**

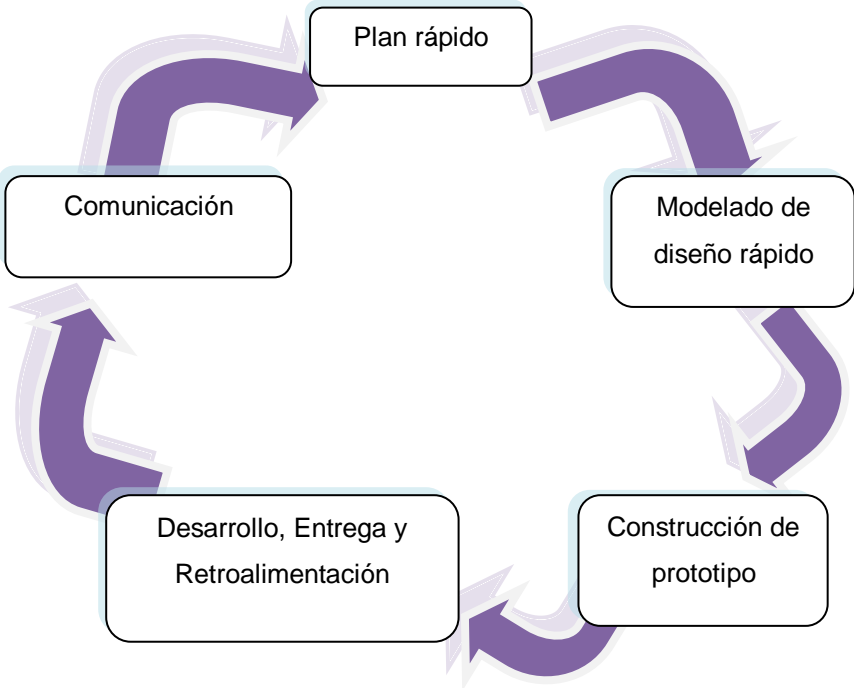
### **3.2.1 Modelo de construcción por prototipos**

Teniendo presente las actividades descritas anteriormente, se propone como metodología de desarrollo de éste proyecto el MODELO DE CONSTRUCCION POR PROTOTIPOS.

Se planteó este modelo porque pertenece a los modelos de desarrollo evolutivo, y es frecuente que los usuarios interesados en el desarrollo del sistema, definan un conjunto de objetivos globales que debe realizar el software, pero no identifican los requerimientos detallados de lo que se espera que realice el sistema o estos no tienen claros los requisitos del nuevo software. En otros casos, el responsable del desarrollo del software puede no estar seguro de la eficacia de un algoritmo, o no haber comprendido plenamente el requerimiento del usuario. Por estas razones el modelo de construcción de prototipos es la mejor opción para el desarrollo del proyecto ya que los funcionarios de la red de datos institucional tienen una visión general de los nuevos requisitos del sistema. Este modelo, permitirá la corrección temprana de errores o la redefinición del sistema en caso de ser necesario, y los prototipos funcionales permitirán la familiarización del usuario con el sistema que se está desarrollando.

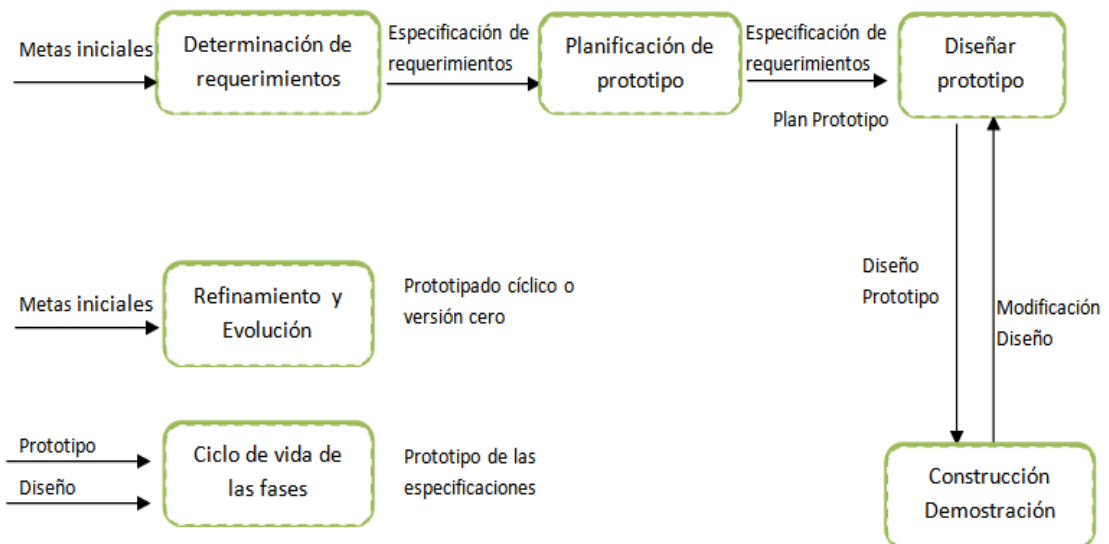
A continuación se observa la estructura del modelo:

Figura 11. Modelo de construcción de prototipos



**3.2.2 Estructura del Modelo de Construcción de Prototipos**

Figura 12. Estructura del modelo



Esta metodología resultó viable para el desarrollo del proyecto y se escogió debido a las siguientes razones:

- En la creación de los prototipos iniciales se debe trabajar con unas ideas aproximadas de lo que desea la División de Servicios de Información, para presentar un producto o prototipo inicial, el cual puede evolucionar y refinarse dando como resultado un prototipo más maduro, que cumpla con todos los requerimientos de los usuarios.
- Con el uso del modelo de prototipos se da la facilidad de mejorar, de manera temprana los prototipos, teniendo en cuenta las sugerencias del usuario solicitante del proyecto, de tal forma que se cubran a cabalidad sus requerimientos.

- En este modelo de desarrollo se debe validar la versión actual del prototipo, para después generar una nueva versión que complemente los nuevos requerimientos, con el fin de evitar retrocesos en el proceso de desarrollo.

### **3.2.3 Procedimiento a seguir para la metodología planteada**

- a. El desarrollo de los prototipos comienza con la especificación de los requisitos, con los usuarios administradores del sistema actual y apoyado en la División de Sistemas de Información de la universidad (DSI).
- b. El equipo de desarrollo y el usuario se reúnen para que se definan los objetivos globales del software, se identifican todos los requisitos conocidos, se realiza el análisis de requerimientos y se redacta la especificación de requisitos iniciales.
- c. Se produce el diseño del prototipo que se enfoca en la representación de los aspectos del software visibles al usuario (por ejemplo, métodos de entrada, esquemas de navegación y formatos de salida). En ésta etapa el usuario adquiere mayor claridad de las funcionalidades del sistema y da las sugerencias necesarias para la construcción de un prototipo más avanzado.
- d. Se procede con la construcción del prototipo que se ha propuesto.
- e. El prototipo es valorado por el usuario y posteriormente se refinarán los aspectos necesarios.
- f. Se produce un proceso iterativo en el que el prototipo es refinado, mejorado para que satisfaga las necesidades del usuario, al mismo tiempo que facilita al desarrollador una mejor comprensión de lo que tiene

que hacer el sistema y así entregar el producto final requerido con una excelente calidad.

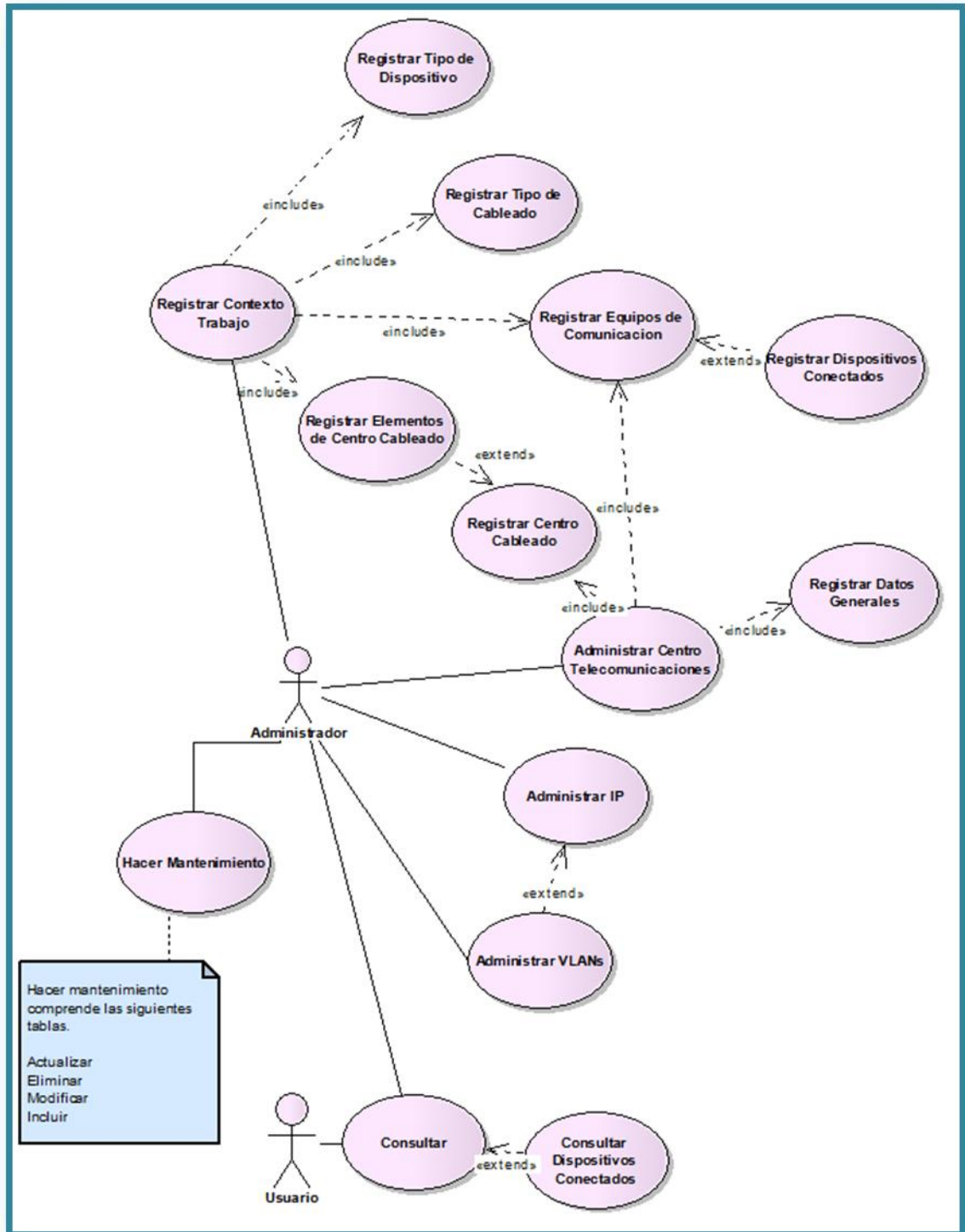
### **3.3 APLICACIÓN DE LA METODOLOGÍA**

#### **3.3.1 DIAGRAMAS UML**

Dentro del desarrollo del proyecto se generaron los diagramas UML que se presentarán a continuación, es de resaltar que durante todo el desarrollo del proyecto se fue ajustando los prototipos de diseño hasta conseguir un prototipo final. Se incluyeron los diagramas siguiendo con los estándares de desarrollo de software utilizados en la División de Servicios de Información, los cuales son: de casos de uso, de clases y de secuencia.

##### **3.3.1.1 Diagrama de Casos de Uso**

Figura 13. Casos de Uso SIMANRED



Por motivos de confidencialidad en la información, del diagrama de casos de uso presentado anteriormente se seleccionaron sólo algunos de ellos para explicarlos en detalle. Entre los escogidos están: Registrar Dispositivo, Administrar Vlans y Registrar Equipo de Comunicación.

### Caso de Uso: **Registrar Dispositivo**

Tabla 1. Detalle del caso de uso - Registrar Dispositivo

<b>REGISTRAR DISPOSITIVOS</b>	
<b>Descripción:</b>	Registrar los dispositivos de red tales como PCs, Servidores, Teléfonos, Impresoras, entre otros
<b>Resumen</b>	El usuario ingresa por el menú correspondiente e ingresa los datos para poder registrar los tipos de dispositivos, y luego tiene la opción de salir.
<b>Actor(es):</b>	Administrador
<b>Casos de Uso Asociados:</b>	Registrar Dispositivo
<b>Precondición:</b>	Tener abierto el modulo de registrar dispositivos. El usuario se loguea como administrador con una clave.
<b>Flujo de eventos del caso de uso:</b>	
PASO	<b>DESCRIPCIÓN</b>
Flujo de éxito	
	<p>El usuario selecciona "Registrar Dispositivos" e ingresa o selecciona los siguientes datos:</p> <ul style="list-style-type: none"> <li>• <b>Código de Dispositivo:</b> Identificación de cada Tipo de dispositivo.</li> <li>• <b>Código Tipo de Dispositivo:</b> identifica el tipo específico del dispositivo, ejemplo 1. Pc, 2. teléfono, 3. impresora, entre otros.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Descripción:</b> Una breve descripción de las características del dispositivo.</li> <li>• <b>Fecha de Instalación:</b> Se refiere a la fecha de creación del registro del dispositivo.</li> <li>• <b>Fecha final:</b> Se refiere a la fecha en que es dado de baja el dispositivo.</li> <li>• <b>Indicativo de dispositivo:</b> Si es Dispositivo principal o derivado.</li> <li>• <b>Mac:</b> Si el dispositivo tiene Mac, la dirección física del dispositivo.</li> <li>• <b>Nombre:</b> El nombre del equipo.</li> <li>• <b>Numero de Inventario:</b> El numero de inventario asignado por planta física para cada dispositivo.</li> <li>• <b>Soporte IPV6:</b> El usuario selecciona si el equipo soporta o no IPV6</li> </ul> <p>El sistema valida los datos ingresados. El sistema almacena los datos y enseña un mensaje de confirmación de Registro Exitoso del sistema o continúa.</p>
<b>Flujo alternativo (n)</b>	
	El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija.
<b>Pos_condición:</b>	El sistema se ha actualizado, o su transacción ha sido exitosa.

### Caso de Uso: Registrar Equipo de Comunicación

Tabla 2. Detalle del caso de uso - Registrar Equipo de Comunicación

<b>REGISTRAR EQUIPO DE COMUNICACIÓN</b>	
<b>Descripción:</b>	Registrar los equipos de comunicación tales como switches,

	Routers, entre otros.
<b>Resumen</b>	El usuario ingresa por el menú correspondiente e ingresa los datos para poder registrar los equipos de comunicación, y luego tiene la opción de salir.
<b>Actor(es):</b>	Administrador
<b>Casos de Uso Asociados:</b>	Registrar Equipo Comunicación
<b>Precondición:</b>	Tener abierto el módulo de registrar equipos de comunicación. El usuario se loguea como administrador, con una clave.
<b>Flujo de eventos del caso de uso:</b>	
<b>PASO</b>	<b>DESCRIPCIÓN</b>
Flujo de éxito	
	<p>1) El usuario selecciona "Registrar equipos de comunicación" e ingresa o selecciona los siguientes datos:</p> <ul style="list-style-type: none"> <li>• <b>Código de Equipo de comunicación:</b> Se refiere a un código característico de cada equipo de comunicación para su identificación.</li> <li>• <b>Código Centro de Telecomunicaciones:</b> identifica el centro de cableado o de telecomunicaciones en el que se encuentra el equipo de comunicación (switch, router, otro)</li> <li>• <b>Dimensión:</b> El tamaño del centro de telecomunicaciones.</li> <li>• <b>Fecha Inicio de Vigencia:</b> Se refiere cuando empieza en funcionamiento el equipo.</li> <li>• <b>Fecha final de vigencia:</b> La fecha en que se termina la vigencia de utilización del equipo.</li> <li>• <b>Mac:</b> se refiere al identificador Mac que tiene por defecto estos dispositivos.</li> <li>• <b>Modelo:</b> Se refiere al modelo del equipo por ejemplo si es un router, marca cisco, se refiere al modelo si es un Inspiron 870 o un 850, etc.</li> <li>• <b>Nombre:</b> El nombre del equipo.</li> <li>• <b>Numero de Inventario:</b> El número de inventario asignado por planta física para cada equipo.</li> <li>• <b>Serial:</b> El serial propio del equipo.</li> <li>• <b>Soporte IPV6:</b> El usuario selecciona si el equipo soporta o no IPV6</li> <li>• <b>Tipo Cableado:</b> Tipo de Cableado hace referencia a que tipo de cable se está utilizando para la conexión de ese equipo con los</li> </ul>

	<p>demás .El tipo de cable puede ser cable de pares a coaxial y fibra.</p> <ul style="list-style-type: none"> <li>• <b>Tipo de conectividad:</b> Stacking o Apilamiento.</li> </ul> <p>El sistema valida los datos ingresados. El sistema almacena los datos y enseña un mensaje de confirmación de Registro Exitoso del sistema.</p>
<b>Flujo alternativo (n)</b>	
	El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija.
<b>Pos_condición:</b>	El sistema se ha actualizado, o su transacción ha sido exitosa.

Caso de Uso: **Administrar Vlan**

Tabla 3. Detalle del caso de uso – Administrar Vlan

<b>ADMINISTRAR VLAN</b>	
<b>Descripción:</b>	Permite la administración de las VLans.
<b>Resumen</b>	El usuario ingresa por el menú correspondiente e ingresa los datos para poder administrar, registrar las VLans, y luego tiene la opción de salir.
<b>Actor(es):</b>	Administrador
<b>Casos de Uso Asociados:</b>	Administrar Vlan
<b>Precondición:</b>	Tener abierto el modulo para la gestión de las VLans. El usuario se loguea como administrador, con una clave.
<b>Flujo de eventos del caso de uso:</b>	
PASO	<b>DESCRIPCIÓN</b>
Flujo de éxito	

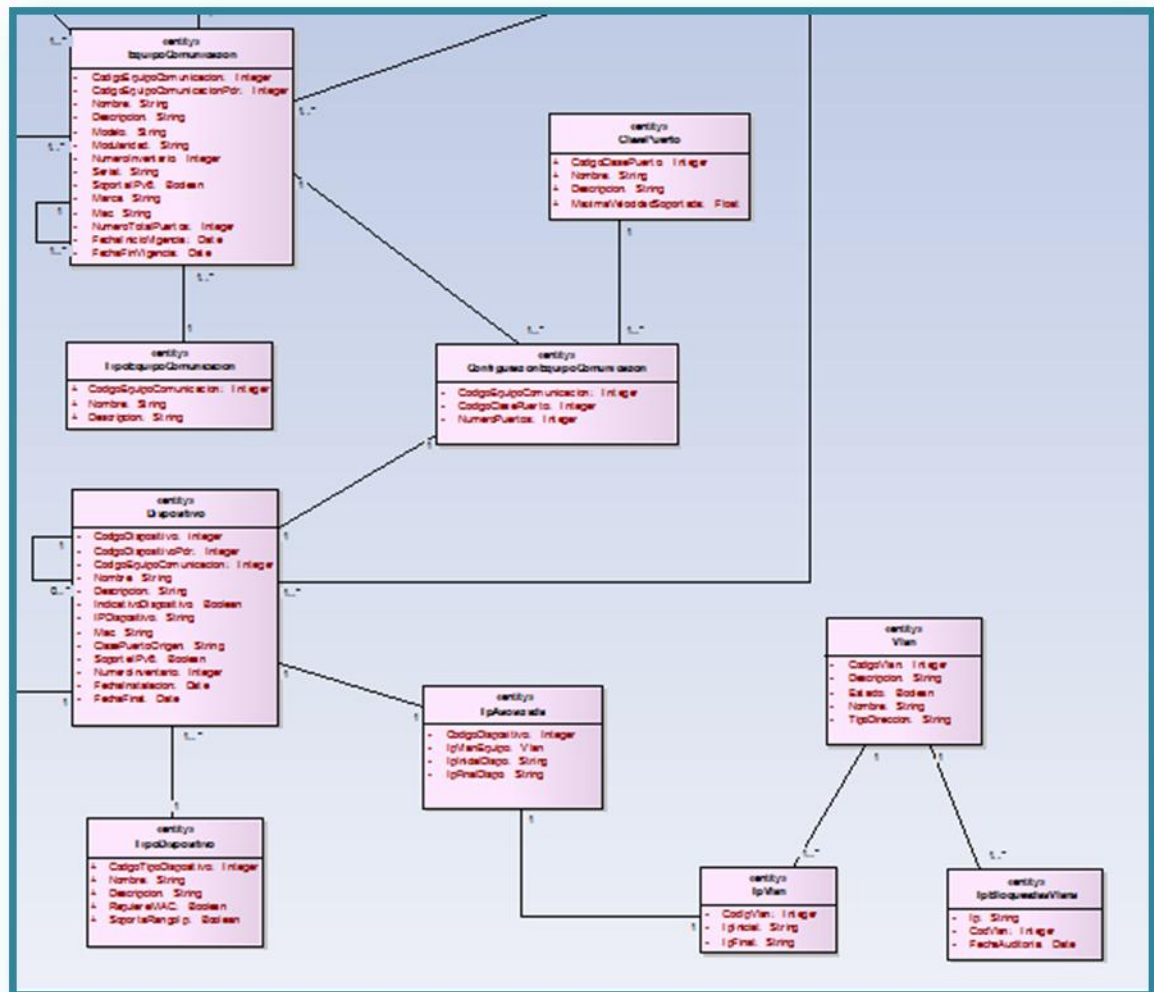
	<p>2) El usuario selecciona "Administrar VLans", Ingresa o selecciona los siguientes datos:</p> <ul style="list-style-type: none"> <li>• <b>Codigo Vlan:</b> Se refiere al identificador de la Vlan.</li> <li>• <b>Descripción:</b> Una breve descripción, para que es utilizada la Vlan, por ejemplo: Vlan Grupos de investigación.</li> <li>• <b>Estado:</b> Se refiere a que si la Vlan esta activa o Inactiva.</li> <li>• <b>Nombre:</b> Este campo se refiere al nombre que se le dará a la Vlan por ejemplo: 37</li> <li>• <b>Tipo Dirección:</b> Este campo se refiere a que si la Vlan es de tipo IpV4 o Ipv6</li> </ul> <p>El sistema valida los datos ingresados. El sistema almacena los datos y enseña un mensaje de confirmación de Registro Exitoso del sistema.</p>
<b>Flujo alternativo (n)</b>	
	El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija.
<b>Pos_condición:</b>	El sistema se ha actualizado, o su transacción ha sido exitosa.

### 3.3.2.1 Diagramas de Clases

El diagrama de clases presentado se ha modificado de acuerdo a las necesidades del cliente durante todo el proyecto, se incorporaron nuevas clases con atributos que lo hicieron aún más consistente.

Por motivos de confidencialidad en la información, sólo se presenta una parte del diagrama de clases desarrollado, se explicara muy brevemente estas clases: Dispositivo, Tipo Dispositivo, Clase Puerto, Equipo Comunicación, Configuración Equipo, Clase Puerto, IP Asociada, IPVlan.

Figura 14. Diagrama de Clases SIMANRED (Fragmento)



### Clase “Dispositivo”

La clase Dispositivo representa los diferentes dispositivos que se pueden asociar a cada uno de los equipos de comunicación o que son utilizados en la infraestructura de la red de datos, tales como PCs, Teléfonos, Impresoras, etc.

Tabla 4. Atributos clase Dispositivo

ATRIBUTOS		
Nombre	Tipo de datos	Descripción
codigoDispositivo	Entero	Identificador del Dispositivo
codigoTipoDispositivo	Entero	Identificador del tipo de dispositivo, se refiere al identificador del tipo de dispositivo que puede ser el dispositivo que se va a registrar.
Nombre	Cadena (String)	Nombre natural del dispositivo.
Descripción	Cadena (String)	Miembro que contiene un texto informativo el dispositivo.
IndicativoDispositivo	Booleano	Miembro que contiene información si el dispositivo es principal(Padre) o si es un derivado de otro(hijo).
IPEquipo	Cadena (String)	Guarda la dirección IP asociada ese dispositivo.
MAC	Cadena (String)	Miembro que guarda la ubicación geográfica del edificio dentro del Campus Universitario.
numeroInventario	Entero	Guarda el número de inventario asociada a este dispositivo, El mismo número que tiene planta física de ese dispositivo.

ClasePuertoOrigen	Smalling	Indica el puerto al que está conectado el dispositivo.
SoportalPV6	Boleano	Guarda si el dispositivo está en capacidad de Soportar IpV6.
FechaInstalacion	Date	Guarda la fecha en que se realizo la instalación del dispositivo.
FechaFinal	Date	Guarda la fecha en que el dispositivo ha sido dado de baja.
Auditoría		

### Clase “TipoDispositivo”

La clase TipoDispositivo representa los diferentes Tipos de dispositivos que se pueden seleccionar cuando creó un dispositivo.

Tabla 5. Atributos clase Dispositivo

ATRIBUTOS		
Nombre	Tipo de datos	Descripción
codigoTipoDispositivo	Entero	Identificador Tipo de Dispositivos.

Nombre	Cadena (String)	Guarda el Nombre que identifica a cada Tipo de Dispositivo de la Universidad, por ejemplo (PC25, Impresora350, etc).
Descripción	Cadena (String)	Miembro que contiene un texto informativo acerca del tipo de dispositivo.
RequiereMAC	Boleano	Miembro que guarda si el dispositivo tiene MAC o si no la tiene.
SoportaRangoIP	Boleano	Guarda si el dispositivo soporta o no soporta un Rango Ip. Toma dos posibilidades: Si o No
Auditoría		

### Clase “EquipoComunicacion”

La clase Equipo Comunicación representa los diferentes equipos de comunicación que son utilizados en la infraestructura de la red de datos, tales como Routers, Switches, etc.

Tabla 6. Atributos clase Equipo Comunicación

ATRIBUTOS		
Nombre	Tipo de datos	Descripción

codigoEquipoComunicacion	Entero	Identificador del Equipo de comunicación
EquipoComunicacionPadre	Entero	Identificador del padre del equipo de comunicación, es decir el identificador del equipo al que está asociado.
Nombre	Cadena (String)	Nombre natural del equipo de comunicación.
Descripción	Cadena (String)	Miembro que contiene un texto informativo sobre el equipo de comunicación.
Modelo	Cadena (String)	Guarda el modelo del equipo de comunicación, por ejemplo si es un router , marca cisco, se refiere al modelo si es un 870 o un 850, etc.
Modularidad	Cadena (String)	Guarda el Tipo de conectividad hace referencia a si es stacking, cascada. Como están conectados estos dispositivos en el rack o gabinete.
NumeroInventario	Entero	Guarda el número de inventario asociada al equipo de comunicación, es el mismo número que tiene planta física de ese equipo.
Serial	Smalling	Guarda el serial del equipo comunicación.

SoportaIPv6	Boleano	Guarda si el equipo está en capacidad de Soportar IpV6.
FechaInicio	Date	Guarda la fecha en que se realizo la instalación del equipo de comunicación.
FechaFinal	Date	Guarda la fecha en que el equipo de comunicación ha sido dado de baja.
Auditoría		

### Clase “IpAsociada”

La clase IpAsociada representa la relación entre un dispositivo y su IP asociada.

Tabla 7. Atributos clase Ip Asociada

ATRIBUTOS		
Nombre	Tipo de datos	Descripción
CodigoDispositivo	Entero	Guarda el Identificador del dispositivo es llave foránea.
IpFinalDispo	Cadena (String)	Guarda el valor final de rango IP, soportando IpV4 o IpV6 (Si el dispositivo o Equipo de comunicación aneja rango)

IpInicialDispo	Cadena (String)	Guarda el valor de Inicio de rango IP, soportando IpV4 o IpV6 (Si el dispositivo o Equipo de comunicación aneja rango)
IpVlan	Vlan	Es el numero Ip que puede tomar el equipo que se conecta a la Vlan.
Auditoría		

### 3.3.2.2 Diagrama de Secuencias

Teniendo en cuenta los estándares de la DSI, se elaboraron los diagramas de secuencia considerados de mayor importancia, el cual da una visión más amplia del funcionamiento y la interacción del sistema con el usuario. En nuestro caso, solamente se describen dos diagramas de secuencia, debido a que se utilizó el prototipo no funcional en Axure, el cual describe la secuencia lógica de los casos de uso.

Se describen los casos de uso de Registrar Centro de Cableado y Registrar Elementos del Centro de Cableado.

Figura 15. Diagrama de Secuencia – Caso de Uso Registrar Centro de Cableado

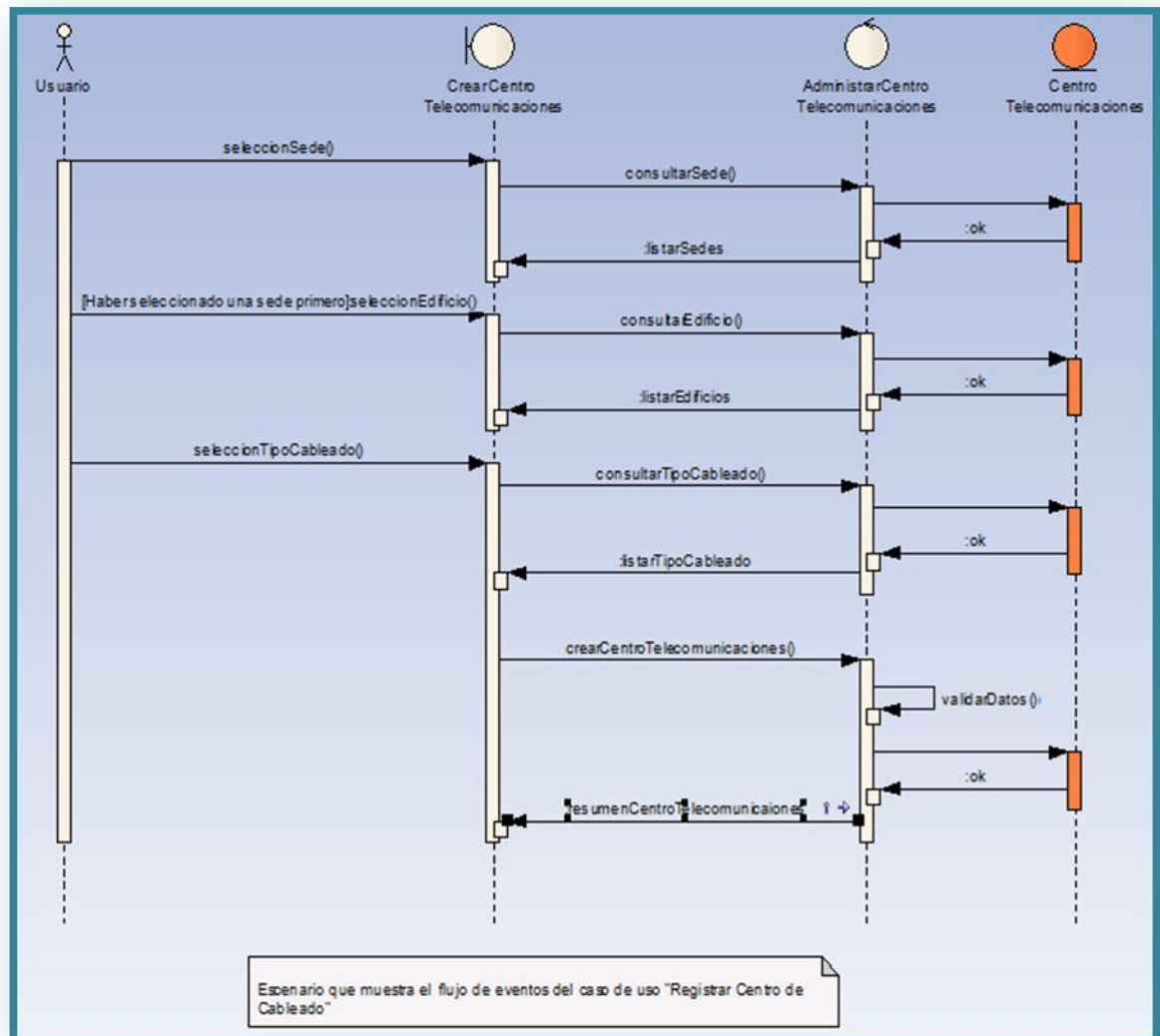
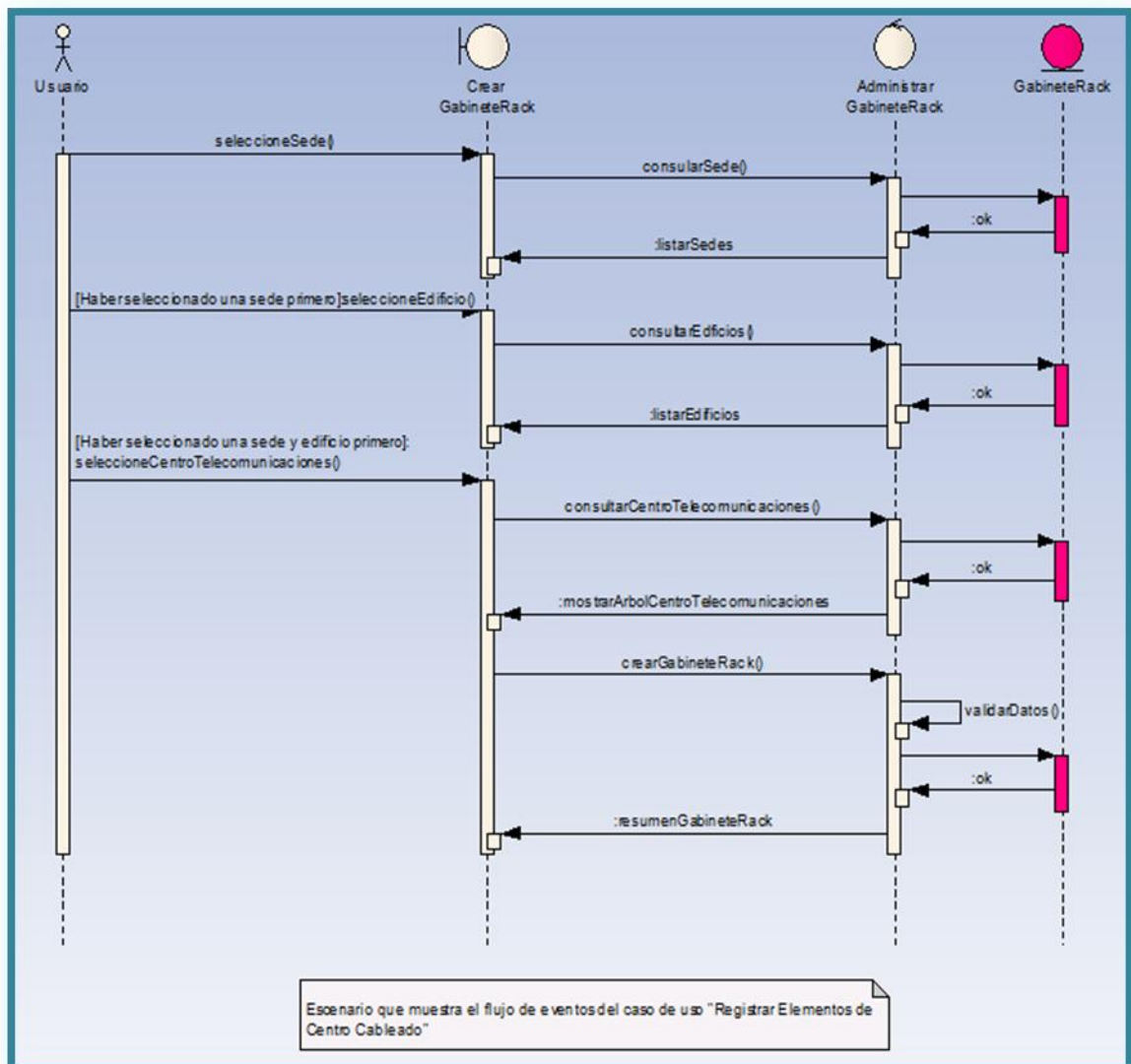


Figura 16. Diagrama de Secuencia – Caso de Uso Registrar Elementos del Centro de Cableado



## 4 PROTOTIPOS

### 4.1 PROTOTIPO NO FUNCIONAL

#### 4.1.1 Primer prototipo

Para la nueva versión del sistema de información para la red de datos de la Universidad Industrial de Santander se desarrolló un prototipo inicial con las funcionalidades básicas de creación y eliminación de registros. El prototipo consistía de una serie de formularios maestro/detalle<sup>21</sup> que manipulaban las clases de los diferentes módulos.

Este primer prototipo consistió en el desarrollo de la interfaz de usuario de los módulos de inicialización, sede, edificio y Toma. La primer interfaz en ser desarrollada fue la de la tabla sede. Dado que las tablas de soporte eran necesarias para la creación de las demás tablas, estas fueron creadas casi al mismo tiempo que la interfaz de sede.

Se implementaron las siguientes funcionalidades básicas:

- Crear registros.
- Ver información detallada de un registro seleccionado.
- Modificar los registros creados.
- Eliminar un registro seleccionado.

A continuación se muestra la interfaz que permite la creación de sede:

---

<sup>21</sup> El esquema maestro detalle hace referencia a aplicaciones que administran un conjunto de tablas relacionadas entre sí, facilitando las operaciones de listado, inserción, borrado y actualización de datos.

En la interfaz de creación de sede se muestra los campos que son requeridos para la creación de una sede, como el Código de la Sede, Un Nombre, la vigencia de la sede, es decir, si se encuentra activa o no y un campo de descripción de la sede que se va a crear.

Figura 17. Crear sede (Primer Prototipo)

El formulario cuenta con el listado de las sedes guardadas en base de datos, la siguiente lista se muestra la aplicación maestro detalle.

Figura 18. Maestro – Detalle (Primer Prototipo)

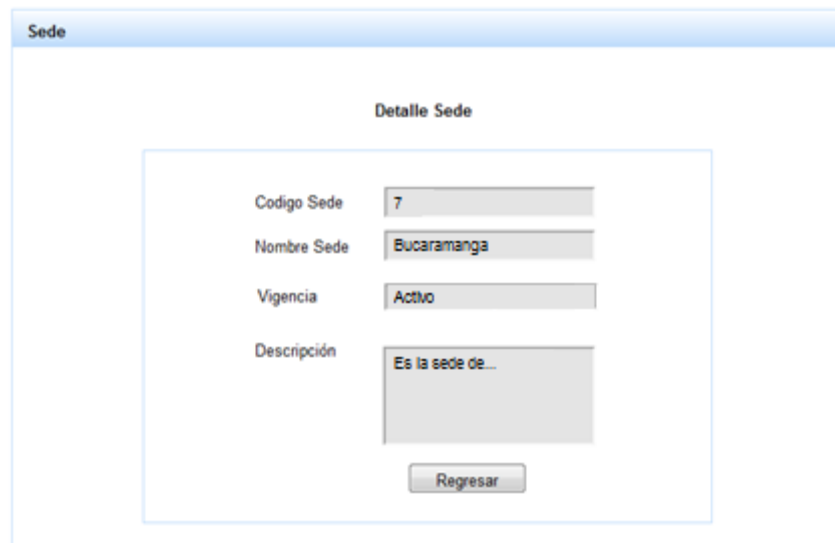
Listado de Sedes			
Nombre Sede	Vigencia	Descripcion	Acciones
Bucarmanga	Activo	Es la sede de...	
Malaga	Activo	Es la sede de malaga...	
Socorro	Activo	Es la sede de socorro ...	

La siguiente imagen pertenece al formato de las GUI (Graphical User Interface) de detalle de sede.

### Acción Detalle

La interfaz de detalle, se hizo con el fin de mostrar a usuario los datos completos de una sede específica, en esta interfaz es de solo lectura y no permite ninguna modificación en ningún campo, es simplemente informativa. Tiene un botón para regresar a la interfaz Crear Sede.

Figura 19. Formulario detalle sede (Primer Prototipo)



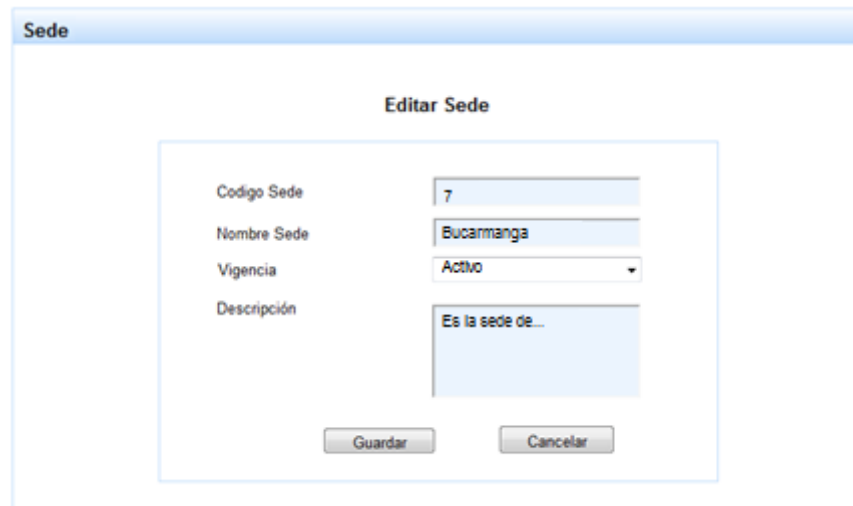
The image shows a software window titled "Sede" containing a form titled "Detalle Sede". The form is enclosed in a light blue border and contains the following elements:

- Codigo Sede:** A text input field containing the number "7".
- Nombre Sede:** A text input field containing the text "Bucaramanga".
- Vigencia:** A text input field containing the text "Activo".
- Descripción:** A text area containing the text "Es la sede de...".
- Regresar:** A button located at the bottom center of the form.

## Acción Editar

En las interfaces de edición, el formulario es similar al usado en la creación, tiene dos opciones; el botón cancelar regresa a la interfaz de crear sede y el botón guardar modifica el registro.

Figura 20. Formulario de editar sede (Primer Prototipo)



El formulario de edición de sede se muestra dentro de una ventana con el título "Sede". El contenido principal es un cuadro con el título "Editar Sede". Este cuadro contiene los siguientes campos:

- Código Sede: un campo de texto con el valor "7".
- Nombre Sede: un campo de texto con el valor "Bucaramanga".
- Vigencia: un menú desplegable con el valor "Activo".
- Descripción: un campo de texto con el valor "Es la sede de...".

Debajo de los campos se encuentran dos botones: "Guardar" y "Cancelar".

## Acción Eliminar

En las interfaces de eliminar registros, se muestran los datos completos del registro específico que se desea eliminar, con las opciones de eliminar y cancelar, al hacer click en eliminar, eliminara el registro automáticamente, por el contrario si hace click en cancelar, re direccionará al usuario a la interfaz de crear.

Figura 21. Formulario de eliminar sede (Primer Prototipo)

The image shows a software window titled "Sede" containing a sub-form titled "Eliminar Sede". The sub-form has four input fields: "Codigo Sede" with the value "7", "Nombre Sede" with the value "Bucaramanga", "Vigencia" with the value "Activo", and "Descripción" with the value "Es la sede de...". Below these fields are two buttons: "Eliminar" and "Cancelar".

#### 4.1.2 Segundo prototipo

El segundo prototipo implementó para cada interfaz de Sede, Edificio, Equipos de comunicación, etc., las acciones básicas de creación, visualización, modificación y eliminación de registros, así como la asociación de elementos derivados (modelo maestro-detalle). Se tomará como ejemplo el módulo de equipos de comunicación

#### Equipos de Comunicación

Se elaboró la interfaz con sus respectivas acciones las cuales permiten la asociación de sedes, edificio, centro de telecomunicaciones, gabinete/Rack, Patch panel, a continuación se muestra la interfaz.

Figura 22. Formulario Crear Equipo Comunicación (Primer Prototipo)

**Equipo Comunicación**

Sede --Seleccionar Sede--  
Edificio --Seleccionar Edificio--  
Centro Telecomunicaciones --Centro Telecomu --  
Gabinete / Rack --Selecione Gabinete / Ra  
Patch Panel --Selecione Patch Panel --

**Crear Equipo Comunicación**

Código Equipo Comunicación  
Nombre Equipo Comunicación  
Equipo Comunicación Padre -- Seleccione Código Padre  
Tipo Dispositivo --Selecione Tipo Dispositi  
Marca --Selecione Marca --  
Modelo  
MAC  
Número Inventario  
Serial  
Modularidad -- Seleccione Modularidad--  
Soporte IPv6 -- Seleccione Si/No--  
Número Total de puertos  
Descripción  
Fecha Inicio Vigencia -- DD/MM/AA --  
Fecha Final Vigencia -- DD/MM/AA --

Guardar Cancelar




Una vez ingresados los datos del formulario, con el botón guardar se almacenan en base de datos, seguido, aparece una tabla con el listado de los equipos registrados hasta el momento. Cada Equipo de Comunicación cuenta con cuatro acciones: Detalle, editar, eliminar y asociar puerto al equipo de comunicación. Esta tabla se muestra únicamente si ya hay registros creados.

Figura 23. Listado Equipo Comunicación (Primer Prototipo)

Listado Equipo Comunicación				
Nombre Equipo Comunicación	Número Inventario	Modularidad	Descripción	Acciones
				   

A continuación se presenta el resumen de las acciones más comunes en los formularios:

Figura 24. Resumen de las acciones (Primer Prototipo)

Acción	Detalles
 <b>Ver detalle</b>	Muestra los datos del registro
 <b>Editar</b>	Permite modificar todos los datos de un registro.
 <b>Eliminar</b>	Elimina el registro

## 4.2 PROTOTIPO FUNCIONAL

El prototipo final cumple a cabalidad con todos los objetivos estipulados en el plan del proyecto, incluyendo las políticas de seguridad instauradas por la DSI de la

Universidad Industrial de Santander. El esquema de seguridad está basado en la estructura de roles usuarios, el cual se define ampliamente en el numeral 4.3

Este prototipo del sistema de información SIMANRED se deriva del No funcional, expuesto en el numeral anterior, con ligeros cambios que se reflejan en la interfaz definitiva de la aplicación. El prototipo final abarca las siguientes funcionalidades:

- Administración completa de la infraestructura de la red, con interfaces agradables y del fácil uso al usuario final.
- Gestión de consultas sobre información específica de alcance de la red de datos institucional.
- Validación de toda la información en el proceso de formulación y en el manejo de pistas de auditoría que permiten visualizar los cambios de datos.

Para ser más específicos, se escogen dos módulos del prototipo final que se detallan a continuación.

### **Módulo de Sedes**

Este módulo cuenta con el formulario de creación, detalle, modificación y eliminación de sedes. Este contenido se encuentra en el menú Estructura Física, submenú Sedes (ver figura 25). Al hacer click en sedes, aparece el formulario de creación, el cual se visualiza en el numeral 1 con sus respectivos campos (Ver figura 26).

Figura 25. Menú Estructura Física (Prototipo Final)

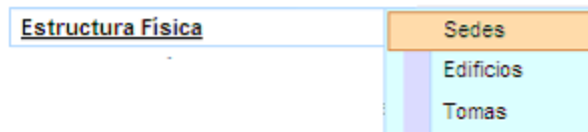


Figura 26. Formulario crear sede (Prototipo Final)

**Crear Sede**

Codigo Sede:

Nombre Sede:

Descripción:  1

Vigencia:

Listado de sedes			
Nombre	Vigencia	Acciones	
CAMPUS CENTRAL	Activo		
BUCARICA	Activo		
FACULTAD DE SALUD	Activo		
BARBOSA	Activo		
BARRANCABERMEJA	Activo		
GUATIGUARA	Activo		
MALAGA	Activo		

Total registros: 7

En el numeral 2, se encuentra el listado de las sedes que se han almacenado en el sistema, cuenta con las acciones básicas de: detalle, modificar y eliminar, explicadas en detalle en el primer prototipo (vea 4.1.1 Primer prototipo )

## Módulo Centro de Telecomunicaciones

El prototipo final de centro de telecomunicaciones fue uno de los que más cambió respecto al prototipo inicial, especialmente en la forma de acceder al formulario y realizar las acciones de Modificar, Eliminar, Ver Detalle y Agregar centro derivado.

Para acceder a este módulo, se debe escoger en el menú Estructura de Centro de Telecomunicaciones, la opción Centro de Telecomunicaciones (ver figura 27).

A continuación, aparecerá un formulario donde se elige una sede y edificio, previamente registrados y finalmente filtra los centros de telecomunicaciones asociados, brindando la posibilidad de crear nuevos centros, o editar los existentes (ver figura 28).

Figura 27. Menú Estructura de Telecomunicaciones (Prototipo Final)

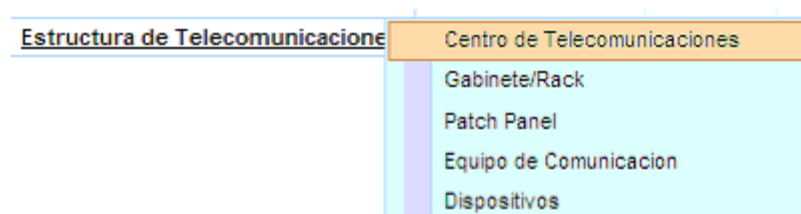


Figura 28. Formulario Centro Telecomunicaciones (Prototipo Final)

Centro de Telecomunicaciones

Sede: GUATIGUARA

Edificio: PLANOS GENERALES

Crear Centro de Telecomunicaciones Primario

- centro planos 1
- centro 65
- centro deriva
- centro57

En la figura anterior, el link que señala el numeral 1, permite la creación de un centro de telecomunicaciones que sea primario en el edificio. El formulario de creación se puede detallar en la figura 29.

En el numeral 2, se encuentran los centros de telecomunicaciones registrados, que son mostrados bajo un esquema de árbol que facilita la navegación y visualización de las dependencias entre los centros de telecomunicaciones. Al hacer click derecho en alguno de ellos, aparecerá un menú con las opciones de modificar, eliminar, ver centro y agregar centros derivados (ver figura 30).

Figura 29. Formulario de Crear Centro de telecomunicación (Prototipo Final)

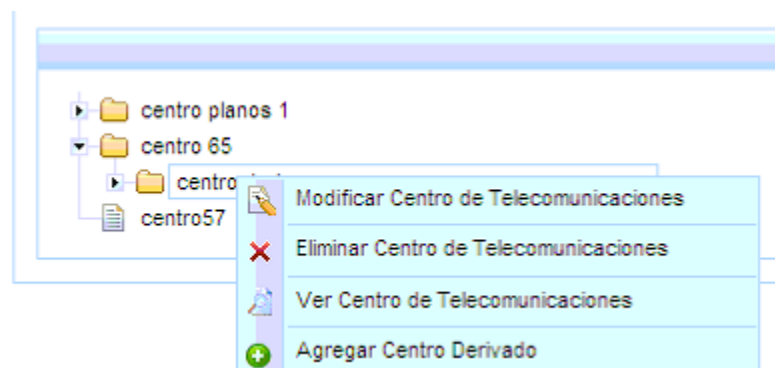
**Crear**

Sede GUATIGUARA  
Edificio PLANOS GENERALES

Nombre Centro de Telecomunicaciones Centro Guatiguta 1  
Indicativo Centro Primario  
Descripción Este centro ....  
Ubicación Esta ubicado ....  
Tipo Cableado Cable UTP - 5E  
Dimensión 5x6x7  
Distancia Centro Cableado 12  
Fecha Inicio vigencia mar 30, 2011  
Fecha Fin vigencia

Guardar Cancelar

Figura 30. Menú Centro Telecomunicaciones (Prototipo Final)



En el menú de modificar, se puede cambiar la información del centro de telecomunicaciones. Cuenta con dos botones, el de modificar y cancelar (ver figura 31).

En la opción de agregar centro derivado, se crean los centros de telecomunicaciones derivados, el formulario es muy similar al de creación de los centros de telecomunicaciones primarios, la diferencia radica en el indicativo del centro, que aparecerá como “derivado” (ver figura 32)

Figura 31. Formulario de Modificar Centro (Prototipo Final)

The image shows a web-based form titled "Modificar" for editing a telecommunications center. The form is organized into several sections:

- Header:** "Modificar" with a close button (X).
- Location Information:** Sede: GUATIGUARA, Edificio: PLANOS GENERALES.
- Main Form Fields:**
  - Nombre Centro de Telecomunicaciones:** Input field containing "centro deriva".
  - Indicativo Centro:** Dropdown menu set to "Derivado".
  - Descripción:** Text area with placeholder "Este centro ...".
  - Ubicación:** Text area with placeholder "Esta ubicado ...".
  - Tipo Cableado:** Dropdown menu set to "Cable UTP - Monomodo".
  - Dimensión:** Input field containing "6x3x2".
  - Distancia Centro Cableado:** Input field containing "10.0".
  - Fecha Inicio vigencia:** Date picker set to "mar 30, 2011".
  - Fecha Fin Vigencia:** Date picker set to "abr 28, 2011".
- Buttons:** "Guardar" and "Cancelar" buttons at the bottom.

Figura 32. Formulario Agregar centro Derivado (Prototipo Final)

The image shows a web form titled "Crear" for adding a derived telecommunication center. The form is organized into several sections:

- Sede:** GUATIGUARA
- Edificio:** PLANOS GENERALES
- Nombre Centro de Telecomunicaciones:** Centro Planos 2
- Indicativo Centro:** Derivado (highlighted with a red oval)
- Descripción:** Este centro ...
- Ubicación:** Esta Ubicado
- Tipo Cableado:** Cable UTP - 12 Hilos
- Dimensión:** 5x7x8
- Distancia Centro Cableado:** 10
- Fecha Inicio vigencia:** feb 3, 2011
- Fecha Fin Vigencia:** abr 20, 2012

At the bottom of the form, there are two buttons: "Guardar" and "Cancelar".

### 4.3 ESQUEMA DE SEGURIDAD UNIVERSIDAD INDUSTRIAL DE SANTANDER

Para cualquier proyecto desarrollado es necesario implementar un sistema de seguridad, y se tiene que identificar los roles asociados a los usuarios potenciales del Sistema de Información de la infraestructura de la red de datos institucional SIMANRED.

Una vez establecidos los roles se tienen que definir los permisos y menús de cada rol, tareas que posteriormente debe realizar el administrador del sistema de seguridad.

Cuando la información descrita en el párrafo anterior se encuentra registrada en la base de datos del sistema de seguridad se procede a configurar la aplicación de simanred con el fin de adaptarla al esquema de seguridad definido específicamente para dicha aplicación.

Para el sistema de información de la red de datos institucional existe dos roles principales:

- ✓ Administrador
- ✓ Usuario

Para este proyecto se utiliza el esquema de seguridad definido por la División de Servicios de Información para los diferentes sistemas de información que apoyan la gestión de la Universidad Industrial de Santander, el cual está basado en la estructura de roles – usuarios.

#### **4.3.1 Estructura de la Base de Datos soporte**

La base de datos que soporta el esquema de seguridad contempla básicamente las siguientes tablas:

**Sistema:** Contiene información de los sistemas de información de la universidad. Para cada sistema se especifica: Nombre, descripción del sistema, fecha y hora de creación en la base de datos, fecha y hora de inicio de vigencia del sistema, fecha y hora de cierre de vigencia del sistema.

**Rol:** contiene información de los diferentes roles definidos para cada sistema de información, como: Nombre asignado al rol, descripción del rol, fecha y hora de

creación, fecha y hora de inicio de vigencia del rol, fecha y hora de cierre de vigencia del rol.

**Usuario:** Contiene información de los posibles usuarios de los sistemas de información. Entre esta información está: tipo y número de documento de identidad del usuario, fecha y hora de creación del usuario, fecha y hora de inicio de vigencia del usuario, fecha y hora de cierre de vigencia del usuario.

**Sistema-rol:** Contiene los roles definidos para cada uno de los sistemas de información, indicando: rol, sistema, fecha y hora de creación del rol – sistema, fecha y hora de inicio de vigencia del rol en el sistema, fecha y hora de cierre de vigencia del rol en el sistema.

**Rol-usuario:** Contempla los usuarios asociados a cada uno de los roles definidos, considerando: Rol, usuario, fecha y hora de creación del rol – usuario, fecha y hora de inicio de vigencia del usuario en el rol, fecha y hora de cierre de vigencia del usuario en el rol.

**Menú-rol-sistema:** Contiene los menús asociados a los roles en los distintos sistemas de información, contemplando: Sistema de información, nombre del menú, descripción del menú, fecha y hora de creación del menú, fecha y hora de inicio de vigencia del menú asociado al rol, fecha y hora de cierre de vigencia del menú asociado al rol.

**Opción-menú-rol:** Contempla las opciones definidas para cada una de los posibles menús establecidos para cada sistema de información. Contiene: Nombre de la opción, descripción de la opción, nombre del menú superior, nombre del menú que contiene la opción, nombre del programa a ejecutar cuando la opción es la de más bajo nivel, fecha y hora de creación de la opción del menú, fecha y hora de inicio de vigencia de la opción, fecha y hora de cierre de la opción.

**Tabla-sistema:** Contiene información de las tablas que conforman la base de datos que soporta cada uno de los sistemas de información. Considera: Sistema de información, nombre de la tabla, descripción de la tabla.

**Tipo-permiso:** Establece para cada tabla de un sistema de información, los roles que tienen permisos para incluir registros, para modificar registros o para eliminar registros en ella. Contiene: Sistema de información, nombre de la tabla, clase de permiso (inclusión, modificación, eliminación de registros), fecha y hora de creación del permiso, fecha y hora de inicio de vigencia del permiso, fecha y hora de fin de vigencia del permiso.

**Acceso-tabla:** Define para las tablas de un sistema de información si un rol tiene permiso sobre toda la información de la tabla o sobre una parte de esta. Considera: Sistema, nombre de la tabla, clase de acceso (total, parcial), fecha y hora de creación del permiso, fecha y hora de inicio de vigencia del permiso, fecha y hora de fin de vigencia del permiso.

**Atributo-tabla:** Establece los atributos sobre los cuales se debe controlar el acceso a una tabla, cuando a un rol se le concede permiso para hacer uso parcial de la información existente en una tabla. Contiene: Sistema de información, nombre de la tabla, nombre del atributo sobre el cual se controla el acceso a la tabla, descripción del atributo, fecha y hora de creación del atributo, fecha y hora de inicio de vigencia del atributo, fecha y hora de fin de vigencia del atributo.

**Valor-atributo-proceso:** Contiene los valores que deben tener los atributos definidos en cada tabla en la tabla atributo – tabla que permiten el acceso a la información asociada a estos valores. Específica: Sistema de información, nombre de la tabla, nombre del atributo, valor del atributo, descripción, fecha y hora de

creación del valor del atributo, fecha y hora de inicio de vigencia del valor del atributo, fecha y hora de fin de vigencia del valor del atributo.

**Acceso-sistema:** Contempla el histórico de acceso que un usuario ha realizado a un sistema, identificando las opciones que ha seleccionado. Contiene: Login de usuario, rol, identificación de la sesión, sistema, opción seleccionada, fecha y hora de ingreso, fecha y hora de salida.

### **4.3.2 Entorno de Navegación**

Para cada sistema de información, la UAA responsable define los roles necesarios para el adecuado uso del sistema de información de acuerdo a las funciones que realice y establece los usuarios asociados a cada uno de ellos.

Para cada rol se define el menú de inicio, el cual le permite a cada usuario que hace parte de este rol, empezar la navegación por las distintas opciones que le ofrece el sistema, hasta llegar al nivel más bajo en el cual se ejecuta el proceso que soporta la actividad que desea realizar.

Este entorno está soportado por las siguientes tablas de las base de datos del esquema de seguridad: Rol, usuario, sistema, sistema-rol, usuario-rol, menú-rol, opción-menú rol, descritas en “ESTRUCTURA DE LA BASE DE DATOS SOPORTE”.

### **4.3.3 Entorno de Control de Datos**

Para los roles definidos en cada uno de los sistemas de información se especifican las tablas a las cuales puede acceder, el tipo de transacción que

puede realizar sobre estas tablas (inclusión, modificación o eliminación de registros), si tiene acceso total o parcial a la información que contiene la tabla.

Para el acceso a la información de la tabla de manera parcial, se debe establecer el atributo o atributos seleccionados, los valores que estos atributos deben tener para autorizar el acceso solicitado.

Este entorno está soportado por las siguientes tablas de las base de datos del esquema de seguridad: Rol, usuario, sistema, sistema-rol, usuario-rol, tabla-sistema, tipo-permiso, acceso-tabla, atributo-tabla, valor atributo proceso, descritas en “ESTRUCTURA DE LA BASE DE DATOS SOPORTE”.

#### **4.3.4 Auditoría**

Todas las tablas que conforman la base de datos soporte del esquema de seguridad tienen el historial de las transacciones realizadas sobre cada una de ellas.

El historial de las transacciones de cada tabla contiene información de los registros incluidos en la tabla, de los registros modificados y de los registros eliminados. Adicionalmente, en cada transacción se especifica: Fecha de la transacción, hora de la transacción, tipo de transacción (I/U/D), tipo y número de documento de identidad del usuario que realizó la transacción, login, rol asociado, dirección IP y MAC del equipo desde el cual llevó a cabo la transacción.

#### **4.3.5 Estándares de la DSI<sup>22</sup>**

---

<sup>22</sup> Fuente: Estándares de la División de Servicios de Información de la Universidad Industrial de Santander

A continuación se detallan los estándares que se consideraron más importantes, la lista completa se puede encontrar en el documento de estándares definido por la DSI.

## **Nombre de archivos, variables, constantes, atributos, métodos y parámetros**

Reglas de sintaxis generales:

- Todos los nombres de los identificadores deben estar en español.
- Siempre se deben utilizar nombres que sean claros, concretos y libres de ambigüedades. Usando palabras completas evitando acrónimos y abreviaturas.
- Los nombres deben estar definidos sin espacios en blanco, sin guiones ( \_ , -), ni comillas ( “ , ’ ), sin operadores ( + , - , / , \*), sin tildes, utilizar la n en vez de la ñ y sin caracteres especiales.
- No se deben diferenciar dos identificadores solo con numerales en cualquier posición. Ejemplo: contador1, contador2, 1contador, 2contador.

Los nombres dentro del código fuente siguen los parámetros establecidos en el estándar general de nombres, con las siguientes particularidades.

- Los nombre de los parámetros de los métodos empiezan con guión de piso, el resto del nombre sigue el estándar para nombres de variables. Una vez dentro del código fuente, se deben asignar a una nueva variable con el mismo nombre pero sin el guión de piso.
- Los nombres de constantes o variables finally, se escriben en mayúscula sostenida separando las palabras por guiones de piso.

## **Sentencias simples:**

Se debe escribir solo una sentencia por línea, y no escribir más de una separadas por punto y coma en la misma línea sin importar lo cortas que puedan ser.

### **Sentencias compuestas:**

Las sentencias compuestas son sentencias que contienen una lista de sentencias encerradas entre llaves "{" y "}" y deben seguir las siguientes reglas de formato.

- Las sentencias internas deben estar tabuladas un nivel más que la sentencia compuesta.
- Las llaves se usan en todas las sentencias compuestas, incluidas las sentencias únicas, cuando forman parte de una estructura de control, como una sentencia if-else o un bucle for. Esto hace más fácil introducir nuevas sentencias sin provocar errores accidentales al olvidarse añadir las llaves.

#### **4.3.6 Documentación básica del programa Fuente de un EJB de Sesión**

En la labor desarrollo la documentación juega un papel muy importante para el mantenimiento de software. Esto es aún más cierto en proyectos grandes, en los que pueden estar involucrados demasiados desarrolladores, todos con culturas de desarrollo diferentes. Aunque todos estos desarrollos cuentan con estándares muy definidos, siempre puede ser necesaria la aclaración sobre ciertos fragmentos o secciones de código.

Históricamente todos los lenguajes de programación han contado con facilidades para la inserción de comentarios dentro del código fuente. Estos comentarios no son compilados puesto que no tienen utilidad funcional pero cuando otro desarrollador ve el código fuente, son necesarios para entender partes que no son claras a simple vista. Java heredó de C la forma de hacer tales comentarios:

Para líneas

// comentario

Para bloques de comentarios

```
/*
```

```
Para bloques enteros de comentarios, con  
varias líneas
```

```
*/
```

Sin embargo estas herramientas no son necesarias para documentar proyectos largos, como es el caso de las aplicaciones empresariales. Para esto Java ofrece Javadoc, que es un generador de documentación a partir de código Java. El código que Javadoc utiliza es aquel que encuentre entre:

```
/** documentación */
```

El propósito de este modelo de documentación no es desaprobar las formas tradicionales, que siguen siendo válidas sobre secciones pequeñas de código. Lo que busca Javadoc es generar documentación clara y precisa sobre clases, sus miembros y sus métodos. Así pues, los comentarios especificados mediante el formato anterior van en la definición de clases, sus métodos y miembros.

### **Ejemplo de documentación de método**

Figura 33. Código Fuente

```
/**
 * Este método retorna los Centros de Telecomunicaciones, ordenados por posición
 * @param aEdificio, es el edificio que se le pasa como parametro para filtrar los centros de telecomunicaciones
 * @return la lista de los centros de telecomunicaciones
 * @throws DSIException
 */
public List<CentroTelecomunicacion> getCentroTelecomunicacionesPscn(Edificio aEdificio) throws DSIException {
    List<CentroTelecomunicacion> centroTelecomunicaciones = new ArrayList<CentroTelecomunicacion>();

    try {
        if (aEdificio != null && aEdificio.getCodigoEdificio() != null) {
            String jpql = "";
            jpql += " SELECT ct FROM CentroTelecomunicacion ct ";
            jpql += " WHERE ct.centroTelecomunicacionPdr is null ";
            jpql += " AND ct.edificio.codigoEdificio=:edificio ";
            jpql += " ORDER BY ct.posicion ASC ";
            Query query = this.em.createQuery(jpql).setParameter("edificio", aEdificio.getCodigoEdificio());
            centroTelecomunicaciones = query.getResultList();
        }
    } catch (Exception e) {
        throw new DSIException(e);
    }
    return centroTelecomunicaciones;
}
```

Antes de la firma del método va una descripción básica de la tarea que cumple el mismo. Pero en esta documentación se pueden especificar otras cosas. Por ejemplo, mediante **@paramaEdificio** la documentación está haciendo referencia al parámetro aEdificio que llega al método y con **@returncentroTelecomunicaciones** hace referencia al valor, en este caso la lista: centroTelecomunicaciones, que retorna el método.

## 5. CONCLUSIONES

La nueva versión del sistema de mantenimiento de la red de datos institucional (SIMANRED) se convierte en una herramienta de gran utilidad, que soluciona los problemas de eficiencia, tiempos de respuesta y usabilidad del anterior sistema.

El exhaustivo análisis del diseño de la interfaz de usuario del nuevo sistema, se ve reflejado en el dinamismo de los programas que lo conforman, su facilidad de uso, y la disposición de elementos visuales como las ayudas en línea y la estructura de directorios.

Debido a la estructura dinámica concebida en el desarrollo, SIMANRED se adapta a la actual infraestructura de red de la Universidad Industrial de Santander, y a su vez permite modificaciones futuras, como el uso de IPV6 y cambios producto de la innovación tecnológica y de la ampliación de la cobertura institucional.

Con la elaboración y desarrollo de este proyecto se logró conocer la infraestructura de red de la institución, experiencia que reforzó nuestros conocimientos de redes al estudiar a fondo la relación entre los elementos, componentes hardware y software de los equipos de comunicación y los estándares inherentes a su instalación en la Universidad Industrial de Santander.

La metodología basada en prototipos facilitó la interacción desarrollador-usuario final, esto se vio reflejado en la fácil obtención de requerimientos y en la satisfacción y aceptación de los prototipos desarrollados.

La nueva versión del sistema de información para la administración de la red de datos institucional realiza de manera eficiente la gestión de todos los elementos

de red, brindando seguridad y confiabilidad en la información de los datos manejados en el sistema.

Debido a la arquitectura multi-capa de la tecnología Java EE5, se ha identificado un alto grado de complejidad en el desarrollo de software, esto puede retrasar la codificación de los programas, ya que existen muchos conceptos que la mayoría de programadores principiantes no conocen. El uso de frameworks de desarrollo de nivel empresarial, específicamente en este contexto: Seam y Rich Faces contrarrestan en gran medida las dificultades a las que usualmente se exponen los programadores.

Los conocimientos adquiridos en las capacitaciones de Java Básico, Java EE5 y en la ejecución de este proyecto; y con la experiencia en el desarrollo de software utilizando frameworks aportan positivamente a nuestra competitividad en la industria del software.

La revisión exhaustiva de los diagramas UML, prototipos y código fuente por parte de expertos en la DSI, ayuda a garantizar la calidad del software y ayuda a fomentar buenas prácticas en todas las fases del desarrollo.

## 6. RECOMENDACIONES

- Java EE5 facilita en gran medida el desarrollo de aplicaciones tipo empresarial de alta calidad, se recomienda que se planteen métodos, formas o actividades para que los estudiantes de la Escuela de Ingeniería de Sistemas e Informática puedan ampliar sus conocimientos en estas tecnologías de punta a nivel mundial.
- Fomentar el vínculo entre la División de Servicios de Información y la Escuela de Ingeniería de Sistemas e Informática para compartir experiencias y conocimientos en el desarrollo de software.
- Establecer un plan de formación para las personas que actualmente manejan la red de datos de la Universidad.

## 7. BIBLIOGRAFIA

Análisis y Diseño Estructurado y Orientado a Objetos de Sistemas Informáticos. Quinta Edición. Universidad Carlos III de Madrid, España. McGraw-Hill, 2003

BERZAL GALIANO, Fernando. Doctor en Informática. Universidad de Granada, España. Relaciones entre clases: Diagramas de clase UML. {En línea}. { Consultado marzo de 2011}. Disponible en: <http://elvex.ugr.es/decsai/java/pdf/3C-Relaciones.pdf>

CCNA Exploration, Network Fundamentals, versión 4, Capítulo 2 Comunicación a través de la red, sección 2.2.1.1 LAN (Red de Área Local).

ICONTEC INTERNATIONAL. EL COMPENDIO DE TESIS Y OTROS TRABAJOS DE GRADO. {En línea}. {Consultado febrero de 2011}. Disponible en: [http://www.ICONTEC.org/BancoConocimiento/C/compendio\\_de\\_tesis\\_y\\_otros\\_trabajos\\_de\\_grado/compendio\\_de\\_tesis\\_y\\_otros\\_trabajos\\_de\\_grado.asp?CodIdioma=ESP](http://www.ICONTEC.org/BancoConocimiento/C/compendio_de_tesis_y_otros_trabajos_de_grado/compendio_de_tesis_y_otros_trabajos_de_grado.asp?CodIdioma=ESP)

KING, Gavin; MUIR, Pete; RICHARDS, Norman; BRYZAK, Shane; YUAN, Michael; OUNGSTROM, Mike; BAUER, Christian; BALUNAS, Jay; ALLEN, Dan; ANDERSEN, Max Rydahl; BERNARD, Emmanuel; KARLSSON, Nicklas; ROTH, Daniel; DREES, Matt; ORSHALICK, Jacob; and NOVOTNY, Marek. Seam a framework for enterprise java 2.2.0.GA. {En línea}. {Consultado marzo de 2011}. Disponible en: <http://www.seamframework.org/Seam2/Documentation>

SCHMULLER, Joseph. Aprendiendo UML en 24 horas. Primera edición. Editorial Prentice Hall, 2001. 5-18p.

SICUMA: Sistemas de Información Cooperativos Universidad de Málaga. España. Tutorial de JavaServer Faces. {En línea}. {Consultado Agosto de 2010}. Disponible en: <http://www.sicuma.uma.es/sicuma/formacion.jsp>

Sparx Systems, Pty Ltd. Tutorial UML 2. {En línea}. {Consultado Noviembre de 2010}. Disponible en: [http://www.sparxsystems.com/resources/uml2\\_tutorial/index.html](http://www.sparxsystems.com/resources/uml2_tutorial/index.html)

SUNMICROSYSTEMS, Inc. The Java EE 5 Tutorial. {En línea}. {Consultado 15 Junio de 2010}. Disponible en: <http://download.oracle.com/javase/5/tutorial/doc/>  
DE AMESCUA SECO, Antonio; CUADRADO GALLEGO, Juan José; ERNICA LAFUENTE, Emilio; GACÍA GUZMÁN, Javier; GARCÍA SÁNCHEZ, Luis; MARTÍNEZ FERNÁNDEZ, Paloma; SÁNCHEZ SEGURA M<sup>a</sup> Isabel. Análisis y Diseño Estructurado y Orientado a Objetos de Sistemas Informáticos. Quinta Edición. Universidad Carlos III de Madrid, España. McGraw-Hill, 2003. 3-5p, 25-28p, 64p.