

Trabajo de Grado
***“ESPECIFICACIÓN DE REQUISITOS PARA EL DISEÑO Y DESARROLLO DE
SISTEMAS TIC EN INGENIERÍA DEL SOFTWARE, UTILIZANDO MODELOS
SUSTENTADOS EN ARQUITECTURA (MDA)”***

PRESENTADO ANTE:
Comité de Trabajos de Grado
Escuela de Ingenierías Eléctrica, Electrónica y
Telecomunicaciones

Por
LUISANGELA OROZCO RAMIREZ

Bucaramanga, Mayo de 2009

***ESPECIFICACIÓN DE REQUISITOS PARA EL DISEÑO Y DESARROLLO DE
SISTEMAS TIC EN INGENIERÍA DEL SOFTWARE, UTILIZANDO MODELOS
SUSTENTADOS EN ARQUITECTURA (MDA)***

AUTORA:

LUISANGELA OROZCO RAMIREZ

Trabajo presentado como parte de los requisitos para optar al título de
Ingeniera electrónica

DIRECTOR:

PHD. RICARDO LLAMOS A VILLALBA

ESCUELA DE INGENIERÍAS ELÉCTRICA ELECTRÓNICA Y
TELECOMUNICACIONES

CODIRECTOR:

ING. SERGIO ENRIQUE MÉNDEZ ACEROS

CENTRO DE INNOVACIÓN Y DESARROLLO PARA LA INVESTIGACIÓN EN
INGENIERÍA DEL SOFTWARE – CIDLIS

**FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA**

2009

DEDICATORIA

*A Dios padre, ser misericordioso y amigo.
A mi padre Jorge "El negro Orozco" quien siempre ha sido
mi modelo a seguir y me ha brindado su apoyo incondicional.
A mi madre Alicia quien siempre me ha brindado su amor
y fortaleza para seguir adelante,
A mi novio Sergio quien es el motor de mi vida.
A toda mi familia y hermanos,
en especial a mi tía Golla y a Elvi por su infinito amor.
A todos mis amigos y profesores,
con quienes siempre tuve algo que compartir y aprender,
Y a todos aquellos que de alguna forma
estuvieron presentes en esta etapa de mi vida.*

AGRADECIMIENTOS

La autora expresa sus agradecimientos a todas las personas que contribuyeron en el desarrollo de este trabajo.

En especial a mi Director Ricardo Llamosa Villalba y Codirector Sergio Enrique Méndez Aceros, por su apoyo y asesoría.

A mi familia y amigos por el apoyo constante, por las palabras de ánimo, y por su comprensión en el desarrollo del proyecto.

Al CIDLIS y la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones, así como a la Universidad Industrial de Santander, por brindarme la oportunidad de adquirir una gran formación profesional.

RESUMEN

TÍTULO: ESPECIFICACIÓN DE REQUISITOS PARA EL DISEÑO Y DESARROLLO DE SISTEMAS TIC EN INGENIERÍA DEL SOFTWARE, UTILIZANDO MODELOS SUSTENTADOS EN ARQUITECTURA (MDA)*

AUTORA: LUISANGELA OROZCO RAMIREZ**

PALABRAS CLAVES: Ingeniería del software, ingeniería de requisitos, especificación de requisitos, Tecnologías de Información y Comunicación (TIC), Modelos Orientados por Arquitectura (MDA), Lenguaje de Modelado Unificado (UML).

DESCRIPCIÓN:

El presente trabajo contiene el planteamiento de un procedimiento para la especificación de requisitos de sistemas TIC, enmarcado bajo el contexto de la ingeniería del software utilizando modelos orientados en arquitectura (MDA), con el cual se pretende mejorar el proceso y los resultados del levantamiento de requisitos, y en sí, facilitar el proceso de desarrollo software.

Para cumplir los objetivos planteados, este trabajo estuvo enmarcado en varias etapas, la primera de ellas fue la recopilación de la información que además de fundamental, fue la base para el desarrollo y culminación de este, aquí se analizaron los temas referentes a la ingeniería del software, realizando una introducción al modelado de objetos y prestando especial atención al ciclo de vida de un producto enfatizando sobre la etapa de especificación de requisitos, luego se analiza el concepto de modelos orientados por arquitectura (MDA) como herramienta de desarrollo software basada en modelos.

Cuando se logró el reconocimiento del tema a tratar, se realizó una combinación entre los conceptos más relevantes a ser aplicados y a partir de esto se desarrolló un ejemplo que demuestra la validez del procedimiento basado en arquitectura MDA para la especificación de requisitos de un sistema de TIC en ingeniería de software.

Por último, se planteó la estructura del procedimiento para la especificación de requisitos de este tipo de sistemas, utilizando MDA y teniendo en cuenta el ciclo de vida de un producto en ingeniería de software.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: PhD. Ricardo Llamasa Villalba. Codirector: Ing. Sergio Enrique Méndez Aceros.

ABSTRACT

TITLE: SPECIFICATION REQUIREMENTS FOR ICT SYSTEMS DESIGN AND DEVELOPMENT IN SOFTWARE ENGINEERING, USING MODEL DRIVE ARCHITECTURE (MDA)*

AUTHOR: LUISANGELA OROZCO RAMIREZ**

KEY WORDS: Software engineering, requirements specification, Information and Communication Technologies (ICT), requirements specification, Model Driven Architecture (MDA), Unified Modeling Language (UML).

DESCRIPTION:

This work contains an approach for the ICT systems specification of requirements, framed in context of software engineering using Model Driven Architecture (MDA), which seeks to improve process and results of eliciting requirements, and as such, facilitates the process of software development.

To meet the objectives, this work was framed in several stages, first stage was the compilation of information that, in addition to fundamental, was the basis for this work development and completion, software engineering related issues were analyzed making an introduction to object modeling and special attention to product life cycle, with emphasis on requirements specification stage, then discusses the concept of model driven architecture (MDA) as a tool for developing software-based models.

When the recognition of the issue to be addressed was, a combination of most important concepts to be applied was and since this, an example that demonstrates the validity of the procedure based on MDA architecture for ICT system requirements specification in software engineering was developed.

Finally, it was the structure of requirements specification procedure for such systems, using MDA and taking into account the software engineering product life cycle.

* Degree Work

** Facultad de Ingenierías Físico-Mecánicas. Electricity, Electronics and Telecommunications Engineerings School. Manager: PhD. Ricardo Llamosa Villalba. Codirector: Eng. Sergio Enrique Méndez Aceros.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
ESTADO DEL ARTE	4
1. LAS TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TIC) Y LA INGENIERÍA DEL SOFTWARE	5
1.1. DEFINICIÓN DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN (TIC)	5
1.2. GESTIÓN DE PROYECTOS DE TIC	7
1.2.1. Atributos de un proyecto TIC	7
1.3. CICLO DE VIDA DE UN PRODUCTO TIC.....	8
1.4. INGENIERÍA DEL SOFTWARE	9
1.4.1. Generalidades del proceso de desarrollo software.....	10
1.4.2. Estándares de Ingeniería de Software.....	11
1.5. CICLO DE VIDA DEL SOFTWARE.....	12
1.6. ANÁLISIS ORIENTADO A OBJETOS.....	15
2. LA INGENIERÍA DE REQUISITOS.....	16
2.1. IMPORTANCIA DE LOS REQUISITOS	16
2.2. CLASIFICACIÓN DE LOS REQUISITOS.....	17
2.2.1. Requisitos Funcionales.....	17
2.2.2. Requisitos No Funcionales	18
2.2.3. Requisitos de interfaz	18
2.3. PROCESOS EN LA ESPECIFICACIÓN DE REQUISITOS	19
2.3.1. Visión del Sistema	20

2.3.2. Obtención y Análisis	21
2.3.3. Especificación	23
2.3.4. Verificación y validación.....	26
2.3.5. Gestión de configuración	26
3. ARQUITECTURA DIRIGIDA POR MODELOS (MDA).....	28
3.1. CONTEXTO	28
3.2. INTRODUCCIÓN A MDA	29
3.3. MODELOS EN MDA	31
3.3.1. Modelo Independiente de la Plataforma (PIM).....	31
3.3.2. Modelo Específico de la Plataforma (PSM)	32
3.3.3. Modelo de Código.....	32
3.4. ESTÁNDARES EN MDA	32
3.5. LA HERRAMIENTA ENTERPRISE ARCHITECT.....	35
3.5.1. Transformaciones MDA	36
3.6. PROPUESTA METODOLÓGICA PARA LA ESPECIFICACIÓN DE REQUISITOS DE SISTEMAS TICS UTILIZANDO MDA.....	37
3.6.1. Políticas del documento.....	37
3.6.2. Recolección de información.....	37
4. CASO DE ESTUDIO.....	39
“DISEÑO DEL CABLEADO ESTRUCTURADO PARA COMUNICACIÓN DE VOZ, DATOS Y VIDEO”	39
4.1. CABLEADO ESTRUCTURADO	39
4.1.1. Subsistemas del cableado estructurado	40

4.2. APLICACIÓN DEL PROCEDIMIENTO	41
4.2.1. “Visión del sistema”	41
4.2.2. Especificación de requisitos del sistema utilizando MDA	47
4.3. CONCLUSIONES.....	65
4.4. RECOMENDACIONES	67
BIBLIOGRAFÍA	68
ANEXOS	84

LISTA DE ESQUEMAS

Esquema 1. Atributos de un proyecto TIC.....	8
Esquema 2. Ciclo de vida de un producto TIC.	9
Esquema 3. Ciclo de vida del software.	13
Esquema 4. Procesos del ciclo de vida del software según ISO 12207.....	13
Esquema 5. Áreas de proceso en la Ingeniería de Requisitos.	19
Esquema 6. Síntesis del documento “Visión del sistema” del Sistema de Gestión de Calidad del CIDLIS.....	21
Esquema 7. Objetivos del proceso de la ingeniería de requisitos	27
Esquema 8. Proceso de desarrollo de software con MDA.	29
Esquema 9. Transformaciones MDA.....	36
Esquema 10.Procedimiento para la especificación de requisitos utilizando MDA.....	38

LISTA DE TABLAS

Tabla 1. Resultados en los proyectos TIC según Standish Group.....	7
Tabla 2. Definición de la ingeniería del software.....	10
Tabla 3. Estándares de Ingeniería de Software.	11
Tabla 4. Procesos en la Ingeniería del Software.....	14
Tabla 5. Errores en la obtención de los requisitos y sus consecuencias.....	17
Tabla 6. Requisitos no funcionales.....	18
Tabla 7. Requisitos de interfaz.	18
Tabla 8. Construcciones básicas de lenguaje de modelado de MOF.....	33
Tabla 9. Diagramas de modelado estructural de UML.	34
Tabla 10. Diagramas de modelado de comportamiento de UML.	35

LISTA DE ANEXOS

ANEXO A. Plantilla para la especificación de requisitos de sistemas TIC utilizando MDA	72
ANEXO B. Guía para el uso de la plantilla	78
ANEXO C. UML con Enterprise Architect.....	84

GLOSARIO

ACTOR: entidad externa que intercambia información con el sistema.

AGREGACIÓN: asociación que indica una relación completa entre dos clases.

ANÁLISIS ORIENTADO A OBJETOS: actividad que hace referencia al modelado con objetos.

ÁREA DE TRABAJO: oficinas en el edificio, donde los usuarios interactúan con los equipos.

ARQUITECTURA: estructura organizada de los componentes de un sistema y sus relaciones.

ASOCIACIÓN: relación entre dos o más clases que indica los vínculos posibles entre sus instancias.

ATRIBUTO: es una propiedad que posee una clase y tiene nombre.

CABLEADO: conjunto de cables, alambres, cordones y elementos de conexión.

CABLEADO ESTRUCTURADO: Forma ordenada y planeada de realizar cableados que permitan integrar servicios de telecomunicaciones.(datos, voz, video).

CANAL (REFERIDO A TELECOMUNICACIONES): trayectoria de transmisión al cual se conecta un equipo de aplicación específica.

CASO DE USO: secuencia de las interacciones entre uno o más actores y el sistema.

CICLO DE VIDA DEL SOFTWARE: son todas las actividades y tareas necesarias para desarrollar un sistema de software.

CLASE: abstracción de un conjunto de objetos que comparten las mismas características.

CLIENTE: persona o compañía interesada en el desarrollo del sistema.

CUARTO DE TELECOMUNICACIONES: área cerrada para alojar los equipos de red, terminaciones de cable y cableado de interconexión.

DESARROLLADOR: persona que cumple un papel en la especificación, diseño y construcción del sistema.

DOMINIO DEL PROBLEMA: representa todos los aspectos del problema del usuario durante las actividades de requerimientos y análisis.

DUCTO: canal cerrado para transportar y proteger cables de datos o alambres de corriente eléctrica generalmente usada para conducirlos bajo tierra o ahogado en concreto.

ESTEREOTIPO: texto encerrado entre paréntesis angulares para expandir el significado de un elemento del modelo.

GENERALIZACIÓN: tipo de relación entre una clase general y otra más especializada.

HERENCIA: técnica de reutilización en la cual una clase hija hereda todos los atributos y operaciones de una clase madre.

IMPLEMENTACIÓN: el modelo de objetos es traducido a código.

INFORMÁTICA: relación de equipos de software y hardware.

INFRAESTRUCTURA DE TELECOMUNICACIONES: conjunto de todos aquellos elementos de canalización que proporcionan el soporte básico para la distribución de todos los cables.

INGENIERÍA DE REQUISITOS: actividad que incluye la obtención de requisitos y el análisis.

INGENIERÍA DEL SOFTWARE: rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas a los problemas de desarrollo de software.

INSTANCIA: miembro de un tipo de dato específico.

INTERCONEXIÓN: conexión directa de un equipo a un bloque de conexión o panel de parcheo de la red de cableado estructurado.

LENGUAJE DE MODELADO UNIFICADO (UML): conjunto de notaciones estándar para la representación de modelos.

MODELADO: actividad en la que se construye una abstracción de un sistema, enfocándose en sus aspectos más interesantes.

MODELO: representación de la función y el comportamiento de un sistema.

MODELO DE OBJETOS: describe la estructura de un sistema desde el punto de vista de objetos, atributos, asociaciones y operaciones.

MODELO ORIENTADO POR ARQUITECTURA (MDA): Forma de construir software utilizando lenguaje de modelado a distintos niveles de abstracción.

OBJETO: instancia de una clase, tiene identidad y atributos.

OBTENCION DE REQRIMIENTOS: etapa en la que se define el propósito del sistema, produciendo el modelo funcional.

PANEL DE PARCHEO (PATCH PANNEL): conjunto de conectores en un mismo plano o ensamble usados para efectuar la terminación de los cables, facilitando la conexión de cruce y la administración de cableado.

PARADIGMA: forma de hacer, enfoque que se adopta para resolver problemas de software.

PORTABILIDAD: facilidad con la que puede transportarse el sistema hacia plataformas de hardware y software diferentes.

PLATAFORMA: conjunto de subsistemas y tecnologías que proporcionan un sistema coherente de funcionalidad a través de interfaces.

REQUERIMIENTO: función que debe tener el sistema (requerimiento funcional) o una restricción sobre el sistema (requisito no funcional).

SISTEMA: conjunto organizado de partes que se comunican para un propósito específico.

SOFTWARE: es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo.

SUBCLASE: clase especializada en una relación de generalización.

SUPERCLASE: clase general en una relación de generalización.

TELECOMUNICACIONES: tipo de tecnología que aporta servicios de voz, datos y video.

TELEMÁTICA: unión entre las telecomunicaciones y la informática.

USABILIDAD: facilidad con la que pueden interactuar los usuarios con el sistema.

ABREVIATURAS

CIDLIS. Centro de Innovación y Desarrollo para la investigación en Ingeniería del Software.

EA. Enterprise Architect.

LAN. Redes de área local (Local Area Network).

MOF. Meta Object Facility

MDA. Modelos Orientados por Arquitectura (Model Driven Architecture).

OMG. Grupo de Gestión de Objetos (Object Management Group)

PIM. Modelo Independiente de la Plataforma.

PSM. Modelo Específico de la Plataforma.

SCE. Sistema de Cableado Estructurado.

SE. Ingeniería del Software (Software Engineering)

TICs. Tecnologías de Información y Comunicación.

UML. Lenguaje de Modelado Unificado.

UTP. Par trenzado no apantallado (Unshielded Twisted Pair). Norma ISO 11801.

INTRODUCCIÓN

Uno de los principales objetivos de la ingeniería de requisitos es servir de base a clientes, usuarios y desarrolladores para alcanzar un acuerdo sobre las necesidades que debe satisfacer el sistema a desarrollar. Por este motivo, la especificación de requisitos es un elemento crucial para la adecuada ejecución de un proyecto de desarrollo de software y para la elaboración con los mejores estándares de calidad del producto final. A lo largo de la historia del desarrollo del software, se ha evolucionado desde los lenguajes ensambladores de los años 50 hasta los lenguajes orientados a objetos y de cuarta generación (4GLs) de nuestros días, de otra parte, la ingeniería del software es una disciplina relativamente reciente que ha tenido una evolución vertiginosa.

En el año 2001 el OMG¹ estableció el MDA² como arquitectura para el desarrollo de aplicaciones, representando un nuevo paradigma en el que los modelos guían todo el proceso de desarrollo software; teniendo como razón principal la obtención de beneficios importantes en aspectos fundamentales como son:

Productividad.

Portabilidad.

Interoperabilidad.

Mantenimiento.

Cabe resaltar que en la actualidad el rápido desarrollo de las tecnologías de información y Comunicación está afectando todos los aspectos de la sociedad,

¹ “OMG” (*OBJECT MANAGEMENT GROUP*). Asociación sin fines de lucro formada por grandes corporaciones, muchas de ellas de la industria del software, como IBM, Apple, Sun Microsystems y HP.

²“MDA” Modelos Orientados por Arquitectura. (Model Driven Architecture)

generando una demanda de productos, que desembocan en estos nuevos paradigmas.

De esta manera, el lector encontrará en este documento el desarrollo de una primera versión de un procedimiento para la especificación de requisitos de un sistema TIC sustentado en MDA, que satisface como herramienta pedagógica, los objetivos de este trabajo de grado.

En el capítulo 1 se hace una introducción a las Tecnologías de la Información y las Comunicaciones y a la ingeniería del software, reconociendo tanto el ciclo de vida de un producto como el ciclo de vida de desarrollo software, así mismo se reconoce la técnica de (programación/modelado) orientada(o) a objetos, usando el lenguaje unificado UML.

En el capítulo 2 se presenta un marco de referencia conceptual acerca de la ingeniería de requisitos como base fundamental para la elaboración del método que se propone para el desarrollo de este trabajo.

En el capítulo 3 se abarca la definición y además se revisa la interacción de MDA en el proceso de desarrollo de software y también se hace un reconocimiento de la herramienta de modelado utilizada.

Por último en el capítulo 4 se hace la especificación a partir de una situación que se aborda como ejemplo, implementando la metodología propuesta y mostrando las distintas fases del procedimiento.

Finalmente se exponen las conclusiones del trabajo y las recomendaciones para trabajos futuros, y se presentan en el anexo A y B, la propuesta para realizar la especificación de requisitos, y en el anexo C se describen los elementos básicos del lenguaje UML utilizados en este trabajo, soportados bajo el contexto de la herramienta Enterprise Architect.

Se espera que, en este trabajo, el lector encuentre un punto de partida que lo motive a investigar y desarrollar al respecto de los temas relacionados con la ingeniería de software y la ingeniería de requisitos.

ESTADO DEL ARTE

Hoy en día el uso de modelos es una práctica aceptada en las diferentes disciplinas de la ingeniería, mayor aún es su uso en la ingeniería del software ya que casi todas las metodologías que utiliza tienen como herramienta principal el enfoque dirigido por modelos.

El concepto de MDA fue presentado oficialmente en el 2001 y en la actualidad está fuertemente respaldada por muchas de las empresas líderes del sector informático como: Sun Microsystems, Microsoft, etc. Desde entonces se ha venido utilizando sustancialmente, es ampliamente reconocido en todo el mundo debido a su integración con las Tecnologías de Información y es promovida desde el ámbito científico-académico.

Una gran cantidad de empresas actualmente hacen uso de los modelos orientados por arquitectura ya que estos se enfocan en varios aspectos como son: recopilación de requisitos, análisis empresarial, proceso de modelado, diseño de sistemas, integración de sistemas, soluciones de diseño, plataformas de integración de código, transformaciones automáticas; proporcionando de esta manera muchas ventajas, y diferentes entornos y herramientas de desarrollo para la construcción de sistemas de calidad y resolviendo también problemas de negocio que se presenten en la vida real.

De igual modo se han realizado numerosos estudios para verificar las ventajas que ofrece, demostrando que con el uso de MDA se incrementa la productividad de la empresa u organización que acoge su implementación.

1. LAS TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TIC) Y LA INGENIERÍA DEL SOFTWARE

1.1. DEFINICIÓN DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN (TIC)

Las Tecnologías de Información y Comunicación (TIC) “*Conjunto convergente de tecnologías de la microelectrónica, la informática (hardware y software), las telecomunicaciones, la televisión y la radio*” [4], actualmente ejercen una enorme influencia en casi todos los ámbitos de nuestras sociedades, brindando nuevas oportunidades para el desarrollo como eje transversal en los sectores económicos y sociales.

La interoperabilidad de redes, productos y aplicaciones, que permita a los usuarios trabajar con cualquier otro sistema o servicio de información es un asunto que recobra importancia diariamente, en la actualidad el término TIC se utiliza para hacer referencia a una gama amplia de servicios, aplicaciones, y tecnologías, que utilizan diversos tipos de equipos y programas informáticos, e interactúan con usuarios y entre sí a través de las redes de telecomunicaciones, tienen como fin el mejoramiento de la calidad de vida de las personas dentro de un entorno, y su integración a un sistema de información interconectado y complementario.

Como concepto sociológico y no informático se refieren a saberes necesarios que hacen referencia a la utilización de múltiples medios informáticos para almacenar, procesar y difundir todo tipo de información, con diferentes finalidades como son la formación educativa, organización, gestión empresarial y la toma de decisiones en general.

“Las TIC se conciben como el universo de dos conjuntos, representados por las tradicionales Tecnologías de la Comunicación (TC), constituidas principalmente por la radio, la televisión y la telefonía convencional y por las Tecnologías de la Información (TI) caracterizadas por la digitalización de las tecnologías de registros de contenidos (informática, de las comunicaciones, telemática y de las interfaces).”[3]

Existen factores que hacen que la industria de desarrollo de TIC tenga un ritmo acelerado, algunos de ellos son consecuencia de la ley de Moore: *“Incremento constante de la capacidad de operación, miniaturización y reducción de costes para la producción de hardware”* [7] y a éstos se ha sumado en la última década el avance de las comunicaciones entre sistemas.

Estas líneas de avance han extendido el ámbito de aplicación del hardware y por ende del software, e incrementado al mismo ritmo exponencial la complejidad de los sistemas en los que se integran.

1.2. GESTIÓN DE PROYECTOS DE TIC

De acuerdo con la definición del PMI³, un proyecto es un esfuerzo temporal realizado para cumplir un propósito único y su gestión conlleva a la aplicación de conocimientos, habilidades, herramientas y técnicas en las diversas actividades que se realicen para cumplir o sobrepasar las necesidades y expectativas de los participantes.

En la tabla 1 se muestran algunos factores que inciden en el resultado de un proyecto para que este sea exitoso o un fracaso.

Tabla 1. Resultados en los proyectos TIC según Standish Group. Tomado de:[13].

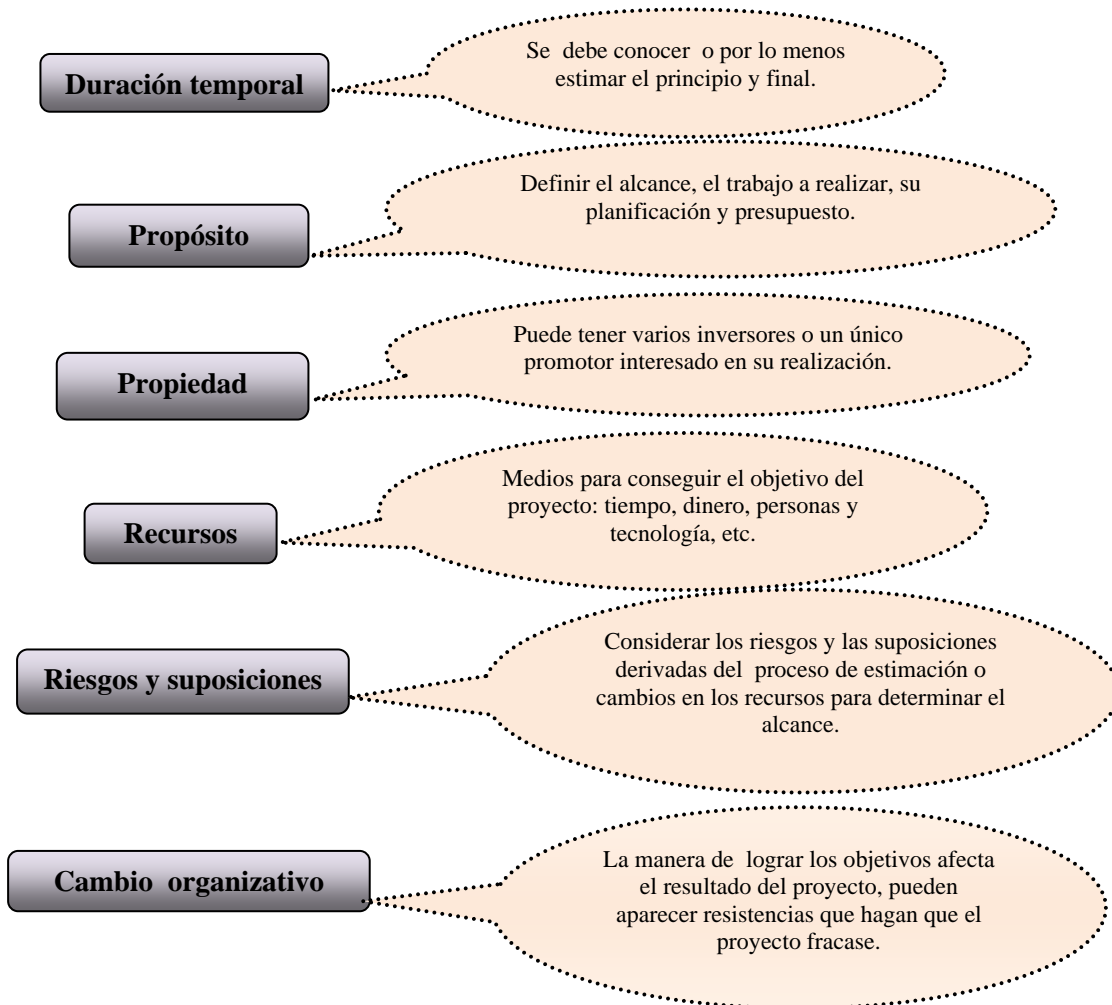
ÉXITO	FRACASO
Implicación del usuario	Requisitos incompletos
Apoyo directivo	Falta de aportación del usuario
Claridad en requisitos	Falta de recursos
Planificación adecuada	Expectativas no realista
Expectativas realistas	Falta de apoyo directivo
División en pequeños proyectos	Cambios en requisitos y especificaciones
Equipo competente	Falta de planificación
Propiedad clara del proyecto	No necesario
Visión y objetivos claros	Falta de gestión TIC
Equipo de alto rendimiento bien orientado	Problemas tecnológico

1.2.1. Atributos de un proyecto TIC

La importancia de conocer los atributos que rigen un proyecto de TIC, radica en el hecho de poder anticipar riesgos, de esta manera se puede obtener un buen

³ Project Management Institute. <http://www.pmi.org>

resultado. El siguiente esquema muestra las principales propiedades que debe poseer un proyecto.



Esquema 1. Atributos de un proyecto TIC. Fuente: Autora del trabajo.

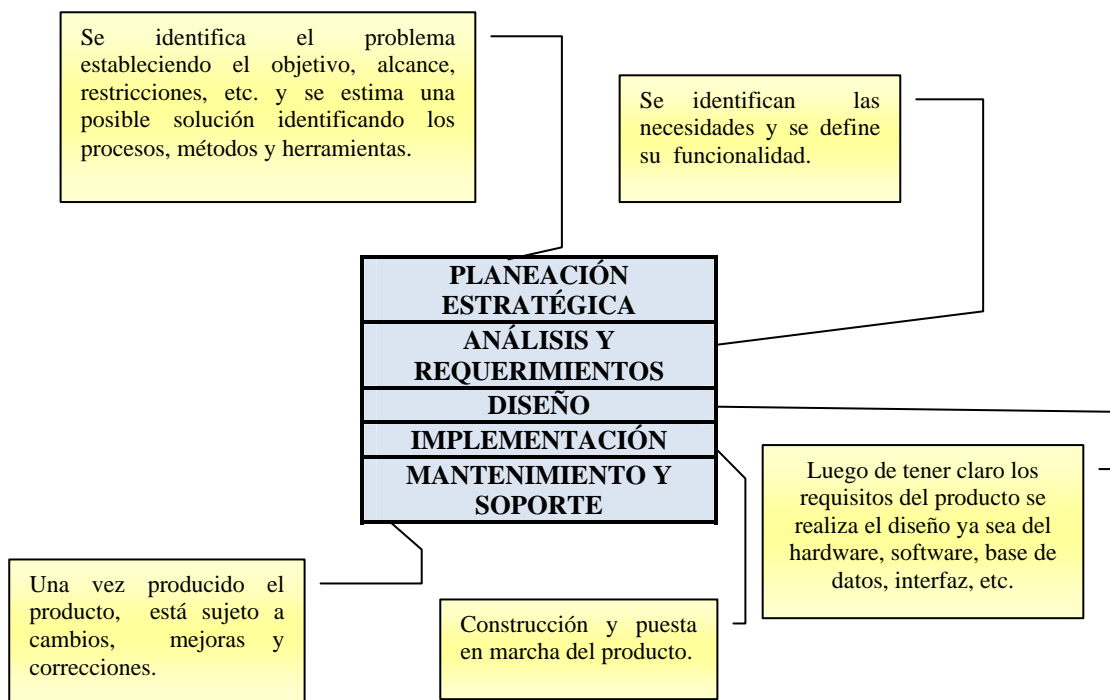
1.3. CICLO DE VIDA DE UN PRODUCTO TIC

El ciclo de vida de un producto está representado por todas las actividades que son necesarias para desarrollarlo y por todos los productos de trabajo que son producidos, su objetivo principal es establecer un marco común de desarrollo a

sabiendas que no todos los productos requieren de las mismas actividades y de la misma secuencia con que se llevan a cabo.

Para este trabajo de investigación se tuvieron en cuenta estas etapas del ciclo de vida de un producto de TIC. Sin embargo, se contemplaron acciones relacionadas desde la etapa de planificación hasta la etapa de diseño, dejando las etapas subsiguientes disponibles para la realización de un trabajo de grado posterior.

El siguiente esquema muestra una aproximación del ciclo de vida de un producto TIC.



Esquema 2. Ciclo de vida de un producto TIC. Fuente: Autora del trabajo.

1.4. INGENIERÍA DEL SOFTWARE

En este apartado se hace una descripción sobre los aspectos principales que soportan la solución de los objetivos planteados, teniendo en cuenta que *“la ingeniería del software ocurre como consecuencia de un proceso denominado ingeniería de sistemas que en lugar de concentrarse únicamente en el software, se concentra en una variedad de elementos, analizando, diseñando y organizando*

esos elementos en un sistema que puede ser un producto, un servicio o una tecnología para la transformación o control de información”[7].

También, la ingeniería del software es considerada como una actividad que usa modelos para la solución de problemas, en esta búsqueda de soluciones se adquiere conocimiento y fundamentación para tomar decisiones acerca del sistema. *La ingeniería de software es una actividad desde la perspectiva del modelado, para la solución de problemas, adquisición de conocimiento, dirigida por una fundamentación*[1]. La tabla 2 muestra algunas definiciones que a lo largo del tiempo se le han dado a la ingeniería del Software.

Tabla 2. Definición de la ingeniería del software. Fuente: Autora del trabajo.

AUTOR	AÑO	DEFINICIÓN
Fritz Bayer	1968	<i>“Establecimiento y uso de principios de ingeniería para obtener software económico que trabaje de forma eficiente en máquinas reales”. [7]</i>
IEEE	1993	<i>“Aplicación de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software. Disciplina para producir software de calidad desarrollado sobre las agendas y costes previstos, satisfaciendo los requisitos”. [7]</i>

1.4.1. Generalidades del proceso de desarrollo software

El proceso de desarrollo de software *“es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo. Concretamente define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo”*. [12]

Para un proceso de desarrollo de software se requiere un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le denomina el ciclo de vida del software⁴ que comprende cuatro grandes fases: concepción,

⁴ El ciclo de vida del software se explica con mayor profundidad en el inciso 1.5

elaboración, construcción y transición. En la concepción se especifica el alcance del proyecto, desarrollando un caso de estudio; la elaboración precisa un plan para la ejecución del proyecto, definiendo las características y determinando su arquitectura, etapa hasta la cual se enfocó este trabajo de grado; en la construcción se elabora el producto; por ultimo en la transición se realiza la transferencia del producto a los usuarios.

1.4.2. Estándares de Ingeniería de Software

A continuación se citan en la tabla 3, las entidades más conocidas internacionalmente por sus trabajos y esfuerzos realizados para la normalización, y reconocimiento de la ingeniería del software; estos son: ISO, IEEE- Computer Society y SEI.

Tabla 3. Estándares de Ingeniería de Software. Fuente: Autora del trabajo.

<p>ISO "Organización Internacional para la Estandarización"</p>	<p>Aportando los siguientes estándares para la Ingeniería del Software:</p> <p>ISO/IEC 12207. Define los procesos que intervienen en el desarrollo, mantenimiento y operación del software.</p> <p>ISO/IEC TR 15504. Extrae modelos para mejorar los procesos.</p>
<p>IEEE Computer Society "Instituto de Ingenieros en Electricidad y Electrónica"</p>	<p>IEEE ha desarrollado estándares para todas las áreas de Ingeniería del Software, algunos son:</p> <p>IEEE Std. 830. Recomendaciones para las especificaciones de software.</p> <p>IEEE Std. 1362. Guía para la descripción de un sistema, mediante el documento ConOps.</p> <p>IEEE Std. 1012. Estándar para la verificación y validación de software.</p> <p>IEEE Std. 1219. Estándar para el mantenimiento del software.</p>
<p>SEI "Instituto de Ingeniería del software"</p>	<p>Entre los aportes más significativos de este instituto en la producción de software se encuentran los modelos de madurez de las Capacidades: CMM y CMMI⁵.</p>

⁵ CMM - CMMI "Modelo de Capacidad y Madurez Integrado".

“Los estándares de ingeniería del software contribuyen la implementación de modelos asociados con el desarrollo, la operación y el mantenimiento de los sistemas software. Dichos estándares describen el tratamiento de necesidades en el desarrollo del software, no solo como requisitos y especificaciones, sino, articulando las necesidades formales en todos los ámbitos directivos”. [5]

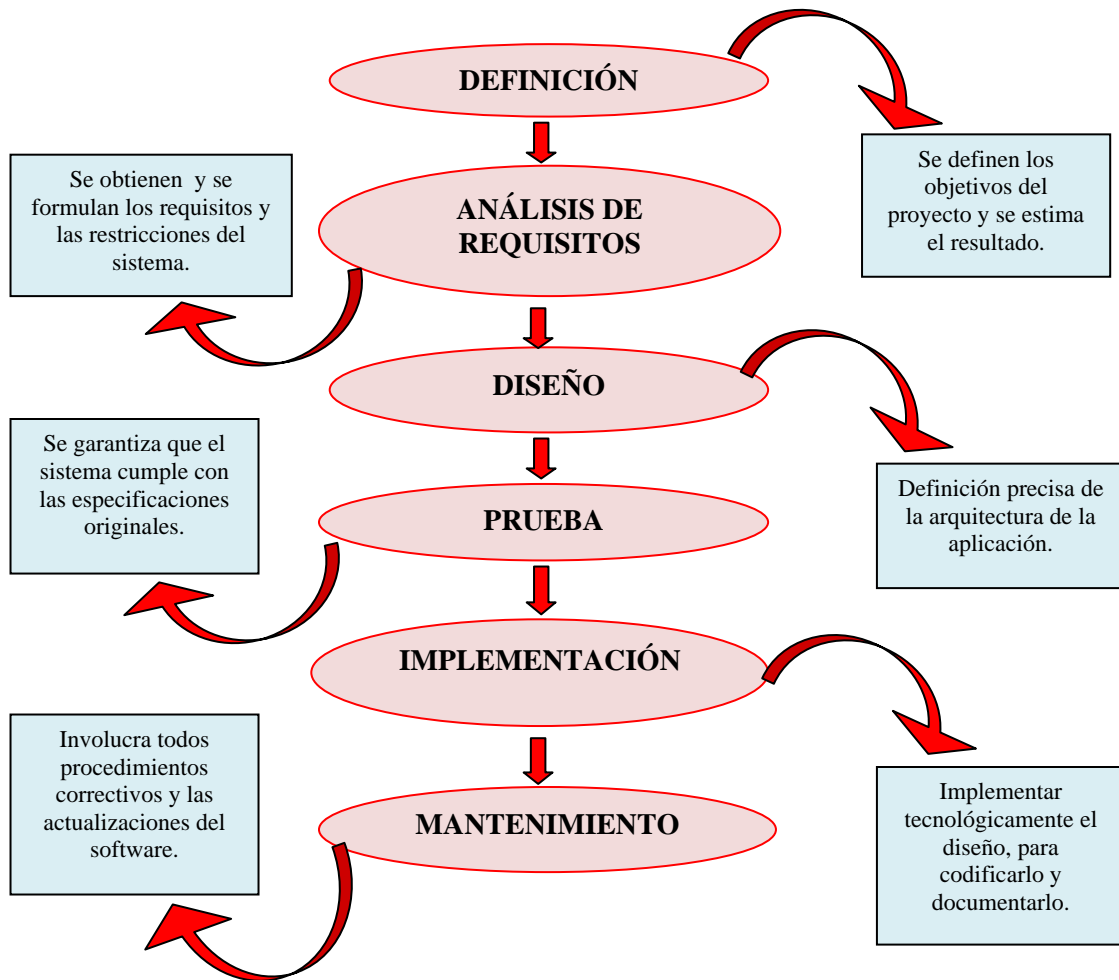
La utilidad de los estándares en el desarrollo de software, radica en la forma como estos agrupan lo mejor y lo más apropiado de las buenas prácticas de desarrollo, ya que engloban los conocimientos y proporcionan un marco para implementar procedimientos de aseguramiento de la calidad, proporcionan continuidad y entendimiento entre el trabajo de personas y las distintas organizaciones.

1.5. CICLO DE VIDA DEL SOFTWARE

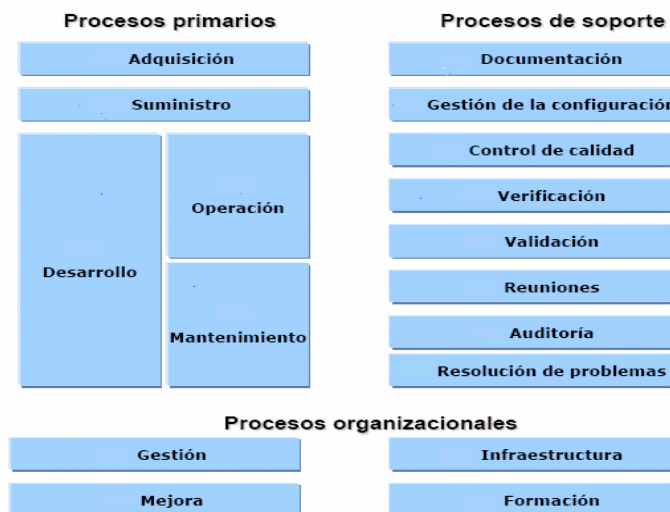
Se entiende por ciclo de vida los procesos y actividades que se realizan para construir, entregar y hacer evolucionar el software, desde que se concibe la idea hasta que se entrega y se retira el producto, estos procesos permiten detectar rápidamente los errores que se pueden presentar en cada etapa del desarrollo, esto a su vez permite concentrarse en la calidad del software, en los plazos de implementación y en los costos asociados.

El ciclo de vida básico del software consta de los procedimientos que se muestran en el esquema 3, sin embargo el orden de cada uno de estos procesos dependen de la aplicación.

El esquema 4 resume los procesos y las actividades que intervienen en el ciclo de vida del software, y en la tabla 4 se muestra una breve definición de estos procesos, junto con las actividades y las tareas que se realizan en cada una de ellas.



Esquema 3. Ciclo de vida del software. Fuente: Autora del trabajo.



Esquema 4. Procesos del ciclo de vida del software según ISO 12207. Tomado de: [13].

Tabla 4. Procesos en la Ingeniería del Software. Fuente: Autora del trabajo.

<p>Procesos Primarios Son aquellos que conceptualizan las ideas iniciales del producto.</p>	<p>Adquisición: Se adquiere el producto. Suministro: Proceso para proporcionar el producto. Desarrollo: Se diseña, construye y se le realizan pruebas al producto. Operación: Proceso que se sigue para el uso del producto. Mantenimiento: Proceso empleado para mantener el producto, cuando se realizan cambios internos y externos a él.</p>
<p>Procesos de Soporte Identificados como los procesos que dan soporte al producto desde un proceso primario, o incluso desde otro proceso de soporte.</p>	<p>Documentación: Actividades en las que se registra información. Gestión de la Configuración: Es mantener un registro de los productos generados en la ejecución de los procesos. Control de Calidad: Comprende las actividades que garantizan que el producto y los procesos asociados son conformes a los requisitos documentados y a las planificaciones. Verificación y Validación: Proceso que involucra las actividades para verificar y validar el producto. Reuniones de Revisión: Reuniones empleadas para evaluar el estado del producto y de las actividades. Auditorías: Actividades para determinar que el proyecto cumple con los requisitos, planes y contratos. Resolución de Problemas: Actividades para analizar y resolver problemas.</p>
<p>Procesos Organizacionales Son los procesos que deben realizarse en el contexto de la organización que va a ejecutar el proyecto.</p>	<p>Gestión: Describe las actividades de gestión de la organización, y la gestión de proyectos. Infraestructura: Incluye entre otros el capital y el personal involucrados. Mejora: Actividades realizadas para mejorar la capacidad de todos los procesos. Formación: Representa las actividades realizadas para identificar e institucionalizar los procesos.</p>

Desde el punto de vista del estándar ISO/IEC 12207, el marco del ciclo de vida del software cubre desde la conceptualización de las ideas iniciales del producto hasta el fin de su uso (retirada). *“Este estándar se encarga de definir los procesos que intervienen en el desarrollo, mantenimiento y operación del software, define el qué, no el cómo; asentando un marco estándar de referencia internacional, pero no se ocupa ni prescribe técnicas específicas; definiendo un proceso como un conjunto de actividades y tareas relacionadas, que al ejecutarse de forma conjunta transforman una entrada en una salida”.* [13]

1.6. ANÁLISIS ORIENTADO A OBJETOS

El análisis orientado a objetos es una técnica que fomenta una metodología basada en el proceso unificado para el desarrollo de software, utiliza como base la notación del lenguaje de modelado unificado (UML)⁶, su propósito es desarrollar software que modele un esquema del mundo real, generando como primera medida un sistema mediante un conjunto de objetos (datos abstractos), luego agregando funcionalidad a los componentes amplía el sistema, para que cuando se desee crear uno nuevo se puedan volver a utilizar los objetos generados y reducir con esto el tiempo de desarrollo, entre más atributos y acciones se tengan en cuenta, en mayor medida se asemeja el modelo con la realidad.

“La construcción de software orientado a objetos es el método de desarrollo de software que basa la arquitectura de cualquier sistema software en módulos deducidos de los tipos de objetos que manipulan (en lugar de basarse en la función o funciones a las que el sistema está destinado asegurar)”.[8]

El enfoque orientado a objetos abarca todo el ciclo de vida del desarrollo del software, los casos de usos y las clases son la base de dicho enfoque.

Casos de usos: los casos de uso hacen referencia a la funcionalidad de un sistema desde el punto de vista del usuario y representan las interacciones entre el usuario y el sistema.

Clases: Es la representación de una categoría o grupo de cosas que tienen atributos y acciones similares dentro del sistema.

⁶ En el capítulo 3 se encuentra la explicación de este lenguaje de modelado y en el anexo C se describen los principales elementos utilizados en este trabajo, se recomienda analizar este tema para lograr una mayor comprensión del presente capítulo.

2. LA INGENIERÍA DE REQUISITOS

En el apartado anterior se mostraron las etapas del proceso en la ingeniería de software y las numerosas tareas que se requieren llevar a cabo, se destaca el análisis de requisitos.

Cuando se realiza el análisis de requisitos se identifican las necesidades y las restricciones sobre las cuales debe operar el sistema, en un proceso que comienza con el análisis del problema y termina con la documentación de los resultados, y que resulta complejo porque se deben identificar todos los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes; la importancia de esta fase radica en el hecho de que los resultados del producto final van a depender en gran medida de este proceso.

De esta manera se plantea un procedimiento general para la especificación de requisitos que se utilizará para sistemas tic, el cual será implementado tomando como ejemplo de aplicación un caso de estudio denominado, “Diseño del cableado estructurado para comunicación de voz, datos y video de un edificio”

Al finalizar el capítulo se propone una plantilla para la especificación de requisitos de sistemas TIC, utilizando como soporte las guías [15] y [16].

2.1. IMPORTANCIA DE LOS REQUISITOS

“Para que un esfuerzo de desarrollo de software tenga éxito, es esencial comprender perfectamente los requisitos del software. Independientemente de lo bien diseñado o codificado que esté un programa, si se ha analizado y especificado pobremente, decepcionará al usuario y desprestigiará al que lo ha desarrollado”. [7]

Los requisitos son de vital importancia a la hora de realizar un proyecto, puesto que los resultados van a depender en gran medida de estos, por tal motivo si se cometen errores en los requisitos estos van a repercutir en las demás fases del proyecto.

En la tabla 5 se mencionan los errores más comunes en las distintas fases relacionadas con la especificación de requisitos y sus consecuencias:

Tabla 5. Errores en la obtención de los requisitos y sus consecuencias. Fuente: Autora del trabajo.

ERROR	CONSECUENCIA
Implicación insuficiente del cliente	Los resultados no se pueden garantizar, ya que no se logra conocer a fondo tanto el problema como todas las funcionalidades del sistema.
Requisitos crecientes y cambiantes	Los cambios en los requisitos pueden incrementar o modificar funcionalidades ya implementadas, además se exceden los costos y la planificación del proyecto.
Requisitos incompletos	Se incrementan las modificaciones del proyecto durante el desarrollo, generando además de problemas de programación y degradación de la arquitectura, un producto con serias deficiencias técnicas
Requisitos ambiguos	La ambigüedad crea funcionalidades que no se ajustan a lo que los usuarios necesitan, puesto que los requisitos se interpretan de diferentes maneras.
Requisitos innecesarios	Se crean funcionalidades que son programadas pero que no son utilizadas, adicionando un costo de desarrollo innecesario.
Requisitos insuficientes (mínimos)	El sistema no satisface a cabalidad todas las necesidades requeridas.

2.2. CLASIFICACIÓN DE LOS REQUISITOS

Los requisitos se pueden clasificar de diversas maneras, para el desarrollo de este trabajo que considera la siguiente clasificación:

2.2.1. Requisitos Funcionales

Los requisitos funcionales definen el comportamiento de un sistema, así como las tareas que debe realizar. Una funcionalidad generalmente se expresa como una declaración en forma verbal, es por ello que según el tipo de sistema que se

desarrolle, los requisitos funcionales se representan en un documento o en un caso de uso.

2.2.2. Requisitos No Funcionales

Definen aspectos que puedan limitar el sistema, pero que resultan deseables desde el punto de vista del usuario. Generalmente comprenden atributos de calidad: tiempos de respuesta, características de usabilidad, facilidad de mantenimiento, lenguaje de programación o plataforma tecnológica, etc. En la tabla 6 se clasifican estos requisitos.

Tabla 6. Requisitos no funcionales. Fuente: Autora del trabajo.

USABILIDAD	Necesidades asociadas al esfuerzo y acciones necesarias para aprender, operar, preparar entradas e interpretar la salida de un sistema.
CONFIABILIDAD	Acciones del sistema que permiten que realice las funciones para las que fue diseñado sin presentar fallos, puede mencionar características como la disponibilidad, el porcentaje de fallas máximo, etc.
SEGURIDAD	Incluye especialmente los temas asociados a que el acceso al sistema sea controlado, que existan restricciones en el acceso a la información, las formas de autorización de ese acceso y la identificación de información que debe ser salvaguardada.
EFICIENCIA	Recursos que requiere el sistema para realizar una función, tienen relación con: el tiempo de respuesta para una transacción y la capacidad, rendimiento del procesamiento.
MANTENIMIENTO Y ACTUALIZACIÓN	Capacidad u habilidad que se tiene para entregar nuevas versiones del producto a los clientes con un mínimo de tiempo de descarga y costo.
SOPORTABILIDAD Y OPERABILIDAD	Hacen referencia a la habilidad de proveer soporte técnico eficiente y a la habilidad de Hospedar y operar el software.

2.2.3. Requisitos de interfaz

En esta clasificación de requisitos que puede verse en la tabla 7 se definen las interacciones del sistema con su entorno.

Tabla 7. Requisitos de interfaz. Fuente: Autora del trabajo.

INTERFACES DE USUARIO	Generalmente se refieren al esquema general de las interfaces gráficas para los roles más importantes del sistema.
INTERFACES DE SOFTWARE	Son las interfaces de software entre componentes o unidades de mayor nivel que se encuentren identificadas en el sistema.
INTERFACES DE HARDWARE	Cualquier interfaz de hardware que debe ser apoyada o enlazada con el software, y que forma parte del conjunto de uso del sistema o que apoya el cumplimiento de requisitos funcionales.
INTERFACES DE COMUNICACIONES	Estas interfaces permiten las comunicaciones a otros sistemas o dispositivos que se comportan como sistemas completos que deben establecerse para dar cumplimiento a requisitos funcionales del sistema.

2.3. PROCESOS EN LA ESPECIFICACIÓN DE REQUISITOS

En el proceso de desarrollo de un sistema, el equipo de desarrollo se enfrenta al problema de la identificación de requisitos, para realizar este proceso, actualmente no existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado. Existe en cambio un conjunto de técnicas, cuyo uso proponen las diferentes metodologías para el desarrollo de aplicaciones. Se debe tener en cuenta que la selección de las técnicas y el éxito de los resultados que se obtengan, depende en gran medida tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participen.

El proceso de especificación de requisitos se puede dividir en cinco grandes actividades tal y como lo muestra el esquema 5. El cumplimiento de los objetivos de este trabajo de grado se ha enfatizado en los tres primeros procesos.

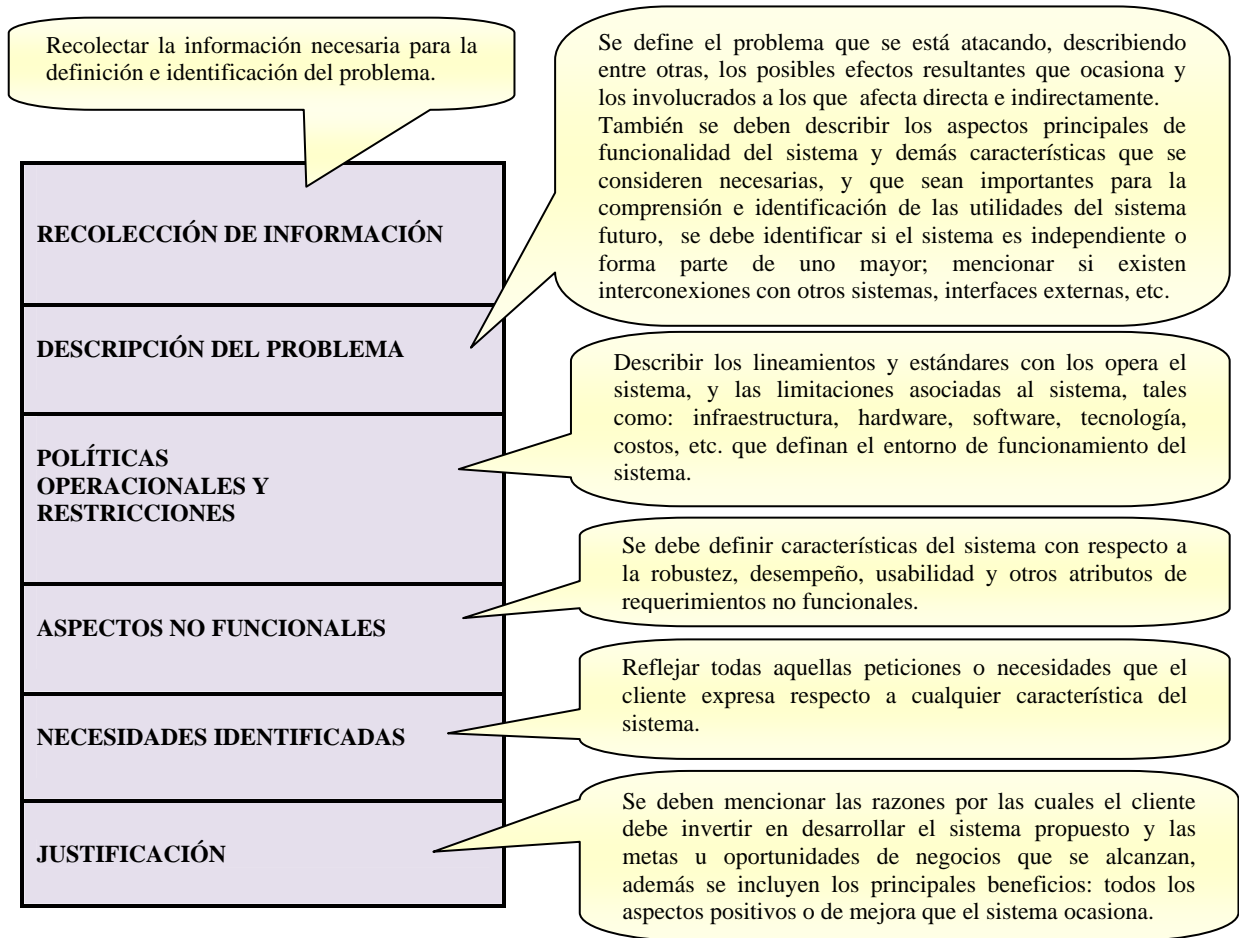


Esquema 5. Áreas de proceso en la Ingeniería de Requisitos. Tomado de:[13]

2.3.1. Visión del Sistema

Es un documento que va dirigido a los usuarios y describe las características del sistema propuesto resumiendo los retos y oportunidades a enfrentar en el proyecto, la visión del sistema es el medio de comunicación que recoge la visión general, cualitativa y cuantitativa de las características del sistema; compartido por la parte cliente y desarrolladora y esta normalizado en el estándar IEEE Std. 1362-1998. [17]

La “Visión del Sistema” se debe escribir de manera clara y resumida, para que pueda ser transmitida y comprendida fácilmente por los involucrados en el proyecto, incluirá descripciones principales del sistema y las necesidades del cliente, evitando ir demasiado al detalle o al uso de lenguaje técnico, así como también debe contener las principales características que se requieren en los productos finales del mismo. A continuación y mediante el esquema 6, se realiza una síntesis del contenido del documento “Visión del Sistema” soportado a través de la guía y la plantilla del CIDLIS. [22], [23].



Esquema 6. Síntesis del documento “Visión del sistema” del Sistema de Gestión de Calidad del CIDLIS. Fuente: Autora del trabajo.

Cabe resaltar que el orden en que se realiza esta actividad no necesariamente se debe efectuar como se mostró en el esquema 6, este va a depender tanto del sistema que se estudia como del equipo de desarrollo, sin embargo para mantener una estructura definida y organizada en el desarrollo de este trabajo se siguió este orden de ideas.

2.3.2. Obtención y Análisis

El primer paso consiste en conocer y comprender las necesidades y problemas del cliente; la captura de requisitos (elicitación) es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae de cualquier fuente de

información disponible, las necesidades que debe cubrir dicho sistema. A continuación se presenta un grupo de técnicas que de forma clásica han sido utilizadas para esta actividad en el proceso de desarrollo de todo tipo de software, permitiendo hacer este proceso más eficiente y preciso.

- ✓ **Entrevistas** [13]: Es una técnica que permite conocer el problema de una forma natural, también permite comprender los requerimientos y las prioridades del cliente, así como los objetivos de la solución buscada. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización y documentación de los resultados.
- ✓ **JAD (Joint Application Development/Desarrollo conjunto de aplicaciones)** [13]: Es una técnica de obtención de requerimientos en la que participan clientes, usuarios y desarrolladores, mediante reuniones que se realizan durante varios días. Se basa principalmente en la dinámica de grupo y el uso de ayudas visuales. El objetivo de esta actividad es mantener un proceso organizado, racional y una filosofía de documentación.
- ✓ **Cuestionarios y Checklists** [13]: Con esta técnica se obtiene información de una o más personas en forma estructurada, mediante la redacción de documentos con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist), además se requiere que el analista conozca el ámbito del problema en el que está trabajando.
- ✓ **Casos de Uso** [13]: Los casos de uso describen los actores y el alcance del sistema, ya que estos representan la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función, la ventaja fundamental de los casos de uso es que el cliente o usuario los puede entender fácilmente, se denotan mediante los

elementos de UML, los cuales se explican con profundidad en el capítulo 3 del presente trabajo.

- ✓ **Prototipos**[13]: Esta técnica es muy útil para elicitar requisitos del cliente y para identificar y eliminar partes arriesgadas de un proyecto ya que permite ensayar posibles soluciones mediante una comprensión más profunda del problema logrando así ahorrar trabajos futuros de corrección.

Una vez obtenida la información necesaria del entorno, es necesario sintetizarla, darle prioridades, analizar posibles contradicciones o conflictos, descomponer el sistema y distribuir las necesidades de cada parte, delimitar los límites del sistema y definir su interacción con el entorno.

2.3.3. Especificación

Para realizar la especificación de requisitos, primero se debe formalizar la descripción del sistema mediante el documento “Visión del Sistema”, el cual es el primer paso a tener en cuenta en la construcción de nuestra aplicación. Cuando ya se conoce el entorno del cliente y sus necesidades, es necesario plasmarlas en forma de requisitos en los documentos que sirven de base de entendimiento y acuerdo entre cliente y desarrollador, y que establecerán tanto la guía de desarrollo como los criterios de validación del producto final. Documentar los requisitos es la condición más importante para gestionarlos correctamente.

Especificación de Requisitos del Sistema (SyRS)

La especificación de requisitos del sistema (SyRS) es un documento en el que se comunican los requisitos del cliente⁷ y las características de un sistema desde el

⁷ Es un término colectivo que puede incluir el cliente del sistema propuesto, el organismo de financiación, el aceptante que firmará la entrega, y los administradores que se encargarán de la supervisión, ejecución, operación y mantenimiento del sistema.

punto de vista del usuario a la comunidad técnica⁸ que especifica y construye el sistema, este documento está basado y es conforme con el estándar IEEE Std 1233-1998. [15]. La recopilación de las necesidades que constituye la especificación y su representación actúa como puente entre los dos grupos y debe ser comprensible por ambos.

El desarrollo de un SyRS incluye la identificación, organización, presentación y modificación de los requisitos. Según la guía de especificación de requisitos del sistema (Std 1233-1998) la colección de los requisitos debe tener las siguientes propiedades⁹: [15]

- Se debe postular solo una vez cada requisito.
- Los requisitos no deben superponerse, es decir, no deben referirse a otro requisito o a las capacidades de otros requisitos.
- Un SyRS debe incluir todos los requisitos identificados por el cliente, así como los necesarios para la definición del sistema.
- El contenido del SyRS debe ser coherente, claro y no contradictorio, y la presentación del material debe ser modificable.
- Deben ser identificados los límites, el alcance y el contexto para el conjunto de requisitos.
- Las versiones del SyRS deben mantenerse a través del tiempo.
- Este debe ser el nivel de abstracción para el sistema que se está definiendo.

Un documento SyRS escrito correctamente ofrece los siguientes beneficios [15]¹⁰:

- Garantía de que el cliente y la comunidad técnica comprenden las necesidades del sistema.

⁸ La comunidad técnica incluye los analistas, los estimadores, diseñadores, desarrolladores, ingenieros, probadores, mantenedores y fabricantes.

⁹ Tomado de la fuente referenciada en [15] y traducido por la autora del trabajo de grado.

¹⁰ Ídem.

- Un método para identificar y corregir problemas y malentendidos entre el cliente y la comunidad técnica.
- Un sistema base para establecer que el sistema cumple con las necesidades del cliente.
- Apoyo para el desarrollo de la planificación y diseño del sistema.
- La ayuda en la evaluación de los efectos de cambios inevitables.

Especificación de Requisitos de Software (ESRO)

El documento de especificación de requisitos software describe las funciones que realiza un determinado producto de software, programa o conjunto de programas en un determinado entorno, en este documento los requisitos del sistema son traducidos en requisitos de producto mediante un lenguaje más técnico. Los aspectos básicos que una descripción de requisitos debe cubrir son [16]¹¹:

- **Funcionalidad.** Descripción de lo que el software debe hacer.
- **Interfaces externos.** Cómo debe interactuar el software con las personas, el sistema de hardware, o con otros sistemas (software y hardware).
- **Rendimiento.** Indicación de la velocidad, disponibilidad, tiempos de respuesta, tiempos de recuperación, tiempos de determinadas funciones.
- **Atributos.** Consideraciones de portabilidad, corrección, mantenibilidad, seguridad, etc.
- **Restricciones de diseño en la implementación.** Indicación de las restricciones que puedan afectar al producto software, ya sean estándares, lenguajes, políticas de integridad de bases de datos, límites de recursos disponibles para el desarrollo, sistema operativo, etc. Las especificaciones de requisitos de software deben evitar incluir requisitos de diseño o de proyecto.

¹¹ Tomado de la fuente referenciada en [16] y traducido por la autora del trabajo de grado.

2.3.4. Verificación y validación

El esfuerzo de verificación y validación debe ajustarse a las características del proyecto, en esta fase se analizan las propiedades de calidad del documento que se crea para la especificación, comprobando que sea completo, correcto, consistente, modificable, trazable, etc.; también se chequean las propiedades de calidad de los requisitos, analizando que sean posibles, necesarios, priorizados, técnicamente correctos, verificables, etc.

2.3.5. Gestión de configuración

La gestión de la configuración del software es una disciplina formal de la ingeniería del software que proporciona métodos y herramientas para identificar y controlar el software durante su desarrollo y posterior uso; debido a que los requisitos probablemente cambiarán durante el desarrollo del sistema, se hace necesario poder trazar en cada cambio todas las partes afectadas, así como lograr medir el impacto que cada modificación implica en la planificación del proyecto.

Esta etapa comprende las siguientes actividades:

- Identificación y establecimiento de las “líneas base”¹².
- Revisión, aprobación o rechazo, control y seguimiento de los cambios.
- Auditorías y revisiones de la evolución de los productos de software.
- Control de la relación del sistema de software en su interfaz de operación.

¹² “Especificación de un producto que ha sido formalmente revisado y aceptado para servir como punto de referencia para su posterior desarrollo, y sólo puede modificarse a través de un procedimiento formal de control de cambio”.[10]

Independientemente de las técnicas o procesos que se apliquen para realizar las diferentes tareas relacionadas con el área de requisitos, son cinco los objetivos que hay que cubrir:



Esquema 7. Objetivos del proceso de la ingeniería de requisitos. Fuente: Autora del trabajo.

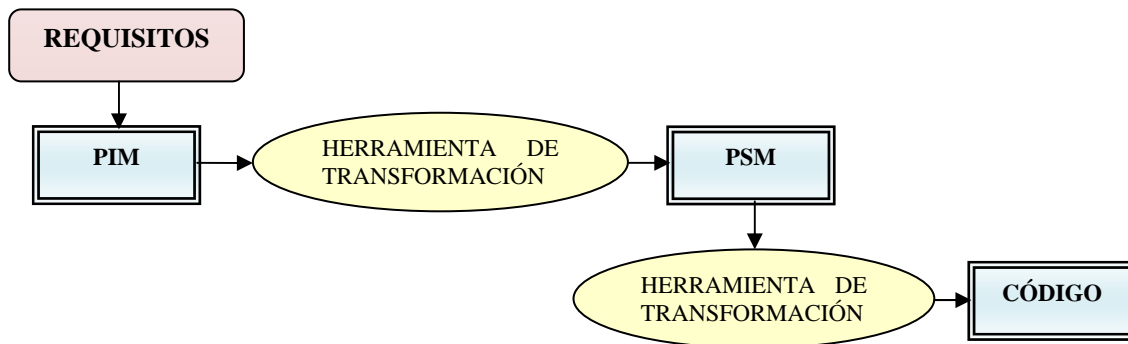
3. ARQUITECTURA DIRIGIDA POR MODELOS (MDA)

3.1. CONTEXTO

El Object Management Group (OMG) es un grupo de empresas informáticas que se creó en 1990 con el fin de fomentar e incrementar el desarrollo de aplicaciones orientadas a objetos. Este grupo ha definido numerosas especificaciones y estándares para la integración de sistemas software, en el año 2001 estableció el MDA como arquitectura para el desarrollo de aplicaciones representando un nuevo paradigma de desarrollo de software en el que los modelos guían todo el proceso de desarrollo, este nuevo paradigma se conoce como Ingeniería de modelos o Desarrollo basado en modelos.

Es evidente que durante el ciclo de vida de una aplicación, este puede sufrir diversos cambios, el hecho de que MDA esté basada en modelos especificados en UML¹³ permite tener un modelo estable de la aplicación, que lo independiza en gran medida de los cambios tecnológicos, de esta forma no sólo se disminuyen los costos a mediano y largo plazo, sino que se disfruta de un entorno de desarrollo y prueba más estable y coherente. Para conseguir estos beneficios, MDA plantea el proceso de desarrollo que se muestra en el esquema 8.

¹³ UML (*Lenguaje de Modelado Unificado*).



Esquema 8. Proceso de desarrollo de software con MDA. Fuente: Autora del trabajo.

Por tanto, MDA incorpora la posibilidad de efectuar las transformaciones entre el modelo independiente de la plataforma (PIM), a modelos específicos de la plataforma (PSM), y de este al modelo de código, por lo que se necesita una herramienta para automatizar esta tarea, en este trabajo se utilizó la herramienta Enterprise Architect¹⁴, la cual se explicará al finalizar este capítulo y se hace énfasis en obtener el modelo independiente de la plataforma (PIM) el cual enmarca el propósito de esta investigación.

3.2. INTRODUCCIÓN A MDA

La Arquitectura Dirigida por Modelos (MDA) es una nueva forma de construir software en la que se usan modelos del sistema a distintos niveles de abstracción por medio del lenguaje de modelado, sus principales objetivos son: mejorar la productividad, la portabilidad, la interoperabilidad y la reutilización de los sistemas.

MDA separa la especificación de las operaciones y los datos de un sistema (requerimientos del negocio), de la especificación de los detalles de la plataforma en la que el sistema será implementado (requerimientos tecnológicos),

¹⁴ Enterprise Architect es una herramienta CASE (Computer Aided Software Engineering) para el diseño y construcción de sistemas de software que soporta la especificación de UML 2.0.

permitiendo que ambos aspectos evolucionen individualmente sin generar dependencias entre sí y además que la lógica de negocio responda a las necesidades del negocio y no dependa de las variaciones técnicas.

Para ello, MDA proporciona las bases para:

- *“Definir un sistema independientemente de la plataforma sobre la que se construye.*
 - *Definir plataformas sobre las que construir los sistemas.*
 - *Elegir una plataforma particular para el sistema.*
 - *Transformar la especificación inicial del sistema a la plataforma elegida”.*
- [28]

A grandes rasgos, el proceso de desarrollo de software con MDA se puede dividir en tres fases¹⁵:

- Construcción de un modelo de alto nivel del sistema independiente de cualquier tecnología o plataforma denominado “Modelo Independiente de la Plataforma” (PIM).
- Transformación del PIM a uno o varios “Modelos Específicos de Plataforma” (PSM), estos modelos son de más bajo nivel que el PIM y describen el sistema de acuerdo con una tecnología de implementación determinada.
- Por último se genera el código a partir de cada PSM.

¹⁵ En la guía oficial de MDA [25] se habla también de un Modelo Independiente de Computación (CIM), como modelo previo al PIM que describe el sistema del negocio. No obstante, este modelo es opcional y para este trabajo no se tendrá en cuenta en el proceso de desarrollo con MDA.

3.3. MODELOS EN MDA

Un modelo es una descripción de todo o parte de un sistema escrito en un lenguaje bien definido, UML¹⁶ es el lenguaje de modelado que se ha adoptado como el principal lenguaje de MDA; entre los distintos tipos de modelos definidos en UML, los modelos de clases, que muestran la vista estática del sistema, son los más importantes ya que el PIM y la mayoría de PSMs son modelos de este tipo. Estos modelos se representan mediante diagramas de clases de UML; estos conceptos son explicados en el apartado de modelado del presente capítulo.

3.3.1. Modelo Independiente de la Plataforma (PIM)

“Un Modelo Independiente de la Plataforma o PIM es un modelo del sistema de alto nivel que representa la estructura, funcionalidad y restricciones del sistema sin aludir a una plataforma determinada, sirve de base para todo el proceso de desarrollo software”[28]. El PIM se construye cuando se quiere representar el modelo de procesos de negocio o modelo de dominio de un sistema, mediante la implementación de diagramas de clases.

Es independiente de la plataforma porque no incluye detalles específicos de una tecnología determinada, es decir, ignora los sistemas operativos, los lenguajes de programación, el hardware y la topología de red, este modelo facilita la creación de diferentes implementaciones del sistema en diferentes plataformas, dejando intacta su estructura y su funcionalidad básica; es fácilmente comprensible por los usuarios del sistema.

¹⁶ Ver apartado 3.4.2

3.3.2. Modelo Específico de la Plataforma (PSM)

“Un Modelo Específico de la Plataforma o PSM es un modelo del sistema con detalles específicos de la plataforma en la que será implementado. Se genera a partir del PIM, así que representa el mismo sistema pero a distinto nivel de abstracción” [28]. A partir de un mismo PIM pueden generarse varios PSMs, cada PSM describe el sistema desde una perspectiva diferente, a diferencia del PIM el PSM incluye detalles específicos de una tecnología determinada como por ejemplo lenguajes de programación, sistema operativo, entre otros.

En este trabajo se utilizó la herramienta Enterprise Architect para realizar las transformaciones de PIM a PSM, cabe resaltar que no se enfatizó mucho en este tema puesto que lo que importa para el cumplimiento de los objetivos de este trabajo es la construcción del PIM a partir de la especificación de requisitos.

3.3.3. Modelo de Código

Luego de que se tiene el PSM, se pasa al modelo de código el cual representa el código desplegable, normalmente en un lenguaje de programación de alto nivel, como Java, C#, C++, Visual Basic, JSP, etc. Idealmente, el modelo de código está listo para compilar, el despliegue de su aplicación es automatizado, pero como se ha mencionado con anterioridad para fines de este trabajo no se hace énfasis en este concepto, dejándolo más bien como punto de partida para trabajos futuros.

3.4. ESTÁNDARES EN MDA

3.4.1 MOF

El Meta Object Facility (MOF), es un estándar que define un lenguaje común y abstracto para definir lenguajes de modelado, y es uno de los más usados en MDA. Usa las cinco construcciones básicas que se muestran en la tabla 8:

Tabla 8. Construcciones básicas de lenguaje de modelado de MOF. Fuente: Autora del proyecto.

Clases	Definen todos los tipos de elementos en un lenguaje de modelado y representan todas las dependencias que pueden crearse en un modelo UML.
Atributos	Definen las propiedades de los elementos del modelo. Los atributos tienen un tipo y una multiplicidad.
Generalización	Define herencia entre clases. La subclase hereda todas las características de la clase padre.
Asociaciones	Definen relaciones entre clases.
Operaciones	Definen operaciones dentro del ámbito de una clase, junto con una lista de parámetros.

3.4.2 El Lenguaje Unificado de Modelado (UML)

UML define las reglas y la notación para especificar sistemas de software, por medio de un conjunto rico de elementos gráficos para el modelado orientado a objetos. *“UML se usa para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real, combina notaciones provenientes desde: Modelado Orientado a Objetos, Modelado de Datos, Modelado de Componentes, Modelado de Flujos de Trabajo”*[31]. Este apartado provee una breve introducción a los diagramas de modelado de UML 2.0¹⁷, soportados en la plataforma de modelado de EA (Enterprise Architect).

Para su mayor comprensión en el anexo C, se presenta un resumen con los elementos y conectores utilizados en este trabajo.

Diagramas de UML

Los conceptos que se explican a continuación son basados en la guía de usuario de Enterprise Architect de Sparx Systems¹⁸ [32] y sus elementos se profundizan en el anexo C.

UML define una serie de diagramas, divididos en dos grupos generales:

¹⁷UML 2.0 es la versión actual del modelado de objetos, en el transcurso del libro se cita UML haciendo referencia a dicha versión.

¹⁸ Sparx Systems es una empresa con sede en Australia con una sólida historia de innovación y desarrollo en el modelado UML, es una contribución de los miembros del Grupo de Gestión de objetos (OMG).

✓ Diagramas de modelado estructural

Los diagramas estructurales definen la arquitectura estática de un modelo. Ellos se usan para modelar los elementos (clases, objetos, interfaces) que constituyen un modelo. Además se usan para modelar las relaciones y dependencias entre elementos.

Tabla 9. Diagramas de modelado estructural de UML. Fuente: Autora del trabajo.

Diagramas de Clase	Los diagramas de clase representan la estructura y el comportamiento estático de un sistema por medio de la representación de uno o más objetos, incluye atributos (datos) y métodos (operaciones o comportamiento).
Diagramas de Objetos	Se encuentran estrechamente relacionados con los diagramas de clase, solo que estos representan los objetos y sus relaciones en tiempo de ejecución.
Diagramas de Componente	Modelan estructuras más complejas o de más alto nivel a partir de los diagramas de objetos, proveen una interfaz que muestra las organizaciones y dependencias entre los componentes del sistema.
Diagramas de Estructura Compuesta	Un diagrama de estructura compuesta son similares a los de clase, pero estos modelan un uso específico de la estructura, reflejando la colaboración interna de clases, interfaces o componentes en tiempo de ejecución
Diagramas de Despliegue	Muestran cómo y dónde se desplegará el sistema.
Diagramas de Paquete	Dividen el modelo en paquetes, describiendo las interacciones entre ellos a un nivel más alto.

✓ Diagramas de modelado de comportamiento

Los diagramas de comportamiento capturan las variedades de interacción y el estado instantáneo dentro de un modelo mientras se ejecuta a través del tiempo.

Tabla 10. Diagramas de modelado de comportamiento de UML. Fuente: autora del trabajo.

Diagramas de Tiempo	Proporcionan una representación visual del comportamiento de los diferentes objetos del modelo con una escala de tiempo.
Diagramas de Secuencia	Un diagrama de secuencia es una representación estructurada del comportamiento de los elementos de un sistema, en una serie de pasos secuenciales a lo largo del tiempo.
Diagramas de Comunicación	Estos diagramas muestran la comunicación e interacción entre los elementos de un sistema en tiempo de ejecución, son parecidos a los diagramas de secuencia, solo que estos se usan para visualizar relaciones inter-objetos.
Diagramas de Descripción de la Interacción	Los diagramas de descripción de la interacción reflejan el flujo de control de las cooperaciones entre otros diagramas de interacción.
Diagramas de Actividad	Se usan para modelar el comportamiento de un sistema, y la manera en que se relacionan estos comportamientos.
Diagramas de Máquina de Estado	Los diagramas de máquina de estados proporcionan el entendimiento de un modelo cuando este se ejecuta.
Diagramas de Caso de Uso	Los diagramas de Caso de Uso se usan para modelar las interacciones que existen entre el sistema y los usuarios, y también describen los requisitos funcionales del sistema.

3.5. LA HERRAMIENTA ENTERPRISE ARCHITECT

Enterprise Architect (EA) es una herramienta CASE¹⁹ basada en UML que abarca todas las fases del desarrollo software (levantamiento, análisis, diseño, desarrollo y mantenimiento), soporta todos los diagramas y modelos UML, describiendo un lenguaje visual por el cual se pueden modelar procesos de negocio, sitios web, interfaces de usuario, redes, configuraciones de hardware, mensajes, es decir los sistemas TICs en general, además es una herramienta que cubre todos los

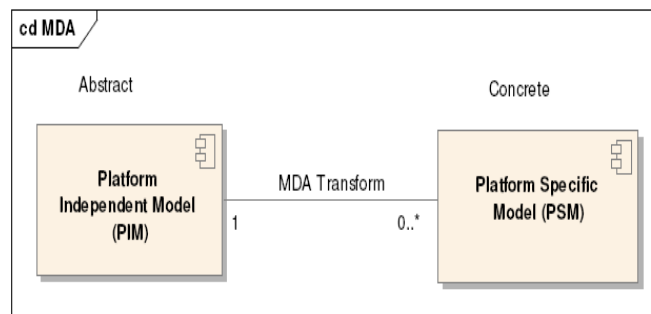
¹⁹ CASE (Computer Aided Software Engineering, "Ingeniería de Software asistida por computadora"). Son herramientas informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero.

aspectos del ciclo de vida del modelado y gestión de proyectos. Algunas de las características claves de Enterprise Architect son:

- Permite modelar elementos de UML ubicándolos en diagramas y paquetes y creando las conexiones entre ellos.
- Documentar los elementos que ha creado.
- Generar código para el software y gestionar requisitos.
- Realizar transformaciones MDA.

3.5.1. Transformaciones MDA

Enterprise Architect provee la capacidad de realizar transformaciones MDA cambiando los modelos de un dominio a otro, efectuando la conversión desde el modelo independiente de la plataforma (PIM) a modelos específicos de la plataforma PSMs (Modelos de datos DDL, C#, EJB, Java, XSD)²⁰.



Esquema 9. Transformaciones MDA. Extraído de EA

Para efectos de este trabajo, es más importante la construcción del PIM ya que este representa el comportamiento estático de un sistema, el cual es creado a partir del modelo de clases.

²⁰No es de interés para este trabajo de grado enfatizar sobre estas plataformas, además la herramienta genera automáticamente los PSMs.

3.6. PROPUESTA METODOLÓGICA PARA LA ESPECIFICACIÓN DE REQUISITOS DE SISTEMAS TIC UTILIZANDO MDA

Teniendo en cuenta las guías para la de especificación de requisitos²¹ se diseña una plantilla tipo documento para la especificación de requisitos de sistemas TIC [anexo A] y la guía de usuario [anexo B], sustentadas a través de los formatos utilizados por el CIDLIS [19].

3.6.1. Políticas del documento

En este documento se deben expresar las necesidades del cliente a través de requisitos del sistema,²² los cuales se redactan de manera clara y concisa para que puedan ser transmitidos y comprendidos fácilmente por los involucrados en el proyecto.

3.6.2. Recolección de información

La obtención y el análisis de los requisitos del sistema es necesaria para la identificación de los conceptos y apartados de la plantilla, esta fase se realiza mediante las técnicas explicadas en el numeral 2.3.2, con ello se logra describir el sistema de acuerdo con las características expresadas en el documento “Visión del Sistema”. Cuando nuevas características o necesidades sean añadidas o modificadas en este documento, dichos cambios deben verse reflejados en la especificación de requisitos del sistema mediante cambios o adiciones a los requisitos, ante esto se debe configurar una nueva versión.

Luego de que las necesidades del cliente son expresadas en un lenguaje natural en forma de requisitos, se extraen las clases que representan la estructura

²¹ Ver bibliografía [15], [17], [18]

²² En este apartado cuando se habla de requisitos del sistema, hace referencia a requisitos de sistemas TICs.

estática del sistema, posteriormente se expresan en un lenguaje más técnico por medio del modelo de clases de la aplicación, que dentro del marco de MDA viene siendo el modelo independiente de la plataforma y representa como tal el modelo de dominio del sistema, para finalmente realizar las transformaciones MDA con la herramienta EA. El esquema 10 muestra de manera general el procedimiento para la especificación de requisitos de sistemas TIC utilizando MDA, propuesto en este trabajo. Este procedimiento se utiliza en el capítulo 4 para especificar los requisitos de nuestro caso de estudio, “Diseño del cableado estructurado para comunicación de datos, voz y video de un edificio”.



Esquema 10. Procedimiento para la especificación de requisitos utilizando MDA. Fuente: Autora del trabajo.

4. CASO DE ESTUDIO

“DISEÑO DEL CABLEADO ESTRUCTURADO PARA COMUNICACIÓN DE VOZ, DATOS Y VIDEO”

El objetivo de este capítulo es realizar la “especificación de requisitos para el diseño del cableado estructurado para comunicación de voz, datos y video de un edificio”, teniendo en cuenta todos los conceptos mencionados en el desarrollo de este trabajo de investigación. El primer paso consiste en analizar el problema, aquí se realiza la descripción del sistema por medio del documento “Visión del sistema” (sección 2.3.1.), a partir de este documento se hace la especificación de requisitos del sistema por medio de la plantilla propuesta en el anexo A.

4.1. CABLEADO ESTRUCTURADO

Un sistema de cableado estructurado (SCE) es una forma ordenada y planeada de realizar cableados que permitan integrar servicios de voz, datos y vídeo; y cuyo objetivo primordial es abastecer un sistema total de transporte de información a través de un medio común en forma lógica y física con la finalidad de controlar los procesos y optimizar los recursos.

El SCE permite conectar equipos de procesamiento de datos, teléfonos, conmutadores, computadoras personales, redes de área local (LAN) y equipos de oficina entre sí; generalmente por medio de una red de computadoras o dispositivos de procesamiento.

A continuación se presentan los subsistemas o áreas que involucra el diseño de cableado estructurado.

4.1.1. Subsistemas del cableado estructurado

Instalación de entrada o acometida: punto donde la instalación exterior y dispositivos asociados entran al edificio, puede estar utilizado por servicios de redes públicas o privadas. En este punto se ubican los dispositivos de protección para sobrecargas de voltaje.

Área de trabajo: espacio donde interactúan los usuarios con los equipos de telecomunicaciones.

Cableado vertical (Backbone): cableado central que permite la interconexión entre los armarios de telecomunicaciones, cuartos de equipo, e instalaciones de entrada.

Cableado horizontal: formado por los cables que se extienden de manera horizontal a través de los pisos y techos desde el cuarto de telecomunicaciones hasta cada subsistema de estaciones de trabajo. Se pueden usar varios tipos de cable para la distribución horizontal. Cada tipo tiene sus propias limitaciones de desempeño, tamaño, costo, y facilidad de uso.

- ✓ Cable: Medio para el flujo de información, y para la interconexión de los diferentes aparatos de telecomunicación (UTP, coaxial y fibra óptica).
- ✓ Canaletas: Son utilizadas para distribuir y soportar el cableado horizontal y conectar hardware entre la salida del área de trabajo y el cuarto de telecomunicaciones.

Cuarto de telecomunicaciones: comprende el armario de telecomunicaciones, en el cual terminan los cables de distribución horizontal y vertical, conectados con puentes o cables de puenteo, a fin de proporcionar conectividad flexible para extender los diversos servicios a los usuarios en las tomas o salidas de telecomunicaciones.

4.2. APLICACIÓN DEL PROCEDIMIENTO

4.2.1. “Visión del sistema”

En la actualidad dentro de cualquier organización, existe la preocupación de cómo manejar los servicios de telecomunicaciones de modo eficaz, es por ello que surge la necesidad de implementar un sistema que permita integrar los servicios de voz, datos y vídeo de forma ordenada y eficiente, esto se logra con el cableado estructurado, que además aporta soluciones a las carencias de infraestructura de redes que pueda tener un edificio (oficinas comerciales, agencias gubernamentales, instituciones educativas y de salud, universidades, etc.)

El diseño del cableado estructurado debe ser flexible de acuerdo con la tecnología existente y debe indicar la funcionalidad mínima requerida por el edificio y por los usuarios, contemplando las siguientes áreas: área de trabajo, cableado vertical, cableado horizontal, cuarto de telecomunicaciones y administración.

Cada área debe cumplir con todos los requerimientos de TIA/EIA de espacios de telecomunicaciones.

Si el edificio se encuentra ocupado se deben tener en cuenta las alteraciones y molestias ocasionadas a los ocupantes del mismo.

Se deben realizar entrevistas al personal involucrado (personal administrativo y técnico, usuarios) sobre las diferentes herramientas telecomunicativas con que cuentan y sobre sus necesidades.

Para iniciar el diseño se deben conocer los planos del edificio y se deben estudiar las necesidades de interconexión y los servicios que se desean tener en sus diferentes áreas de trabajo teniendo en cuenta principalmente las siguientes consideraciones:

Área de trabajo

Los equipos de comunicaciones (teléfonos, fax, computadoras, etc.) pueden localizarse en cualquier área del edificio.

Todos los adaptadores, filtros, o acopladores usados para adaptar equipo electrónico diverso al sistema de cableado estructurado, deben ser ajenos a la toma o salida de telecomunicaciones.

Deben respetarse las distancias límite impuestas por las normas nacionales e internacionales para las distintas aplicaciones.

En esta área se deben incluir los *patch-cords* (conectores) que unen los equipos al área de trabajo, los cuales deben ser originales de fábrica, de acuerdo con la norma ANSI TIA/EIA 568B.

El conector debe estar diseñado con un mecanismo integral de bloqueo que proteja el ajuste mecánico de la conexión, el cual después de haber sido insertado, provea protección para no ser extraído de forma accidental.

Cableado vertical

Deben cumplir con las especificaciones de la norma ANSI/TIA/EIA 568B y ANSI/TIA/EIA 569A.

Nunca se deben cruzar los cables eléctricos y los de comunicaciones en ningún lugar.

Cableado horizontal

Se deben definir las rutas más adecuadas para distribuir la totalidad del cableado a lo largo de un piso y determinar el tipo de elemento a utilizar para transportar el cable (bandejas o canaletas de aluminio o de lamina, tuberías y ductos metálicos, o en mampostería, zócalos de divisiones o ventanas y muebles, canaletas perimetrales o por cielo raso, entre otros).

Los ductos y canalizaciones deben ser construidas con canaletas que cumplan con todos los requerimientos de TIA/EIA.

No se debe pasar cerca de tomas de agua, fuentes de humedad así como de zonas de alta temperatura.

Si hay cruce de cables, deberán hacerlo en ángulo recto para evitar el acoplamiento y no se permite hacer puentes en cableado.

No se deben cruzar los cables eléctricos y los de comunicaciones en ningún lugar.

Las canaletas deben fijarse a la parte superior de las paredes.

No se permite fijar las canaletas a la pared a través de adhesivos o pegamentos.

Cuarto de telecomunicaciones

El espacio del cuarto de telecomunicaciones debe ser utilizado exclusivamente para funciones de telecomunicaciones y servicios auxiliares relacionados con estos, y por ningún motivo debe ser compartido con instalaciones eléctricas diferentes a las requeridas por los equipos.

No debe estar cerca de lugares que se pueden inundar, como baños o lugares que puedan tener fugas de agua, lejos de fuentes de vapor, excesiva humedad y ambientes corrosivos que dañen los equipos o conexiones y de toda fuente de interferencia electromagnética.

Armario de telecomunicaciones:

Debe ser diseñado y equipado de acuerdo con ANSI/TIA/569 y cumplir con las especificaciones de ANSI/EIA-310.

Debe estar acondicionado eléctrica y ambientalmente para los equipos a instalar y acceder al sistema de toma a tierra de telecomunicaciones según lo especificado por la ANSI/EIA/TIA 607.

Debe tener puertas sin umbral, con bisagras que permitan abrirla hacia afuera, o deslizarla de un lado a otro, o desmontarla, y estar provista de una cerradura para seguridad.

Administración

✓ Identificadores de cables:

A cada cable se le debe asignar un único identificador, el cual servirá como enlace hacia el registro de cable correspondiente. Este identificador debe ser marcado en las etiquetas del cable.

Cuando se empalmen cables de la misma característica, deben ser considerados y administrados como un solo cable y cuando se empalman cables de diferentes capacidades en pares, se deben administrar como cables separados e independientes.

✓ Etiquetas de cables:

Todo el sistema de cableado deberá estar identificado por medio de una etiqueta auto adherible.

Los cables de los diferentes subsistemas de cableado deben ser etiquetados en cada uno de sus extremos. Para una administración completa, se deben colocar etiquetas en el cable en localizaciones intermedias tales como en extremos de tuberías, puntos de empalme en el cableado principal, registros subterráneos convencionales y en cajas de registro.

Otras consideraciones:

✓ Servicios de telecomunicaciones:

Para la comunicación de datos se debe definir la red de computadores teniendo en cuenta consideraciones como la topología, protocolo, servidor, entre otras.

Para la comunicación de voz se debe proporcionar como mínimo el servicio telefónico básico.

Para comunicación de video al sistema se deben proporcionar productos que permitan la supervisión y control, desde una central de monitoreo, de una o varias instalaciones técnicamente aisladas o distribuidas geográficamente.

✓ **Especificaciones técnicas para la red eléctrica:**

Se tendrán en cuenta las normas respectivas para este tipo de instalaciones en las partes que sean aplicables de acuerdo con los siguientes códigos:

- NEC “National Electrical Code”
- NEMA “National Electrical Manufactures Asociation”
- ICONTEC “Instituto Colombiano de Normas Técnicas 2050”

✓ **Puesta a tierra:**

Todos los componentes metálicos tanto de la estructura (tuberías y canaletas entre otros) como del mismo cableado (blindaje paneles y equipo) deben ser debidamente llevados a tierra para evitar descargas por acumulación de estática.

Todas las salidas eléctricas para computadores deben ser polarizadas y llevadas a una tierra común.

Para asegurar el máximo beneficio del SCE, el diseño se debe regir por las siguientes normas:

ANSI/EIA/TIA-568B y ANSI/EIA/TIA-569A²³ (Norma de construcción comercial para vías y espacios de telecomunicaciones), *“proporciona directrices para conformar ubicaciones, áreas, y vías a través de las cuales se instalan los equipos y medios de telecomunicaciones. También detalla algunas consideraciones a seguir cuando se diseñan y construyen edificios que incluyan sistemas de telecomunicaciones”*. [35]

ANSI/TIA/EIA-606A²⁴, (Norma de administración para la infraestructura de telecomunicaciones en edificios comerciales), que proporciona normas para la codificación de colores, etiquetado, y documentación de un sistema de cableado instalado.

ANSI/TIA/EIA-607A²⁵, (Requisitos de aterrizado y protección para telecomunicaciones en edificios comerciales), que describe los métodos estándares para distribuir las señales de tierra a través de un edificio.

En la implementación del diseño del cableado estructurado, los materiales a utilizar deben ser nuevos, de primera calidad y deben basarse en el estándar EIA/TIA-568B y EIA/TIA 569A “Commercial Building Standard for Telecommunications Pathways and Spaces”.

El fabricante de los componentes del sistema deberá tener certificación ISO 9001.

²³ ANSI/EIA/TIA-568B Y 569A “Commercial Building Standard for Telecommunications Pathways and Spaces”

²⁴ ANSI/TIA/EIA-606A “Administration Standard for the Telecommunications Commercial Building dura of Comercial Buildings”

²⁵ ANSI/TIA/EIA-607A “Commercial Building Grounding and Bonding Requeriments for Telecommunications”

4.2.2. Especificación de requisitos del sistema utilizando MDA

“DISEÑO DEL CABLEADO ESTRUCTURADO PARA COMUNICACIÓN DE VOZ, DATOS Y VIDEO DE UN EDIFICIO”

Autores:

Luisangela Orozco Ramírez

Dirigido a:

Centro de innovación y desarrollo para la investigación en ingeniería del software -
CIDLIS

Revisado Por:

Equipo de Revisión y Calidad CIDLIS

Lugar y fecha de emisión:

Bucaramanga, Abril de 2009

CONTENIDO

Primera parte.

1. Identificación.....	47
2. Introducción.....	49
3. Propósito del documento.....	49
4. Definiciones, acrónimos y abreviaturas.....	49
5. Funcionalidad general del sistema.....	49
6. Especificación de Requisitos.....	49
6.1 Requisitos funcionales.....	51
6.2 Requisitos No funcionales.....	51
6.3 Requisitos de Interfaz.....	51
7. Estándares aplicables.....	51
8. Cambios.....	52
9. Información adicional.....	52
10. Acta de aprobación.....	52

Segunda parte.

11. Clases.....	53
11.1 Especificación de las clases.....	53
11.2 Diagrama modelado de clases.....	58

Tercera parte.

12. Transformaciones MDA.....	59
-------------------------------	----

Primera parte.

2. Introducción

Este documento se organiza en tres partes, la primera de ellas contiene la especificación de requisitos para el diseño del cableado estructurado para comunicación de voz, datos y video de un edificio, la segunda contiene el modelo de dominio del sistema representado por medio del diagrama de clases y la tercera la implementación de las transformaciones MDA.

El objetivo del diseño es especificar el trazado de rutas y las normas técnicas para el cableado estructurado de un edificio, y lograr con ello reducir tiempo y costos, aumentar la calidad y la cantidad de la información.

El diseño debe cumplir con los requerimientos, las especificaciones técnicas y la normativa de acuerdo a los estándares nacionales e internacionales.

3. Propósito del documento

El objeto de este documento es especificar los requerimientos funcionales y no funcionales que se deberán seguir para diseñar un sistema de cableado estructurado para comunicación de voz, datos y video de un edificio.

Así mismo, establece las características del sistema de cableado y define la ingeniería de detalle del mismo con el objeto de identificar en forma clara y precisa cada uno de los componentes involucrados en el sistema y subsistemas, y establecer así, su ubicación dentro del área física de la instalación e identificar las normas que rigen el diseño.

Los requisitos contenidos en este documento estarán sujetos a las revisiones realizadas por el equipo de calidad del CIDLIS generándose las correspondientes versiones del presente documento, hasta conseguir su aprobación por parte del representante del cliente.

4. Definiciones, acrónimos y abreviaturas

TÉRMINO	DESCRIPCIÓN
ESPECIFICACIÓN DE REQUISITOS	Documento estructurado que presenta los resultados de la definición de necesidades, la conceptualización operacional y las actividades de análisis de un diseño de sistema o producto software.
REQUISITO	Necesidad o expectativa establecida o habitualmente implícita u obligatoria (NTC- ISO 9000).
REQUISITO FUNCIONAL	Operación o función que debe ser efectuada por un sistema o componente del

	mismo.
REQUISITO NO FUNCIONAL	Condición o restricción establecida para la implementación de los requisitos funcionales.
MODELO	Representación de la función y el comportamiento de un sistema
CLASE	Abstracción de un conjunto de objetos que comparten las mismas características
ACTOR	Representa un rol que el usuario puede tomar respecto al sistema.
SCE	Sistema de cableado estructurado.
SISTEMA	Grupo de elementos relacionados que interactúan para realizar una tarea.
CABLEADO ESTRUCTURADO	Forma ordenada y planeada de realizar cableados que permitan integrar servicios de telecomunicaciones.(datos, voz, video).

5. Funcionalidad general del sistema

El diseño debe proporcionar soporte para que el edificio cuente con una red de cableado estructurado de telecomunicaciones que facilitará la implementación de los servicios de voz, datos y video y asegurará su calidad.

Además, permite que el edificio pueda contar con una red estructurada de forma tal que deje centralizar información, almacenar, manejar y compartir recursos, soportar aplicaciones sobre voz (telefonía), datos, video (vídeo vigilancia) y otras aplicaciones, suministrando la confiabilidad y la escalabilidad de los recursos computacionales.

6. Especificación de Requisitos

6.1 Requisitos funcionales

Requisito 1.

CÓDIGO	REQ_01
NOMBRE	DEFINICIÓN DE LA SITUACIÓN GEOGRÁFICA
DESCRIPCIÓN O CARACTERÍSTICAS	Para el diseño se debe describir la localización del edificio, además de especificar si está ocupado o desocupado.
TIPO	Requisito Funcional

Requisito 2.

CÓDIGO	REQ_02
NOMBRE	DESCRIPCIÓN ARQUITECTÓNICA DEL EDIFICIO
DESCRIPCIÓN O CARACTERÍSTICAS	Se debe contar con el plano del edificio a efectos de poder esquematizar el cableado. Identificar los lugares del edificio donde se requiere puntos de interconexión.
TIPO	Requisito Funcional

Requisito 3.

CÓDIGO	REQ_03
NOMBRE	COMUNICACIÓN DE DATOS
DESCRIPCIÓN CARACTERÍSTICAS O	Definir la red de computadores teniendo en cuenta consideraciones como la topología, protocolo, servidor, entre otras.
TIPO	Requisito Funcional

Requisito 4.

CÓDIGO	REQ_04
NOMBRE	COMUNICACIÓN DE VOZ
DESCRIPCIÓN CARACTERÍSTICAS O	Para la comunicación de voz se debe proporcionar como mínimo servicio telefónico básico.
TIPO	Requisito Funcional

Requisito 5.

CÓDIGO	REQ_05
NOMBRE	COMUNICACIÓN DE VIDEO
DESCRIPCIÓN CARACTERÍSTICAS O	Para comunicación de video se debe proporcionar al sistema de productos que permitan la supervisión y control, desde una central de monitoreo, de una o varias instalaciones técnicamente aisladas o distribuidas geográficamente.
TIPO	Requisito Funcional

Requisito 6.

CÓDIGO	REQ_06
NOMBRE	DEFINICIÓN DE LAS ÁREAS DE TRABAJO
DESCRIPCIÓN CARACTERÍSTICAS O	<p>El diseño debe determinar los dispositivos de interconexión y los equipos de comunicaciones que serán necesarios para el diseño.</p> <p>Todos los adaptadores, filtros, o acopladores usados para adaptar equipo electrónico diverso al sistema de cableado estructurado, deben ser ajenos a la toma o salida de telecomunicaciones.</p> <p>Deben respetarse las distancias límite impuestas por las normas nacionales e internacionales para las distintas aplicaciones.</p> <p>En esta área se deben incluir los conectores que unen los equipos al área de trabajo, los cuales deben ser originales de fábrica, de acuerdo con la norma ANSI TIA/EIA 568B.</p> <p>El conector debe estar diseñado con un mecanismo integral de bloqueo que proteja el ajuste mecánico de la conexión, el cual después de haber sido insertado, provea protección para no ser extraído de forma accidental.</p>
TIPO	Requisito Funcional

Requisito 7.

CÓDIGO	REQ_07
NOMBRE	DEFINICIÓN DEL CUARTO DE TELECOMUNICACIONES
DESCRIPCIÓN CARACTERÍSTICAS ○	<p>El diseño debe proporcionar un sitio estratégico donde funcionará el cuarto de comunicaciones y los cuartos de equipo.</p> <p>El espacio del cuarto de telecomunicaciones debe ser utilizado exclusivamente para funciones de telecomunicaciones y servicios auxiliares relacionados con estos, y por ningún motivo debe ser compartido con instalaciones eléctricas diferentes a las requeridas por los equipos.</p> <p>No debe estar cerca de lugares que se pueden inundar, como cerca de baños, o lugares que puedan tener fugas de agua y debe estar lejos de fuentes de vapor, excesiva humedad y ambientes corrosivos que dañen los equipos o conexiones y de toda fuente de interferencia electromagnética.</p> <p>Debe estar alejado por lo menos 3 metros, de toda fuente de interferencia electromagnética.</p> <p>El armario de telecomunicaciones debe ser diseñado y equipado de acuerdo con ANSI/TIA/569 y cumplir con las especificaciones de ANSI/EIA-310.</p>
TIPO	Requisito Funcional

Requisito 8.

CÓDIGO	REQ_08
NOMBRE	CONSIDERACIONES DE CABLEADO
DESCRIPCIÓN CARACTERÍSTICAS ○	<p>Determinar los dispositivos de interconexión que serán necesarios para el diseño.</p> <p>El cableado deben cumplir con las especificaciones de la norma ANSI/TIA/EIA 568B y ANSI/TIA/EIA 569A.</p> <p>Nunca se deben cruzar los cables eléctricos y los de comunicaciones en ningún lugar.</p> <p>La distancia entre los cables de red y los de la corriente eléctrica deben ser superiores a 30 cm.</p> <p>La longitud del cable deberá respetar la máxima distancia establecida por el estándar TIA/EIA-568-A.</p> <p>Se deben definir las rutas más adecuadas para distribuir la totalidad del cableado y determinar el tipo de elemento a utilizar para transportar el cable (bandejas o canaletas de aluminio o de lamina, tuberías y ductos metálicos, o en mampostería, zócalos de divisiones o ventanas y muebles, canaletas perimetrales o por cielo raso, entre otros).</p> <p>Los ductos y canalizaciones deben ser construidas con canaletas que cumplan con todos los requerimientos de TIA/EIA.</p> <p>Se debe evitar pasar cerca de tomas de agua, fuentes de humedad así como de zonas de alta temperatura.</p> <p>Si hay cruce de cables, deberán hacerlo en ángulo recto para evitar el acoplamiento y no se permiten hacer puentes en cableado.</p> <p>Nunca se deben cruzar los cables eléctricos y los de comunicaciones en ningún lugar.</p> <p>Las canaletas deben fijarse a la parte superior de las paredes.</p> <p>No se permite fijar las canaletas a la pared a través de adhesivos o pegamentos.</p>
TIPO	Requisito Funcional

Requisito 9.

CÓDIGO	REQ_09
NOMBRE	ELEMENTOS DE ADMINISTRACIÓN
DESCRIPCIÓN CARACTERÍSTICAS O	Todo el sistema de cableado deberá estar identificado por medio de una etiqueta autoadherible. A cada cable se le debe asignar un único identificador, este identificador debe ser marcado en las etiquetas del cable. Cuando se empalmen cables de la misma característica, deben ser considerados y administrados como un solo cable y cuando se empalman cables de diferentes capacidades en pares, se deben administrar como cables separados e independientes. Para una administración completa, se deben colocar etiquetas en el cable en localizaciones intermedias tales como en extremos de tuberías, puntos de empalme en el cableado principal, registros subterráneos convencionales y en cajas de registro.
TIPO	Requisito Funcional

Requisito 10.

CÓDIGO	REQ_10
NOMBRE	PROTECCIÓN ELÉCTRICA
DESCRIPCIÓN CARACTERÍSTICAS O	Se deben considerar las especificaciones técnicas para la red eléctrica, de acuerdo a las norma TIA/EIA-697. Todos los componentes metálicos tanto de la estructura (tuberías y canaletas entre otros) como del mismo cableado (blindaje paneles y equipo) deben ser debidamente llevados a tierra para evitar descargas por acumulación de estática. Todas las salidas eléctricas para computadores deben ser polarizadas y llevadas a una tierra común. Todos los cables de puesta a tierra deben tener una aislacion. El principal punto de entrada/cuarto de equipos en cada edificio debe ser equipado con una barra principal de tierra. Cada cuarto de datos debe ser provisto con una barra de puesta a tierra
TIPO	Requisito Funcional

Requisito 11.

CÓDIGO	REQ_11
NOMBRE	INFORMACIÓN DE LA RED DE TELECOMUNICACIONES
DESCRIPCIÓN CARACTERÍSTICAS O	Se debe proporcionar información sobre los servicios prestados.
TIPO	Requisito Funcional

6.2 Requisitos No funcionales

Los requisitos no funcionales corresponden a condiciones o restricciones que debe cumplir el diseño y son complementarios a los requisitos funcionales del sistema.

Requisito 12.

CÓDIGO	REQ_12
NOMBRE	EL DISEÑO DEBE SEGUIR LAS NORMAS ANSI TIA/EIA
DESCRIPCIÓN O CARACTERÍSTICAS	Para asegurar el máximo beneficio del SCE, el diseño se debe regir por las siguientes normas: ANSI/EIA/TIA-568B y ANSI/EIA/TIA-569”Norma de construcción comercial para vías y espacios de telecomunicaciones”. ANSI/TIA/EIA-606, “Norma de administración para la infraestructura de telecomunicaciones en edificios comerciales” ANSI/TIA/EIA-607, “Requisitos de aterrizado y protección para telecomunicaciones en edificios comerciales”
TIPO	Requisito No Funcional

6.3 Requisitos de Interfaz

Requisito 13.

CÓDIGO	REQ_13
NOMBRE	INFORMACIÓN DEL SISTEMA OPERATIVO
DESCRIPCIÓN O CARACTERÍSTICAS	El diseño debe informar todas las características del sistema operativo del edificio, así como el servidor del sistema, software de administración, topología, protocolos, etc.
TIPO	Requisito de Software

7. Estándares aplicables

ESTÁNDAR	DESCRIPCIÓN
EIA/TIA-568 B.1 “Commercial Building Telecommunications Cabling Standard”	Directrices generales de diseño y construcción de un sistema de telecomunicaciones en categoría 6.
EIA/TIA-568 B.2 “Commercial Building Telecommunications Cabling Standard”	Directrices de los diferentes componentes de un sistema de telecomunicaciones basado en transmisión en cables de pares trenzados.
EIA/TIA-569A “Commercial Building Standard for Telecommunications Pathways and Spaces”	Estandariza prácticas de diseño y construcción dentro o entre edificios, que son hechas en soporte de medios y/o equipos de telecomunicaciones tales como canaletas y guías, facilidades de entrada al edificio, armarios y/o closets de comunicaciones y cuartos de equipos.
EIA/TIA-606A “Administration Standard for the Telecommunications Infrastructure of Commercial Buildings”	Guías para marcar y administrar los componentes de un sistema de Red de datos.
EIA/TIA-607 “Commercial Building Grounding and Bonding Requirements for Telecommunications”	Describe los métodos estándares para distribuir las señales de tierra a través de un edificio.

8. Cambios

El desarrollo del diseño de SCE y sus productos estarán sometidos a las revisiones, verificaciones y demás mecanismos de control establecidos por el CIDLIS en su sistema de gestión de calidad y por los representantes técnicos en servicios de telecomunicaciones (ingenieros, contratista, instaladores, etc.)

FICHA DE CONTROL DE CAMBIOS

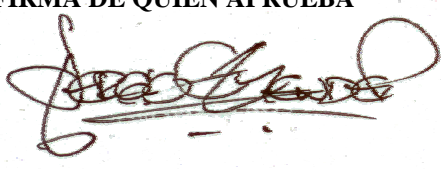
FECHA	VERSIÓN	DESCRIPCIÓN DEL CAMBIO
Abril de 2009	VERSIÓN. 1	

9. Información adicional

Una vez que se acepte el diseño se debe implementar e inspeccionar todo el sistema de cableado estructurado para verificar que cumple con todos los requisitos.

Instalada la infraestructura de telecomunicaciones diseñada se debe poner en marcha un plan de mantenimiento preventivo, así como también la capacitación y adiestramiento al personal que va a estar a cargo de los diferentes equipos.

10. Acta de aprobación

ACTA DE APROBACIÓN	
CLIENTE	CIDLIS
FECHA	Abril 20 de 2.009
NOMBRE DE QUIEN APRUEBA Sergio Enrique Méndez Aceros _____	NOMBRE DE QUIEN ENTREGA Luisangela Orozco Ramírez _____
FIRMA DE QUIEN APRUEBA  _____	FIRMA DE QUIEN ENTREGA _____ _____

Segunda parte.

11. Clases

11.1. Especificación de las clases

Clase 1.

CÓDIGO	CL_01
NOMBRE	PLANO
DESCRIPCIÓN	Identificación de las plantas del edificio, dependencias.
REQUISITO ASOCIADO	REQ-01 REQ-02

Clase 2.

CÓDIGO	CL_02
NOMBRE	CABLEADO_ESTRUCTURADO
DESCRIPCIÓN	Definición del diseño del sistema de cableado estructural de edificio. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-08 REQ-11

Clase 3.

CÓDIGO	CL_03
NOMBRE	AREA_TRABAJO
DESCRIPCIÓN	Determinar los dispositivos de interconexión y los equipos de comunicaciones.
REQUISITO ASOCIADO	REQ-02 REQ-06 REQ-14

Clase 4.

CÓDIGO	CL_04
NOMBRE	CABLE
DESCRIPCIÓN	Identificar propiedades del cable. Determinar el tipo de cableado. Cumplir con las especificaciones de la norma ANSI/TIA/EIA 568B y ANSI/TIA/EIA 569A.
REQUISITO ASOCIADO	REQ-02 REQ-06 REQ-08 REQ-09 REQ-10 REQ-11 REQ-12

Clase 5.

CÓDIGO	CL_05
NOMBRE	CABLEADO_VERTICAL
DESCRIPCIÓN	Definir ruta, tipo de cable. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-02 REQ-06 REQ-08 REQ-09 REQ-10 REQ-11 REQ-12

Clase 6.

CÓDIGO	CL_06
NOMBRE	CABLEADO_HORIZONTAL
DESCRIPCIÓN	Definir ruta, tipo de cable. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-02 REQ-06 REQ-08 REQ-09 REQ-10 REQ-11 REQ-12

Clase 7.

CÓDIGO	CL_07
NOMBRE	CUARTO_TELECOMUNICACIONES
DESCRIPCIÓN	Determinar sitio estratégico donde funcionará el cuarto de comunicaciones. Funciones y servicios. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-02 REQ-07 REQ-11 REQ-12 REQ-13

Clase 8.

CÓDIGO	CL_08
NOMBRE	ARMARIO
DESCRIPCIÓN	Cumplir con las especificaciones de la norma ANSI/TIA/569 y ANSI/TIA/310
REQUISITO ASOCIADO	REQ-02 REQ-07 REQ-11 REQ-12

Clase 9.

CÓDIGO	CL_09
NOMBRE	ADMINISTRACIÓN
DESCRIPCIÓN	Identificación del sistema de cableado por medio de una etiqueta. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-08 REQ-09 REQ-11 REQ-12

Clase 10.

CÓDIGO	CL_10
NOMBRE	RED_TELECOMUNICACIONES
DESCRIPCIÓN	Prestar servicios de datos, voz y video. Identificación del sistema y equipos. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-03 REQ-04 REQ-05 REQ-06 REQ-08 REQ-11 REQ-12 REQ-13 REQ-14

Clase 11.

CÓDIGO	CL_11
NOMBRE	COMUNICACIÓN_VOZ
DESCRIPCIÓN	Determinar servicio de telefonía. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-03 REQ-04 REQ-05 REQ-06 REQ-08 REQ-11 REQ-12 REQ-13 REQ-14

Clase 12.

CÓDIGO	CL_12
NOMBRE	COMUNICACIÓN_DATOS
DESCRIPCIÓN	Definir red para el servicio de comunicación de datos. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-03 REQ-04 REQ-05 REQ-06 REQ-08 REQ-11 REQ-12 REQ-13 REQ-14

Clase 13.

CÓDIGO	CL_13
NOMBRE	COMUNICACIÓN_VIDEO
DESCRIPCIÓN	Definir la comunicación de video, y productos que permitan la supervisión y control, desde una central de monitoreo. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-03 REQ-04 REQ-05 REQ-06 REQ-08 REQ-11 REQ-12 REQ-13 REQ-14

Clase 14.

CÓDIGO	CL_14
NOMBRE	RED_COMPUTADORES
DESCRIPCIÓN	Definir la red de computadores teniendo en cuenta consideraciones como la topología, protocolo, servidor, entre otras. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-03 REQ-04 REQ-05 REQ-06 REQ-08 REQ-10 REQ-11 REQ-12 REQ-13 REQ-14

Clase 15.

CÓDIGO	CL_15
NOMBRE	TELEFONÍA
DESCRIPCIÓN	Definir el servicio de comunicación de voz. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-03 REQ-04 REQ-05 REQ-06 REQ-08 REQ-10 REQ-11 REQ-12 REQ-13 REQ-14

Clase 16.

CÓDIGO	CL_16
NOMBRE	CAMARA
DESCRIPCIÓN	Definir el servicio de video vigilancia. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-03 REQ-04 REQ-05 REQ-06 REQ-08 REQ-10 REQ-11 REQ-12 REQ-13 REQ-14

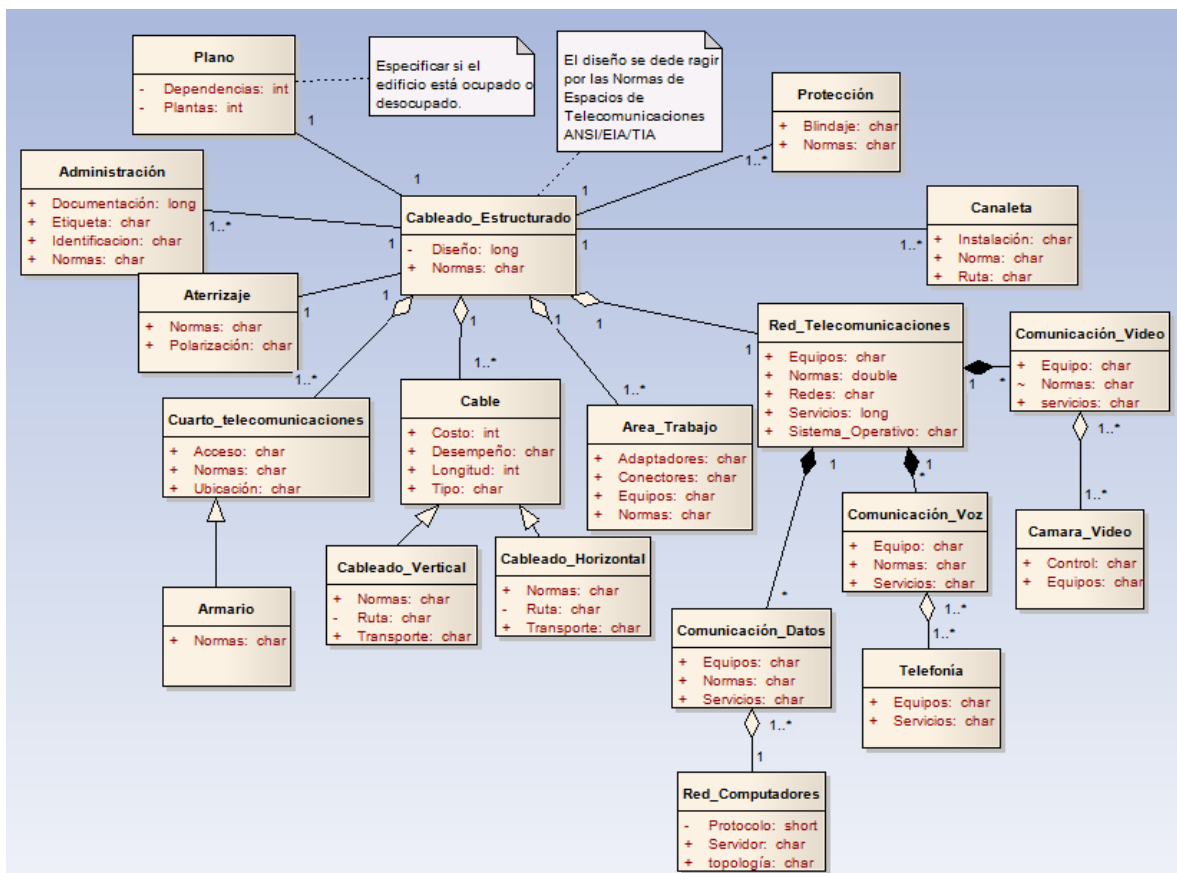
Clase 17.

CÓDIGO	CL_06
NOMBRE	CANALETA
DESCRIPCIÓN	Definir ruta para transportar cable. Cumplir con las especificaciones de las normas ANSI/TIA/EIA.
REQUISITO ASOCIADO	REQ-08 REQ-11 REQ-12

Clase 18.

CÓDIGO	CL_06
NOMBRE	ATERRIZAJE
DESCRIPCIÓN	Determinar polarización y puesta a tierra.
REQUISITO ASOCIADO	REQ-02 REQ-03 REQ-04 REQ-05 REQ-06 REQ-10 REQ-11 REQ-12 REQ-14

11.2. Diagrama de modelado de clases

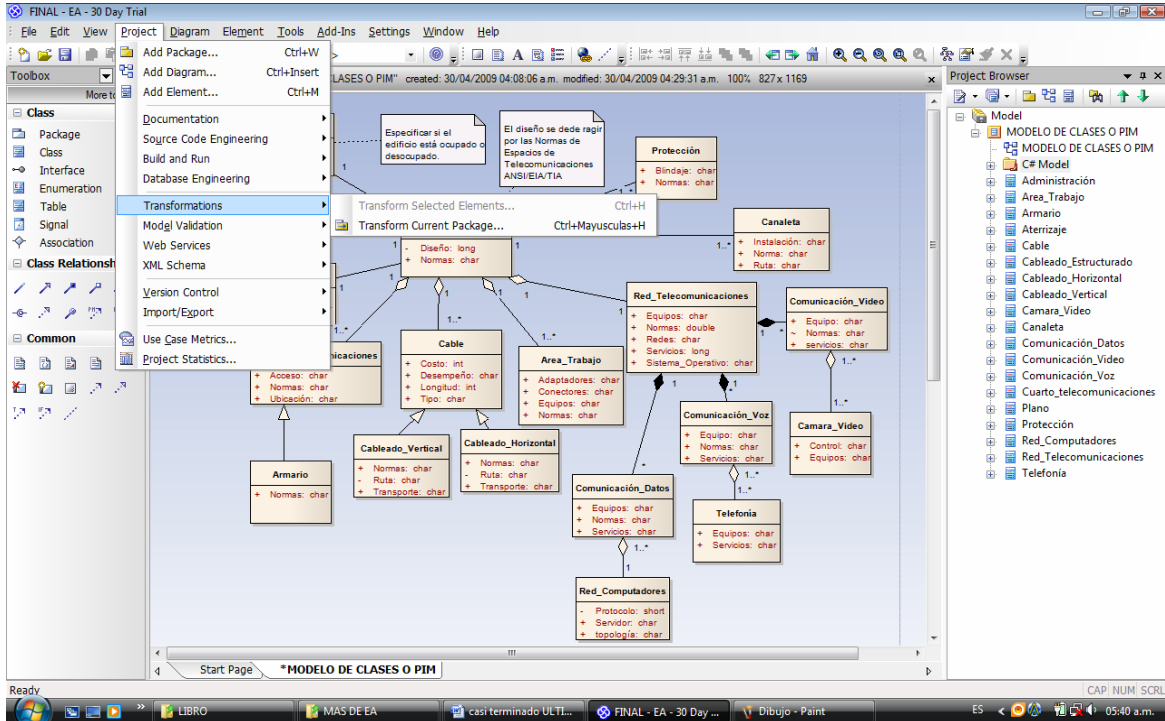


Este diagrama representa el dominio del problema en forma estática, entre más se extraigan las clase más cercano se hace la representación del modelo a la realidad del sistema.

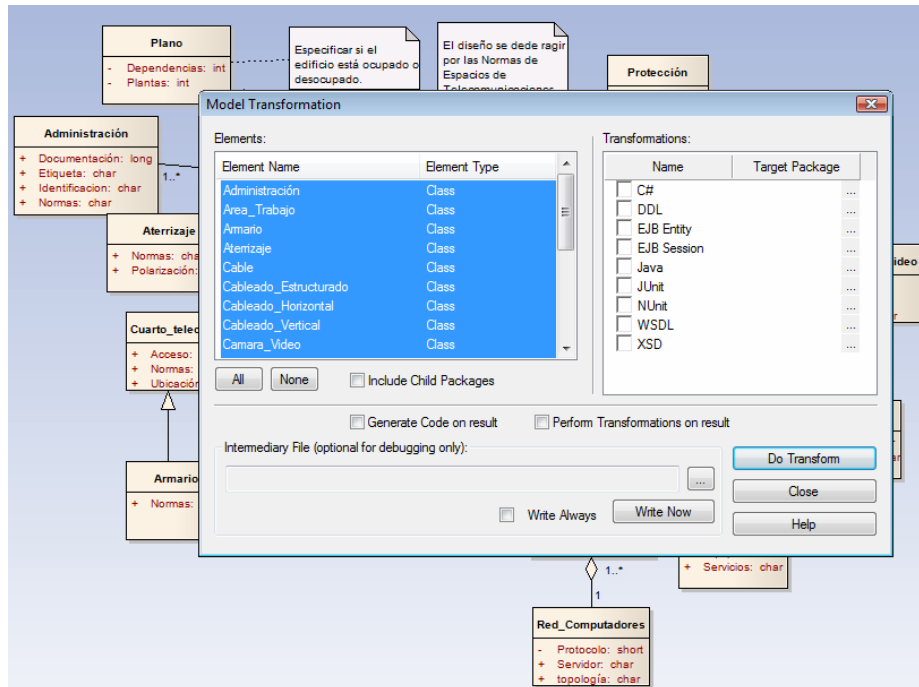
NOTA: En MDA este es el PIM o modelo independiente de la plataforma.

Tercera parte.

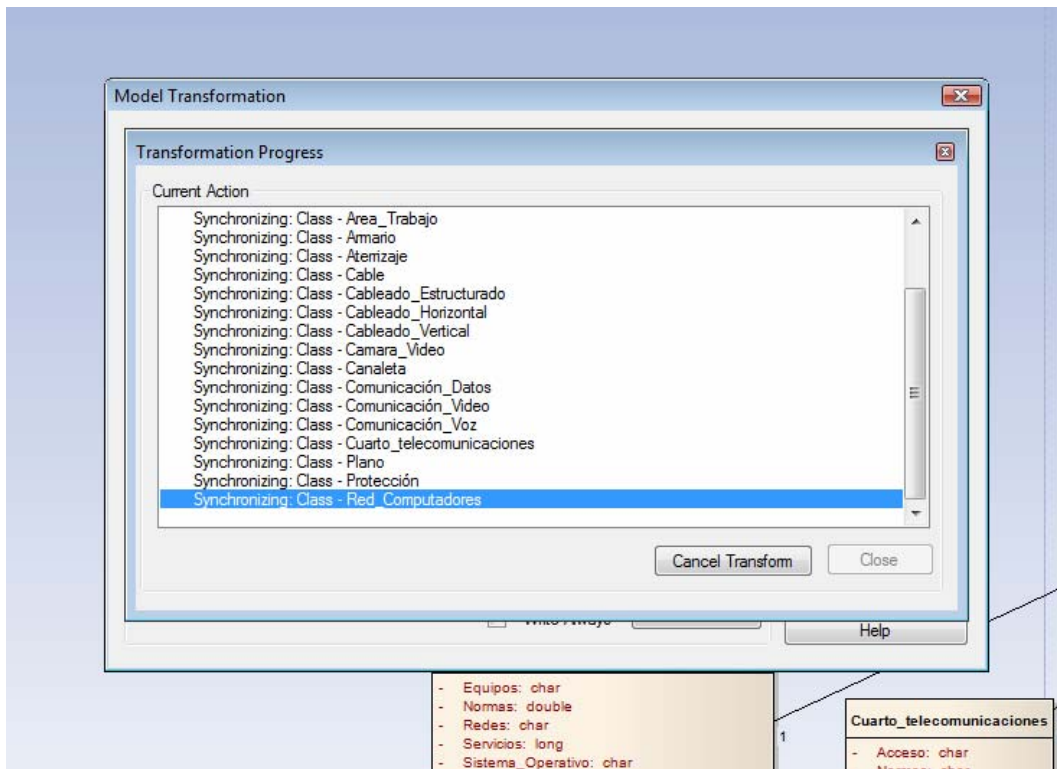
12. Transformaciones MDA



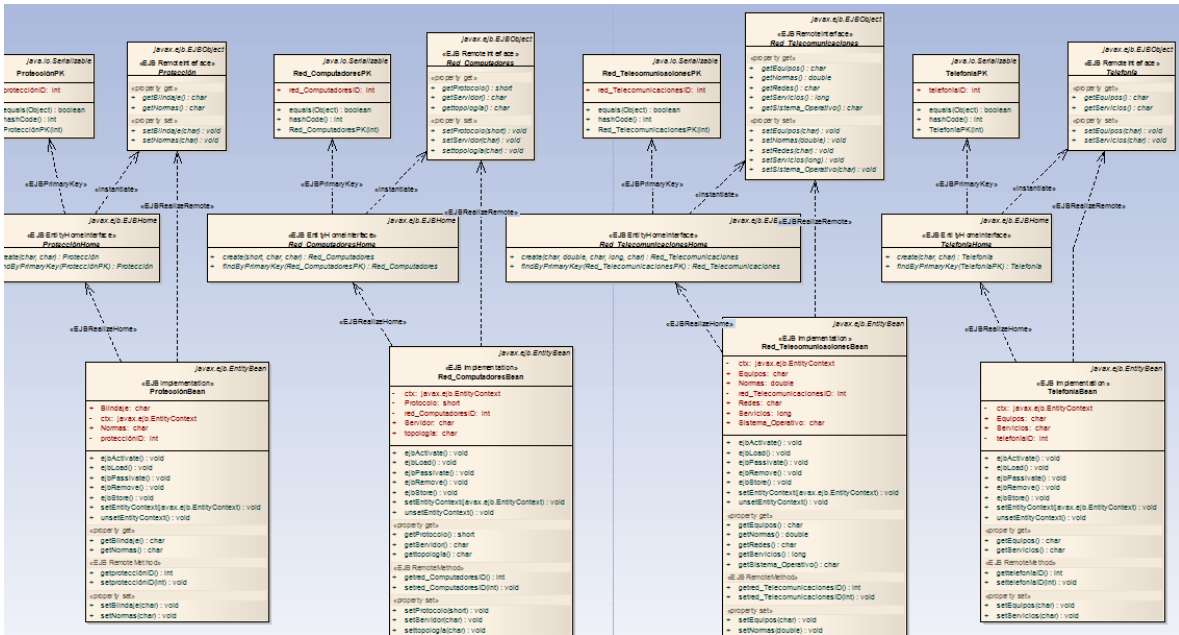
Se selecciona la plataforma en que se desea la representación del sistema.



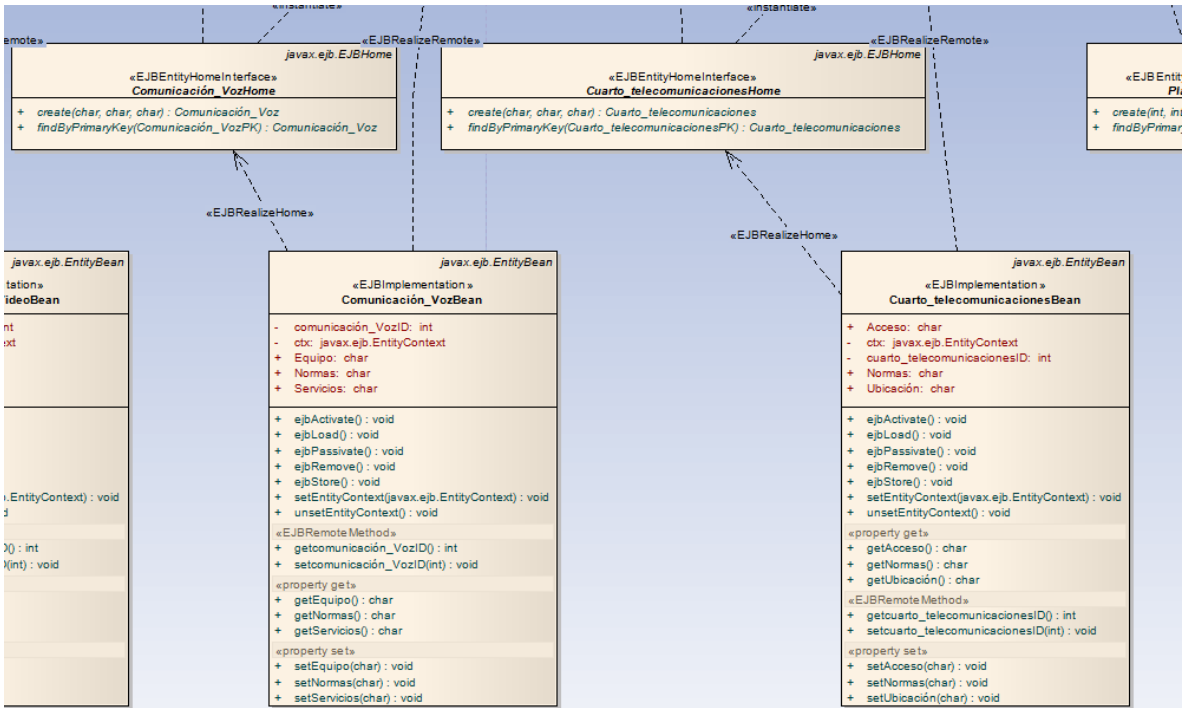
Para el ejemplo se escogió EJB Entity.



PSM de la aplicación.



Acercamiento a la clase Comunicación_Voz y Cuarto_telecomunicaciones, en el PSM.



4.3. CONCLUSIONES

Se establecieron variables para el análisis de vida del desarrollo de sistemas TIC desde la perspectiva del ciclo de vida del producto en ingeniería de software, como lo son: la obtención y el análisis de requisitos, el dominio del sistema y su representación, el modelado de objetos y la implementación de una herramienta de soporte; en coherencia con MDA.

Se pudo reconocer que la etapa de requisitos es la más importante dentro del desarrollo de un sistema de software, ya que repercute directamente sobre todas las demás etapas, lo que conlleva a que los resultados se encuentren directamente ligados con ellos.

Reconocidas las variables a tratar y el ciclo de vida del producto software correspondiente, y habiendo efectuado una profundización debida en el marco teórico, se estableció un procedimiento destinado a la especificación de requisitos de sistemas TIC teniendo en cuenta el modelo de MDA y estándares de ingeniería de software propuestos por el IEEE, el cual se compiló en una plantilla para este fin. Adicionalmente, se incluyó en el procedimiento propuesto una serie de instrucciones encaminadas a desplegar la especificación en la herramienta Enterprise Architect, con la finalidad de automatizar y gestionar las acciones posteriores en el marco del ciclo de vida del producto software.

Tomando como caso de estudio el diseño del cableado estructurado para comunicación de voz, datos y video en un edificio, se llevó a cabo la especificación de requisitos de este sistema TIC, sustentando en el procedimiento propuesto y con aceptación de la entidad interesada. Mediante esto, puede notarse que es posible efectuar el procedimiento propuesto a cualquier sistema TIC.

Se comprobó que un sistema cambia a lo largo del tiempo y que este es un factor que no se puede evitar, sin embargo con MDA se logró optimizar esta situación porque esta herramienta permite efectuar cambios en el dominio del

problema, es decir, en el modelo independiente de la plataforma (PIM) logrando que éstos se trasladen automáticamente al resto del sistema.

Se elaboró una guía para la especificación de requisitos respecto de la plantilla propuesta como procedimiento para tal fin.

Con el desarrollo de la investigación se presentó un crecimiento académico, debido a que se pudo profundizar sobre una disciplina que poco se trata en nuestra carrera de ingeniería electrónica, pero que día a día adquiere más reconocimiento e influencia en todas las áreas, es por ello que se comprendió que la ingeniería de software es más que programas de computador y va mas allá de la actividad principal de programación, que en cambio es una actividad desde la perspectiva de modelado para resolver problemas y adquirir conocimiento.

En síntesis, se elaboró un procedimiento para sistemas TIC desde la perspectiva de la ingeniería del software que abarca las primeras etapas del ciclo de vida del desarrollo de software, como lo son: la definición, especificación y diseño. Esta acción ha dejado paso abierto para que otros trabajos de grado realicen procedimientos para las etapas subsiguientes del ciclo de vida del producto en ingeniería de software: desarrollo, integración, pruebas y mantenimiento.

A partir de la elaboración de este trabajo, la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones, así como el Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software (CIDLIS) cuentan con un procedimiento para la especificación de requisitos destinada al desarrollo de sistemas TIC basado en MDA y desde la perspectiva de la ingeniería de software.

4.4. RECOMENDACIONES

Se recomienda llevar a cabo la continuación de este trabajo de grado desarrollando procedimientos para las etapas de desarrollo, integración, pruebas y mantenimiento del sistema TIC.

Igualmente, se recomienda que los proyectos relacionados con TIC aborden las etapas del ciclo de vida del producto en ingeniería de software dado que la metodología propuesta permite tener claridad en el desarrollo de este tipo de proyectos.

Fomentar la investigación en temas relacionados con la ingeniería del software por medio de trabajos de grado, en conjunto con el Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software (CIDLIS), así como también incentivar en el estudiante la investigación en las nuevas Tecnologías de Información y Comunicación (NTIC).

Así mismo, se propone que la Universidad de impulso a la apertura de pasantías en grupos de investigación y otras modalidades de trabajo de grado que conlleven a formar habilidades y competencias en temas relacionados con ingeniería de software para que los estudiantes de ingenierías obtengan experiencias que les sean de utilidad en el desarrollo de su pronto ejercicio profesional.

Dado que este tipo de trabajos y proyectos relacionados con la ingeniería de software, permite la integración con las demás carreras relacionadas con esta área, facilitando a los estudiantes una formación integral en varias áreas de la ingeniería dentro del campo de acción académico de la facultad de ingenierías físico-mecánicas; se recomienda que se planteen más proyectos tanto a nivel de aula como interinstitucionales en esta disciplina para incrementar la experiencia de los integrantes de la comunidad de esta facultad en esta rama de la ingeniería.

BIBLIOGRAFÍA

[1] BRUEGGE, Bernd. DUTOIT, Allen. Ingeniería de software orientada a objetos. Prentice Hall, 2002.

[2] HERNÁNDEZ SAMPIERI, Roberto. FERNÁNDEZ COLLADO, Carlos. BAPTISTA LUCIO, Pilar. Metodología de la Investigación. Tercera Edición. México D.F, McGraw-Hill Interamericana, 2003.

[3] Tecnologías de la Información y la Comunicación. Disponible en: <http://es.wikipedia.org/wiki> . Última visita: Noviembre de 2008.

[4] BRAUDE, Erik J. Software Engineering: An Object-Oriented Perspective, 2001.

[5] LLAMOSAS VILLALBA, Ricardo. Empresas Informáticas: Ingeniería del software para el emprendimiento industrial informático. CIDLIS. 2001.

[6] MORENO, Ángel. Gestión de proyectos software. Departamento de automática, Universidad de Alcalá. Disponible en: <http://it.aut.uah.es/enrique/docencia/it/proyectos/teoria/t8pit0607.pdf> . Última visita: Diciembre de 2008.

[7] PRESSMAN, Roger. Ingeniería del Software. Un enfoque práctico. Quinta Ed. McGraw Hill, 2002.

[8] BERTRAND, Meyer. Construcción de software orientado a objetos. Prentice Hall, 1999.

[9] Foros del web. Ingeniería del software y gestión de proyectos. Disponible en: <http://www.forosdelweb.com/f50/que-ingenieria-del-software-127123/> . Última visita: Enero de 2009.

[10] ZABALA RUIZ, J. Apuntes sobre ingeniería del software, 2008. Disponible en:<http://www.angelfire.com>. Última visita: Enero de 2009.

[11] JACOBSON, J. BOOCH, G. RUMBAUGH, I. El Proceso Unificado de Desarrollo Software. Addison Wesley, 1999.

[12] JACOBSON, I. "Applying UML in The Unified Process" Presentación. Rational Software, 1998. Disponible en: <http://www.rational.com/uml> . Última visita: Marzo de /2009.

[13] PALACIO BAÑERES, Juan. Compendio de Ingeniería del Software, 2006. Disponible en:
http://www.navegapolis.net/files/cis/CIS_2_04.pdf. ISBN: 0710040047711. Última visita: Abril/2009.

[14] SCHACH, Stephen. Ingeniería de software clásica y orientada a objetos. Mc Graw Hill, 2006.

[15] IEEE, Guide for Developing System Requirements Specifications. Standard 1233–1998, Institute of Electrical and Electronics Engineers, 1998.

[16] IEEE, Recommended Practice for Software Requirements Specifications. Standard 830–1993, Institute of Electrical and Electronics Engineers, 1993.

[17] IEEE, Guide for Developing System Requirements Specifications. Standard 1362–1998, Institute of Electrical and Electronics Engineers, 1998.

[18] CIDLIS, Guía para elaboración de Especificación de Requisitos de Sistema. Bucaramanga, Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software, 2008.

[19] CIDLIS, Plantilla para la Especificación de Requisitos de Sistema. Bucaramanga, Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software, 2008.

[20] CIDLIS, Guía para la elaboración de Especificación de Requisitos de Software. Bucaramanga, Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software, 2008.

[21] CIDLIS, Plantilla para la Especificación de Requisitos de Software. Bucaramanga, Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software, 2008.

[22] CIDLIS, guía para elaboración del Documento Visión del Sistema. Bucaramanga, Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software, 2008.

[23] CIDLIS, Plantilla para elaboración del Documento Visión del Sistema. Bucaramanga, Centro de Innovación y Desarrollo para la Investigación en Ingeniería del Software, 2008.

[24] GUTTMAN, Michael. PARODI, John. Real-life MDA: Solving Business Problems with Model Driven Architecture, 2007.

[25] MDA Versión Guía 1.0.1, 2001. Disponible en: <http://www.omg.org/mda>
Última visita: Marzo de 2009.

[26] VALERIO, Adrián. MDA: Reusabilidad Orientada al Negocio, 2006. Disponible en: <http://es.wikipedia.org/mda>. Última visita: Marzo de 2009.

[27] BROWN, Alan. An introduction to Model Driven Architecture: MDA and today's systems, 2004. Disponible en: <http://es.wikipedia.org/mda> Última visita: Abril/2009.

[28] RODRIGUEZ VICENTE, Jesús. Ingeniería de modelos con MDA, Facultad de Informática de la Universidad de Murcia, 2004. Disponible en: <http://www.gisum.uma.es> Última visita: Marzo de 2009.

[29] SCHMULLER, Joseph. Aprendiendo UML en 24 horas. Prentice Hall, 1997.

[30] LETERIER TORREZ, Patricio. Desarrollo de software orientado a objetos usando UML. Universidad Politécnica de Valencia, 2007. Disponible en: <http://www.dsic.upv.es/um1> Última visita: Abril/2009.

[31] Tutorial de UML. Disponible en: <http://www.sparxsystems.com/um1tutorial>. Última visita: Marzo de 2009.

[32] Guía de usuario de Enterprise Architect 7.0. Disponible en: <http://www.sparxsystems.com.ar/download/ayuda/index.html> . Última visita: Abril/2009.

[33] OLIVA ALONSO, Nuria. Sistemas de Cableado Estructurado. Alfaomega Editor, 2007.

[34] HUIDOBRO MOYA, José Manuel. MILLAN TEJEDOR, Ramón Jesús. Tecnologías de Telecomunicaciones. Alfaomega Editor, 2008.

[35] MARTINEZ OBREGÓN, Jesús. Diseño y administración de redes. Atlantic International University, 2008. Disponible en: <http://html.rincondelvago.com/redes-computacionales.html>. Última visita: Abril de 2009.

[36] DIAZ, Edward. COVA, Rubén. Diseño de una Red Lan para el Edificio de Cursos Básicos de la Universidad de Oriente Núcleo de Sucre, 2006. Disponible en: <http://www.monografias.com>. Última visita: Abril de 2009.

ANEXO A
PLANTILLA PARA LA ESPECIFICACIÓN DE REQUISITOS DE SISTEMAS TICS
UTILIZANDO MDA

[Escribir título del proyecto]

Autores:

[Nombre de los autores del documento]

Dirigido a:

[Cliente del proyecto]

Revisado Por:

[Escriba el rol y nombre de quien revisa]

Lugar y fecha de emisión:

[Ciudad y fecha de elaboración del documento]

CONTENIDO

Primera parte.

1. Identificación.....	75
2. Introducción.....	77
3. Propósito del documento.....	77
4. Definiciones, acrónimos y abreviaturas.....	77
5. Funcionalidad general del sistema.....	77
6. Especificación de Requisitos.....	77
6.1 Requisitos funcionales.....	77
6.2 Requisitos No funcionales.....	77
6.3 Requisitos de Interfaz.....	78
7. Estándares aplicables.....	78
8. Cambios.....	78
9. Información adicional.....	78
10. Acta de aprobación.....	78

Segunda parte.

11. Clases.....	79
11.1 Especificación de las clases.....	79
11.2 Diagrama modelado de clases.....	79

Tercera parte

12. Transformaciones MDA.....	79
-------------------------------	----

Primera parte.

2. Introducción

[Inserte aquí el texto]

3. Propósito del documento

[Inserte aquí el texto]

4. Definiciones, acrónimos y abreviaturas

TÉRMINO	DESCRIPCIÓN

5. Funcionalidad general del sistema

[Inserte aquí el texto]

6. Especificación de Requisitos

6.1. Requisitos funcionales

CÓDIGO	
NOMBRE	
DESCRIPCIÓN O CARACTERÍSTICAS	
TIPO	

6.2 Requisitos No funcionales

CÓDIGO	
NOMBRE	
DESCRIPCIÓN O CARACTERÍSTICAS	
TIPO	

6.3 Requisitos de Interfaz

CÓDIGO	
NOMBRE	
DESCRIPCIÓN O CARACTERÍSTICAS	
TIPO	

7. Estándares aplicables

ESTÁNDAR	DESCRIPCIÓN

8. Cambios

FICHA DE CONTROL DE CAMBIOS

FECHA	VERSIÓN	DESCRIPCIÓN DEL CAMBIO

9. Información adicional

[Inserte aquí el texto]

10. Acta de aprobación

ACTA DE APROBACIÓN	
CLIENTE	
FECHA	
NOMRE DE QUIEN APRUEBA _____	NOMRE DE QUIEN ENTREGA _____
FIRMA DE QUIEN APRUEBA _____	FIRMA DE QUIEN ENTREGA _____

Segunda parte.

11. Clases

11.1 Especificación de las clases

CÓDIGO	
NOMBRE	
DESCRIPCIÓN	
REQUISITO ASOCIADO	

11.2. Diagrama de modelado de clases

[Inserte aquí el diagrama]

Tercera parte.

12. Transformaciones MDA

[Inserte aquí el diagrama de la transformación]

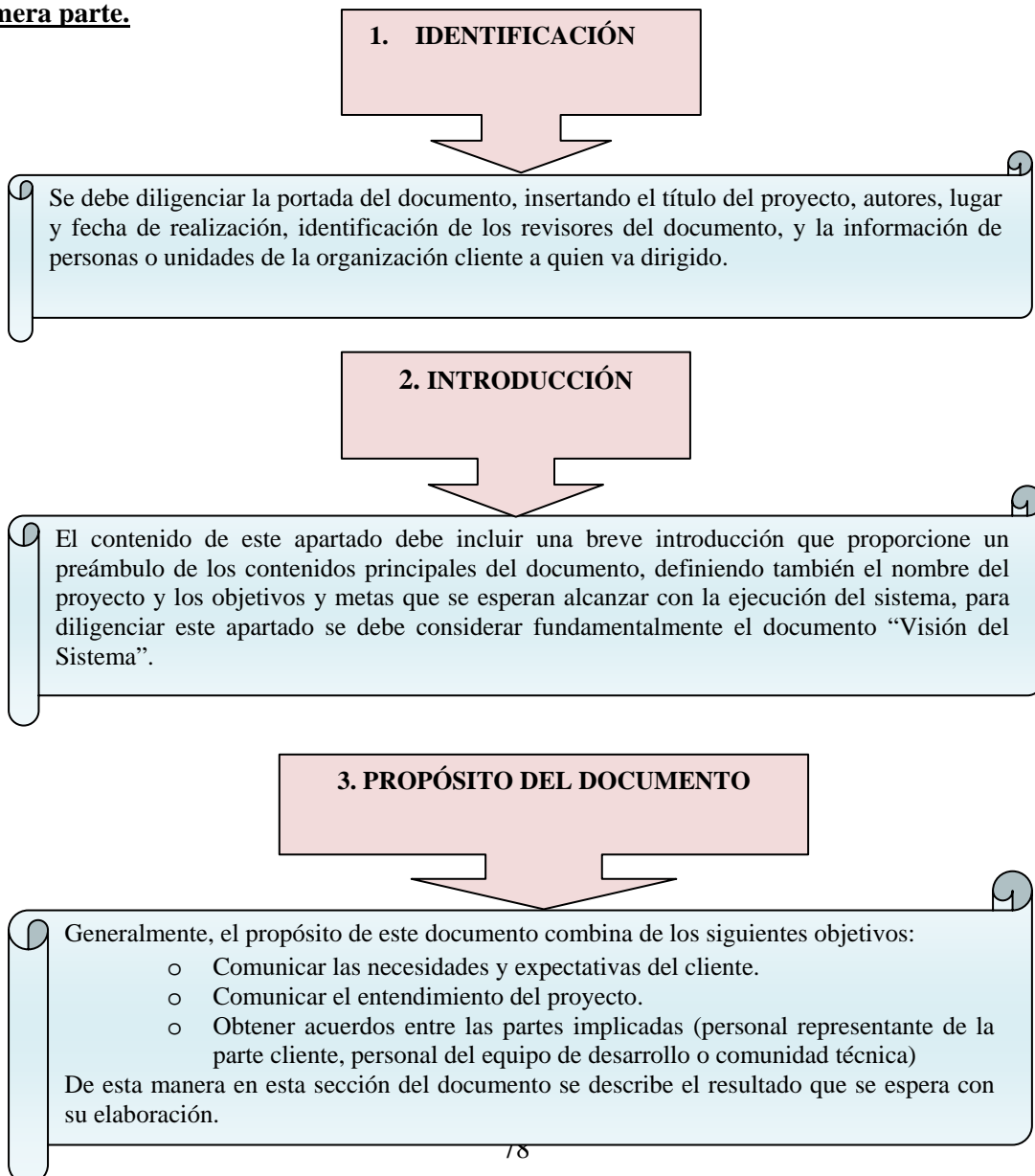
ANEXO B

GUÍA PARA EL USO DE LA PLANTILLA

Luego de describir el sistema en el documento “Visión del sistema” [apartado 2.3.1.1], se realiza la fase de especificación de requisitos mediante la plantilla propuesta en el anexo A, para diligenciar esta plantilla, se deben seguir las pautas descritas en la presente guía de usuario.

Las secciones que no se consideren aplicables al sistema descrito podrán indicarse como no aplicables (NA).

Primera parte.



4. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

En este apartado se debe mostrar las definiciones de todos los términos, siglas y abreviaciones requeridas para entender la especificación de requisitos, organizadas en la tabla presentada en la plantilla.

5. FUNCIONALIDAD GENERAL DEL SISTEMA

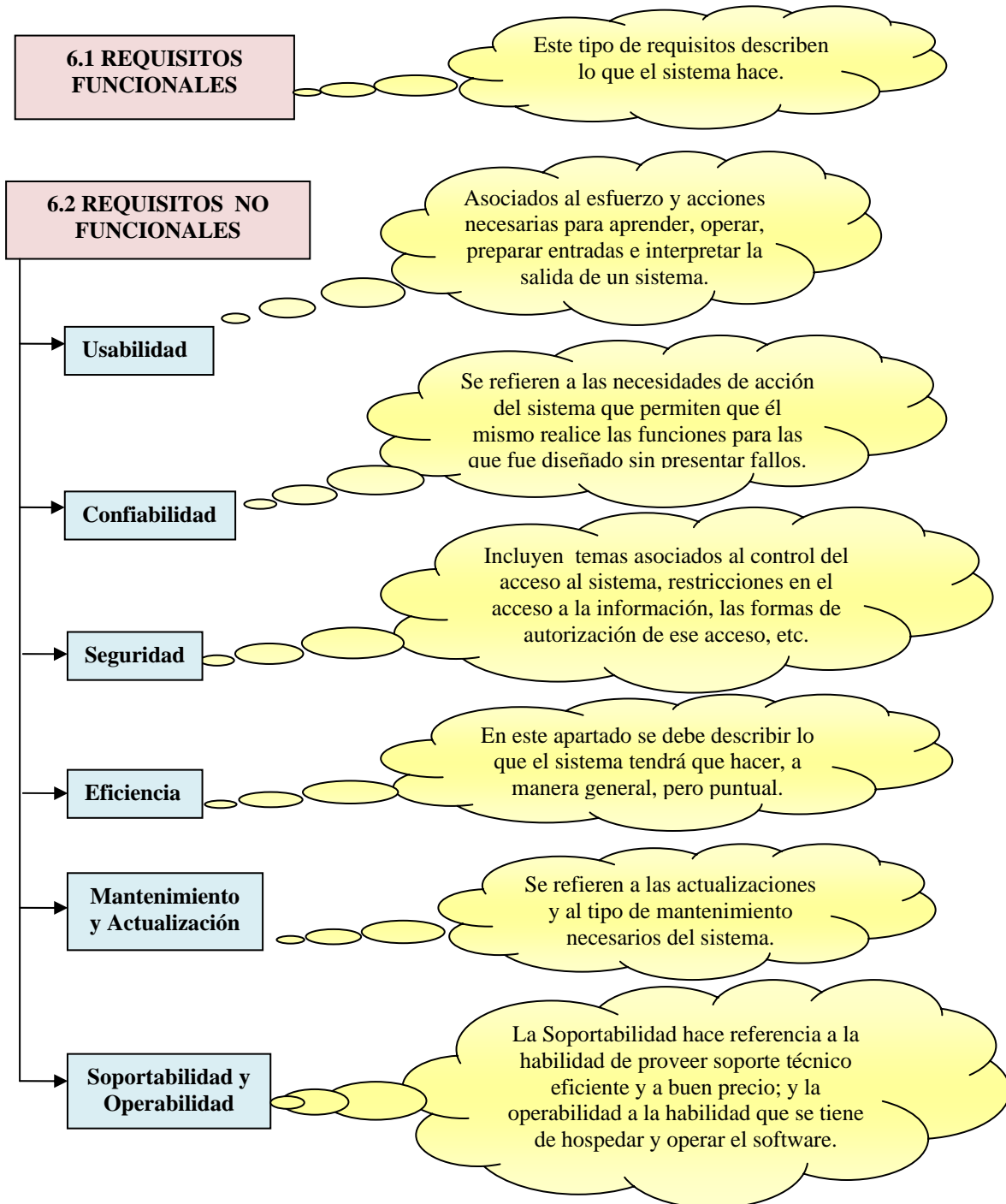
Se definirá a manera general la funcionalidad que tendrá el sistema, es decir, lo que hará y también lo que no hará. Se debe hacer un diagrama global de las partes hasta ahora definidas del sistema, que permita abstraer la funcionalidad del mismo, reflejando el sistema como un todo y la interacción más importante entre cada uno de sus elementos. Es importante que la funcionalidad del sistema sea lo más simple y clara posible, para que todos los lectores del documento puedan entenderla completamente.

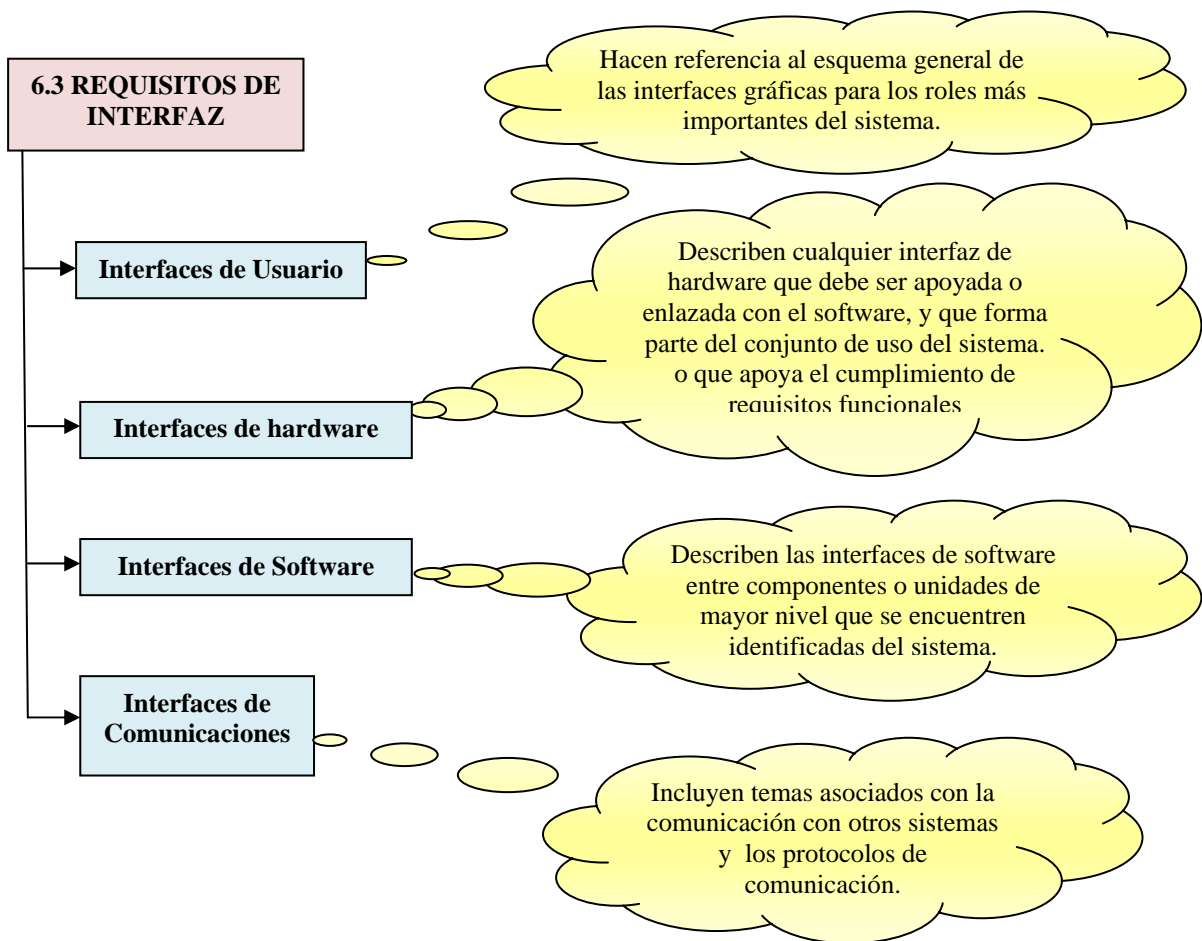
6. ESPECIFICACIÓN DE REQUISITOS

La tabla que presenta en la plantilla se debe diligenciar de la siguiente manera:

- ✓ **código:** se escribe el código de identificación del requisito, conformado por la partícula REQ y un número consecutivo de cifras correspondiente al número del requisito.
- ✓ **Nombre:** se le asigna un nombre al requisito, el cual debe estar conformado por las palabras claves que definan claramente la acción que logra el requisito en el sistema.
- ✓ **Descripción o características:** se explican las acciones que se logran a través del requisito y las condiciones de realización de la misma. Las características a contemplar a través de los diferentes requisitos son aquellas que fueron previamente definidas en el documento “Visión del Sistema”.
Esta descripción se debe realizar en suficiente detalle, tal que suministre la información necesaria que sirva de ayuda para la extracción de las clases del sistema.
- ✓ **Tipo:** se especifica si el requisito es funcional, no funcional o de interfaz.

A continuación se presenta un resumen de la clasificación de requisitos, de manera que puedan ser identificados fácilmente en el sistema de estudio.





7. ESTÁNDARES APLICABLES

En este apartado se debe describir cualquier estándar que se aplique al sistema, estos pueden ser: estándares de calidad, aspectos legales, interoperabilidad, internacionalización, estándares de seguridad de la información, compatibilidad del sistema operativo, estándares de programación, de producción o gestión, etc.

8. CAMBIOS

Esta sección contiene las carencias, defectos o debilidades del sistema que motivan al desarrollo de un nuevo, o a las modificaciones del existente.
Se pueden presentar cambios en la capacidad, en el proceso del sistema, en las interfaces, el personal, cambios de entorno, operaciones y soporte.
La descripción de los cambios deseados se debe consignar en la ficha de control de cambios que aparece en la plantilla, y se debe especificar la versión del documento.

9. INFORMACIÓN ADICIONAL

Se debe consignar cualquier información adicional que facilite la comprensión del documento.

10. ACTA DE APROBACIÓN

Finalmente se debe diligenciar el acta de revisión/aprobación cuando se lleve a cabo la validación de los requisitos descritos en el presente documento.
En el acta se consignarán los datos de identificación: nombre del cliente, nombre de quien revisa o aprueba y la fecha; y los nombres y firmas de quienes aprueban y entregan la especificación de los requisitos.

Segunda parte.

11. CLASES

Este ítem y los que siguen se realizan siempre y cuando sea aprobada el acta del apartado anterior, es decir la primera parte del procedimiento.
Una clase es una descripción de un conjunto de objetos con las mismas características, por tanto en esta sección las clases se deberán nombrar como un sustantivo extraído del análisis de requisitos y se deben organizar en la tabla que se muestra en la especificación de las clases.

11.1 ESPECIFICACIÓN DE LAS CLASES

Una clase es una descripción de un conjunto de objetos con las mismas características, por tanto en esta sección las clases se deberán nombrar como un sustantivo extraído del análisis de requisitos y se deben organizar en la tabla que se muestra en la plantilla, diligenciándola de la siguiente manera:

- ✓ **código:** se escribe el código de identificación de la clase, conformado por la partícula CL y un número consecutivo de cifras correspondiente al número de la clase.
- ✓ **Nombre:** se le asigna un nombre a la clase, el cual debe ser un sustantivo extraído del requisito.
- ✓ **Descripción:** se refleja el comportamiento de la clase en el sistema, mediante la explicación de los atributos (propiedades) y procesos (acciones) que tiene dicha clase.
- ✓ **Requisito relacionado:** se nombra el requisito que representa la dependencia con la clase.

11.2 DIAGRAMA DE MODELADO DE CLASES

El diagrama de clases modela la vista estática del sistema, teniendo bien definidas las clases se realiza el diagrama utilizando la herramienta de modelado Enterprise Architect.

Tercera parte.

12. TRANSFORMACIONES MDA

El diagrama de clases representa el PIM del sistema, a partir de este y como procedimiento final se realizan las transformaciones a PSM con EA.

ANEXO C

UML CON ENTERPRISE ARCHITECT

El siguiente resumen es basado en la guía de usuario de Enterprise Architect disponible en <http://www.sparxsystems.com.ar/download/ayuda/index.html>

Elementos de UML

Los modelos se construyen en UML a partir de elementos, cada elemento tiene un propósito, una notación y regla diferente. Estos elementos se crean y se muestran en los diferentes diagramas de modelado.

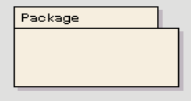
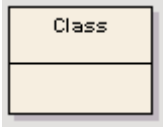
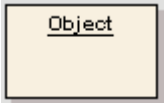
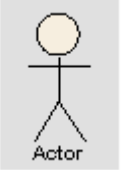
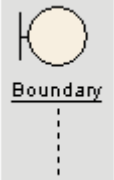


Los elementos se pueden extender casi sin límite utilizando estereotipos²⁶, generalmente un estereotipo representa una diferencia de uso, existen con la misma forma de atributos y relaciones, pero con una intención diferente.





El estereotipo generalmente se muestra como el siguiente esquema, donde «enumeración» es el estereotipo.



La siguiente tabla muestra los elementos principales del modelado de UML que se involucran en el desarrollo de este trabajo, su representación en EA y sus modos de usos.



²⁶ UML soporta un gran número de estereotipos, que son mecanismos propios para extender o alterar el significado, la presentación y la sintaxis de un elemento de modelado. Los diferentes elementos de modelado poseen distintos estereotipos asociados a ellos.

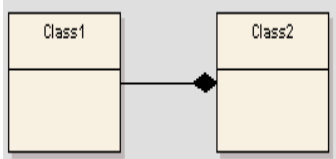
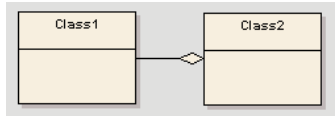
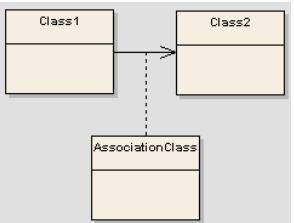
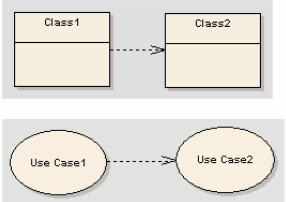
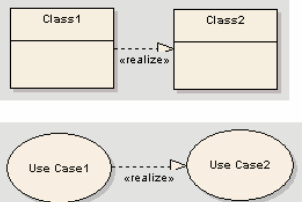
<i>Elemento</i>	<i>Representación</i>	<i>Modo de uso</i>	<i>Definición</i>
PAQUETE		Diagramas de clase Diagramas de objetos Diagramas de despliegue Diagramas de paquete	Un paquete es un elemento especial que puede contener otros elementos.
CLASE		Diagramas de clase Diagramas de estructura compuesta Diagramas de paquete	Una clase es una representación de uno o más objetos, que refleja la estructura y el comportamiento de un sistema, puede tener atributos y operaciones. Los modelos de Clases describen la estructura lógica del sistema.
OBJETO		Diagramas de clase Diagramas de objetos Diagramas de componente Diagramas de paquete	Un objeto representa una instancia particular de una clase en tiempo de ejecución. Los objetos se pueden utilizar para capturar rápidamente todas las cosas que son de relevancia en el dominio del sistema.
ACTOR		Diagramas de secuencia Diagramas de comunicación Diagramas de casos de uso	Un actor es un usuario del sistema; los actores se asocian normalmente con los casos de uso. La interacción de un actor se representa en los diagramas de casos de uso y detalla las funciones que el sistema debe proveer para satisfacer los requisitos del usuario.
LIMITE		Diagramas de clase Diagramas de secuencia Diagramas de comunicación	Un límite es una clase estereotipada que modela algunos límites del sistema, normalmente una pantalla de interfaz de usuario. Se usa en la fase conceptual para capturar la interacción del usuario con el sistema a un nivel de pantalla (o algún otro tipo de interfaz de límite).
CONTROL		Diagramas de clase Diagramas de secuencia Diagramas de comunicación	Un control es un tipo de clase que se estereotipa para representar a una entidad controladora, es decir organiza y programa otras actividades y elementos
ENTIDAD		Diagramas de clase Diagramas de secuencia Diagramas de comunicación	Una entidad es un mecanismo de pertenencia que captura la información o el conocimiento en un sistema.

INICIAL		Diagramas de actividad Diagramas de descripción de la interacción Diagramas de maquinas de estado	Este elemento es utilizado por los diagramas de actividades y de estados para definir el comienzo de un flujo cuando se invoca una actividad.
FINAL		Diagramas de actividad Diagramas de descripción de la interacción Diagramas de maquinas de estado	El elemento final indica que una actividad se ha completado.
ACTIVIDAD		Diagramas de actividad	Una actividad organiza y especifica la participación de elementos en acciones o subactividades para reflejar el control y el flujo de datos de un proceso.
CASO DE USO		Diagramas de casos de uso	Es la especificación de una secuencia de acciones, incluyendo variantes, que puede llevar a cabo un sistema, interactuando con los actores.

Conectores de UML

Esta tabla muestra los principales conectores del modelado de UML que se utilizan para unir los diagramas involucrados en el desarrollo de este trabajo, así como también su representación en EA y sus modos de usos

Conector	Representación	Modo de uso	Definición
ASOCIACIÓN		Diagramas de clase Diagramas de objetos Diagramas de paquete	Este tipo de conector se usa para representar gráficamente una asociación sencilla de elementos, puede incluir roles específicos en cada extremo como cardinalidad, dirección y restricciones.
GENERALIZACIÓN		Diagramas de clase Diagramas de objetos de componente Diagramas de despliegue Diagramas de paquete	Se usa una generalización para indicar herencia. Dibujada desde el clasificador específico al clasificador general, cada instancia del clasificador específico también es una instancia indirecta del clasificador general. Así, el clasificador específico hereda las características

			del clasificador más general.
COMPOCICIÓN		Diagramas de clase Diagramas de objetos Diagramas de paquete	Una agregación de composición sirve para representar que un elemento que está compuesto de otros más pequeños. Si se suprime elemento padre también lo hacen los demás.
AGREGACIÓN		Diagramas de clase Diagramas de objetos Diagramas de paquete	Una agregación muestra que un elemento contiene o está compuesto de otros elementos, se usa en los modelos de clases para mostrar que los elementos más complejos se constituyen de una colección de elementos más simples.
CLASE ASOCIACION		Diagramas de clase Diagramas de objetos Diagramas de despliegue Diagramas de paquete	Una clase asociación permite que una asociación tenga atributos y operaciones, viéndose como una asociación que también tiene características de clase, o como una clase que también tiene características de asociación.
DEPENDENCIA		Diagramas de clase Diagramas de objetos de componente Diagramas de despliegue Diagramas de paquete	Una dependencia es una relación que significa que un elemento de modelado simple o un conjunto de ellos requieren otros elementos de modelado para su especificación o implementación; tiene muchas derivaciones, tales como Realización, Instanciación y Uso.
REALIZAR		Diagramas de clase Diagramas de objetos de componente Diagramas de despliegue Diagramas de paquete	Una Realización significa que el conjunto de elementos cliente son una implementación del conjunto proveedor.

