

UNIVERSIDAD DE ORIENTE.
NÚCLEO DE ANZOÁTEGUI.
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS.
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS.



**DISEÑO DE UN SISTEMA DE INFORMACIÓN QUE PERMITA
LA AUTOMATIZACIÓN DE LAS ACTIVIDADES
RELACIONADAS CON EL MANTENIMIENTO A EQUIPOS
UTILIZADOS EN LÍNEAS DE PRODUCCIÓN EN UNA PLANTA
ENSAMBLADORA DE VEHÍCULOS.**

REALIZADO POR:

Mayerlin Carolina Benevento Trinchese

Carmen Virginia Sánchez Aguilar

Trabajo de Grado presentado como requisito parcial para optar al Título de
Ingeniero de Sistemas.

BARCELONA, FEBRERO DE 2009.

UNIVERSIDAD DE ORIENTE.
NÚCLEO DE ANZOÁTEGUI.
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS.
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS.



**DISEÑO DE UN SISTEMA DE INFORMACIÓN QUE PERMITA
LA AUTOMATIZACIÓN DE LAS ACTIVIDADES
RELACIONADAS CON EL MANTENIMIENTO A EQUIPOS
UTILIZADOS EN LÍNEAS DE PRODUCCIÓN EN UNA PLANTA
ENSAMBLADORA DE VEHÍCULOS.**

REVISADO Y APROBADO POR:

Ing. Zulirais García
Asesor Académico

BARCELONA, FEBRERO DE 2009.

UNIVERSIDAD DE ORIENTE.
NÚCLEO DE ANZOÁTEGUI.
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS.
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS.



**DISEÑO DE UN SISTEMA DE INFORMACIÓN QUE PERMITA
LA AUTOMATIZACIÓN DE LAS ACTIVIDADES
RELACIONADAS CON EL MANTENIMIENTO A EQUIPOS
UTILIZADOS EN LÍNEAS DE PRODUCCIÓN EN UNA PLANTA
ENSAMBLADORA DE VEHÍCULOS.**

JURADO CALIFICADOR:

EXCELENTE

Ing. Zulirais García
Asesor Académico

Ing. Manuel Carrasquero
Jurado principal

Ing. Héctor Moisés
Jurado principal

BARCELONA, FEBRERO DE 2009.

RESOLUCIÓN

De acuerdo al Artículo 44 del Reglamento de Trabajos de Grados de la Universidad de Oriente:

“Los trabajos de grado son de exclusiva propiedad de la universidad y solo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al consejo universitario”.

DEDICATORIAS

A Dios Todopoderoso, por darme la vida, por haberme permitido llegar hasta este punto y haberme dado salud, amor y fortaleza para lograr mis objetivos.

A mi abuela Teresa, por haber sido tan especial conmigo, por todo su amor y su enseñanza.

A mi madre Eleonora, por quererme y apoyarme en toda mi carrera. A mi hermanita Elianny, por su cariño y comprensión. Espero que para ti este sea un ejemplo a seguir, y que sepas que este triunfo es de las dos.

A mi tía Teresita por ser tan especial, por haberme apoyado en todo momento, por sus consejos y sus valores que me ha permitido ser una persona de bien, pero más que nada, por su amor y comprensión. A mis tíos, José, Eliécer, Maryori, María Teresa y Rosa. A mis primos, que me resulta muy difícil poder nombrarlos en tan poco espacio y a los cuales quiero muchísimo. En especial a mi prima y amiga Vane la cual quiero demasiado y la que siempre ha compartido todas las etapas de mi vida.

A mi gran amiga Eglisauris Zapata, por compartir momentos inolvidables, llenos de experiencias positivas y negativas pero que nos han hecho crecer y fortalecer nuestra amistad.

A mis amigos y compañeros, que nos apoyamos mutuamente en nuestra formación profesional y que hasta ahora, seguimos siendo amigos: Elías Mata, Sonia Bastardo, Nohemí Chirínos, Marianny Idrogo, Alexis Ramos, Carlos Mata, Virginia Bolívar.

Mayerlin Carolina Benevento Trinchese

DEDICATORIAS

A dios todo poderoso, por darme la vida y permitir el logro de mis metas, gracias DIOS.

A mi madre y a mi padre, quienes inculcaron en mí las ganas de avanzar en mis estudios y luchar por conseguir todas las metas propuestas, por su amor y por el apoyo brindado.

Carmen Virginia Sánchez Aguilar

AGRADECIMIENTOS

A Dios, infinitas gracias por haberme dado la sabiduría y el entendimiento para poder llegar al final de mi carrera, por proveerme de todo lo necesario para salir adelante y por todo lo que me has dado.

A mi abuela Teresa, por su excelente crianza, gracias a ti soy la persona que soy hoy aunque en estos momentos por tu enfermedad ya no recuerdes, igual este triunfo es para ti.

A mi madre Eleonora, por ser parte de mi formación y de este gran logro.

A mi hermana Elianny, por ser parte importante de mi vida y estar siempre presente.

A Ennodio Velásquez, quien fue de gran apoyo y ayuda en todo este trayecto y quien debo infinitos agradecimientos.

A mis amigos, José Elías Mata, Sonia Bastardo, Alexis Ramos, Nohemí Chirinos, Marianny Idrogo, Carlos Mata, Virginia Bolívar, que gracias al equipo que formamos logramos llegar hasta el final del camino. ¡Los aprecio Mucho!

A mi compañera de tesis Virginia Sánchez, por todo el tiempo compartido a lo largo de la carrera, por su comprensión y paciencia para superar tantos momentos difíciles en nuestro trabajo de grado.

A mi amiga de toda la vida Egli, por escucharme y apoyarme en todos los momentos de mi vida.

A todos mis familiares y amigos, Que de una u otra manera estuvieron pendientes a lo largo de este proceso, brindando su apoyo incondicional.

A mi asesora Zulirais García, por su valiosa orientación y excelente asesoría en el desarrollo de este proyecto.

A la Universidad de Oriente, que hizo posible a través de estos años mi formación académica profesional.

Gracias, mil gracias a todos, y muchos otros que se me escapan, siempre serán parte de este sueño hecho realidad.

Mayerlin Carolina Benevento Trinchese.

AGRADECIMIENTOS

A dios todo poderoso por darme la vida y permitirme poder cumplir mi meta tan anhelada, tener la opción de poder egresar como Ingeniero de Sistemas en la Universidad de Oriente, dándome la fortaleza necesaria para enfrentar todas las adversidades encontradas en el devenir del transcurso como estudiante en la tan gloriosa UDO, a la cual le agradezco por la formación integral obtenida durante el transcurso de mi carrera, mil gracias mi tan gloriosa UNIVERSIDAD DE ORIENTE!!!.

A mi madre, quien es el pilar de mi vida, mi sentido de ser, esa personita que me ha brindado su apoyo en todos y cada uno de mis proyectos, quien calma mi sed de vivir, gracias madre, mil veces gracias, eres la persona más importante de mi vida, sin ti no sería nadie, ya que gracias a tu amor, tu comprensión, tu apoyo, y sobre todo por tus sacrificios, en mi nombre y en el de mis hermanos, te agradezco por la paciencia que tuviste, y que seguro vas a seguir teniendo, por esas palabras de aliento cuando más lo necesitaba, y cualquier palabra no llega a cubrir el agradecimiento que siento hacia ti... eres el mejor ejemplo que la vida me ha brindado.

A mi padre, quien me inculco la mayoría de los valores que permitieron forjar mi carácter, y poder enfrentar con éxito los obstáculos que se presentaron en el transcurso de mi vida, aunque no estés cerca.

A mi familia, a mis hermanos, sobrinos, cuñados, y a mis tíos, primos, y abuelos, que aunque no se encuentren cerca, me han brindado su apoyo incondicional, lo cual me proporcionó el incentivo para no decaer y seguir adelante.

A mi novio Clodomiro Andrés Torres, quien me ha apoyado en todo momento, eres una de las pocas personas especiales que dios a cruzado en mí en camino, lo cual agradezco con todas las fuerzas de mi corazón, ya que día a día me ayudas a crecer

como persona, brindándome tu apoyo, tu comprensión, tu amor, dándome las fuerzas necesarias para seguir adelante, y por todo el apoyo brindado en el transcurso de mi carrera de Ingeniería de Sistemas, Te Amo mi amor... muchas gracias.

A mis amigos, Marianni (amiga como extraño esas amanecidas haciendo los proyectos, y sobre todo los desayunos a las 5am, TQM), Nohemi, Alexis, Antonio, Ricardo, Joanna, Puchito (Carlos), Anthony Paruta, Carmen Paruta, Nelson Sánchez, Sarina Cedeño, gracias por su apoyo y la compenetración en el grupo de estudio, ya que gracias a ello afrontamos con éxito todos los retos que se nos presentaron en la gran casa de estudios conocida como “La UDO”.

A mi amiga y compañera de Tesis Mayerlin Benevento, a quien agradezco su comprensión, su paciencia y su ayuda en el desarrollo del presente proyecto de investigación, gracias amiga.

A Ennodio Velásquez, quien nos facilitó los medios para obtener la base para el desarrollo del presente proyecto.

Y a todos aquellos que de una u otra forma ayudaron a cumplir esta meta, gracias de todo corazón a todos...

Carmen Virginia Sánchez Aguilar

RESUMEN

El presente proyecto de investigación consiste en el diseño de un sistema de información que permite la automatización de las actividades relacionadas con el mantenimiento a equipos utilizados en líneas de producción en una planta ensambladora de vehículos. Este trabajo fue realizado para satisfacer los requerimientos del departamento de mantenimiento de dicha ensambladora; con el principal objetivo de realizar un estudio de las actividades llevadas a cabo en el área, se logró el diseño de un sistema automatizado que permita procesar, almacenar y generar la información referente al mantenimiento de los equipos utilizados en las líneas de producción de la empresa. El proceso de diseño incluyó el levantamiento de información de interés, mediante entrevistas realizadas al personal de la empresa; así como el diseño de sus interfaces y de su base de datos. Todo esto se realizó cubriendo las fases de inicio y elaboración del Proceso Unificado de Desarrollo de Software, mediante la utilización del Lenguaje Unificado de Modelado (UML), en la cual se formó un concepto general del sistema y se llevo a cabo el análisis del negocio; finalmente se desarrollaron los diagramas correspondientes.

INDICE GENERAL

RESOLUCIÓN.....	IV
DEDICATORIAS.....	V
AGRADECIMIENTOS	VI
RESUMEN.....	X
INDICE GENERAL	XII

CAPITULO I: ASPECTOS GENERALES

1.1 PLANTEAMIENTO DEL PROBLEMA	1
1.2. OBJETIVOS	3
1.2.1. Objetivo General	3
1.2.2. Objetivos Específicos.....	4

CAPITULO II: MARCO TEÓRICO

2.1. ANTECEDENTES.....	5
2.2. MARCO TEORICO.....	8
2.2.1. Mantenimiento	8
2.2.1.1. Objetivos del Mantenimiento.....	8
2.2.1.2. Funciones del Mantenimiento.....	9
2.2.2. Sistemas de Información.....	10
2.2.2.1. Definición.....	10
2.2.2.2. Actividades que Realiza un Sistema de Información.....	11
2.2.2.3. Elementos que Componen a un Sistema de Información [11].....	12
2.2.3. Análisis y Diseño Orientado a Objetos	15
2.2.3.1. Definición.....	15
2.2.4. Programación Orientada a Objetos [10].....	16
2.2.4.1. Definición.....	16
2.2.4.2. Conceptos Fundamentales.....	16

2.2.4.3. Características de la POO.....	18
2.2.5. Proceso Unificado	20
2.2.5.1. Definición [15]	21
2.2.5.2. Principios Básicos del Proceso Unificado	21
2.2.5.3. Vida del Proceso Unificado	23
2.2.6. Lenguaje Unificado de Modelado (UML) [12].....	25
2.2.6.1. Definición [13]	25
2.2.6.2. Utilidad de UML	27
2.2.6.3. Bloques Básicos de UML	27
2.2.7. Base de Datos.....	43
2.2.7.1. Definición.....	43
2.2.7.2. Tipos de Base de Datos	44
2.2.7.3. Modelo de Bases de Datos [5].....	46
2.2.8. Sistema de Gestión de Base de Datos (SGBD) [6].....	50
2.2.8.1. Definición.....	50
2.2.8.2. Objetivos de SGBD.....	51
2.2.9. Modelo Relacional de Base de Datos.....	52
2.2.9.1. Definición [19]	52
2.2.9.2. Descripción [1].....	53
2.2.9.3. Base de Datos Relacional.....	55
2.2.9.4. Formas Normales	55
2.2.10 Interfaz de Usuario [7].....	56
2.2.10.1. Definición.....	56
2.2.10.2. Principales Funciones de las Interfaces de Usuario	57
2.2.10.3. Tipos de Interfaces de Usuario.....	57
2.2.11. Reportes	58

CAPITULO III: FASE DE INICIO

3.1. CONTEXTO DEL SISTEMA	60
3.1.1. Modelo de Dominio	62

3.1.2. Glosario de Términos.....	64
3.2. RIESGOS DEL SISTEMA	66
3.3. REQUERIMIENTOS DEL SISTEMA.....	68
3.3.1. Identificación de Actores	70
3.3.2. Casos de Uso.....	71
3.3.3. Modelo de Casos de Uso.....	73
3.4. ANÁLISIS	77
3.4.1. Análisis de los Casos de Uso	78
3.4.1.1. Identificación de las Clases de Análisis	78
3.4.1.2. Diagramas de Colaboración	89
3.4.2. Análisis de la Arquitectura.....	100
3.4.2.1. Identificación de los Paquetes de Análisis	101
3.5.DISEÑO	102
3.6.EVALUACIÓN DE LA FASE DE INICIO	103

CAPITULO IV: FASE DE ELABORACIÓN

4.1. ANALISIS	108
4.1.1. Diagrama de Clases de Diseño.....	108
4.1.1.1. Diseño de las Clases.....	110
4.1.2. Diagramas de Secuencia	114
4.1.3. Diagrama de Capas	152
4.2. DISEÑO.....	152
4.2.1. Base de Datos.....	153
4.2.1.1. Diseño de las Tablas	153
4.2.1.2. Modelo Relacional	161
4.2.2. Diseño de las Interfaces de Usuario	163
4.2.3. Diseño de los Reportes Impresos	175
4.3. EVALUACIÓN DE LA FASE DE ELABORACIÓN	178
CONCLUSIONES.....	152
RECOMENDACIONES.....	154

BIBLIOGRAFÍA.....	155
ANEXOS.....	¡Error! Marcador no definido.58

CAPITULO I

ASPECTOS GENERALES

1.1 PLANTEAMIENTO DEL PROBLEMA

La empresa MMC Automotriz, ubicada en la zona industrial los Montones de Barcelona, Estado Anzoátegui, cuenta con un área total de terreno de 116.223,98 m². El proceso principal que se lleva a cabo en dicha empresa es el ensamblaje de vehículos de las marcas Mitsubishi y Hyundai, y se encuentra operativa desde el tres de Agosto de 1990. Para el proceso de ensamblaje, la empresa cuenta principalmente con materia prima producida por proveedores foráneos (Japón y Corea), complementado con material suministrado por proveedores locales.

Existen seis líneas de producción, las cuales son: electropunto, latonería, pintura, vestidura, línea alta y línea final, y estas a su vez están conformadas por diversas zonas, en las que la fusión entre mano de obra capacitada y maquinaria especializada le agrega valor a la materia prima, para obtener como resultado un producto de un alto estándar de calidad en el mercado automotriz nacional.

La empresa debe velar por el buen estado y óptimo funcionamiento de los equipos especializados que intervienen en el proceso de ensamblaje, trabajo desempeñado por el Departamento de Mantenimiento de la misma, el cual se encarga de llevar a cabo labores de mantenimiento preventivo y correctivo de los equipos e instalaciones. La labor principal del Departamento de Mantenimiento es asegurar el normal funcionamiento de los equipos productivos y mantener la infraestructura adecuadamente, para conservar la capacidad de producción de la planta, con bajos costos y alta seguridad, y para facilitar este proceso cuenta con personal especializado



como electromecánicos, electricistas y otros, coordinados por los supervisores y gerentes del área.

Debido al alto nivel de producción que tiene la empresa se genera un gran volumen de información que es necesario analizar, procesar y almacenar, para acceder de manera eficaz a la misma, estas operaciones son realizadas de forma manual por personal del Departamento de Mantenimiento, consumiendo parte importante del recurso humano disponible. Las imprecisiones producto del error humano y los altos tiempos de respuesta que genera el procesar grandes volúmenes de data reduce el grado de respuesta que pueda ofrecer el área en situaciones de contingencia, que tienden a demorar las labores productivas de la empresa.

La aplicación de herramientas tecnológicas que permitan disminuir la ejecución de arduos procesos manuales, los cuales tienen considerables porcentajes de errores, y la reducción de actividades del mismo perfil, pueden dar paso a un método de trabajo que permita distribuir todos estos procesos de una forma más equitativa en el tiempo, lo que concluiría en un mejor engranaje en las operaciones relacionadas con el mantenimiento de los equipos utilizados en el proceso de ensamblaje de vehículos.

Por tal motivo, MMC Automotriz ha tenido la necesidad de fomentar el diseño de un Sistema de Información que permita automatizar las actividades relacionadas con el mantenimiento a equipos utilizados en líneas de producción en dicha planta ensambladora de vehículos, el cual permitirá principalmente manejar de manera eficaz y eficiente la información obtenida en el proceso bajo estudio.

Para realizar el diseño del Sistema de Información se aplicó la metodología de Proceso Unificado de Desarrollo de Software, el cual solo contempló las fases del diseño del sistema, abarcando la fase de inicio y la fase de elaboración, llevando a cabo los flujos de trabajo: requisitos, análisis y diseño, mediante la utilización del



Lenguaje Unificado de Modelado (UML). En la etapa de inicio, se procedió a levantar la información necesaria, para posteriormente analizarla. Luego, se especificaron los requerimientos del sistema, y se diseñaron las diferentes propuestas a fin de buscar mejoras al sistema actual. Finalmente, en la etapa de elaboración, se planteó la arquitectura del sistema propuesto, al igual que se diseñó la base de datos y una interfaz que sea de fácil manejo para los usuarios.

El proyecto será la base fundamental para la futura construcción del Sistema de Información, a partir de la arquitectura del software diseñada, lo cual permitirá controlar las actividades de mantenimiento realizadas a los equipos, evitando así demoras en el funcionamiento del sistema, que en la mayoría de los casos se traducen en resultados deficientes, optimizando de manera real el desempeño del mismo.

La originalidad de este proyecto radicó en el hecho que el Departamento de Mantenimiento de MMC Automotriz no cuenta con un Sistema de Información que permita automatizar las actividades relacionadas con el mantenimiento a equipos utilizados en líneas de producción, y por lo tanto, es la primera vez que se realiza una propuesta modelando este sistema.

1.2. OBJETIVOS

1.2.1. Objetivo General

Diseñar un Sistema de Información que permita la automatización de las actividades relacionadas con el mantenimiento a equipos utilizados en Líneas de Producción en una Planta Ensambladora de Vehículos.



1.2.2. Objetivos Específicos

- ◆ Identificar los requisitos del sistema de mantenimiento a equipos utilizados en líneas de producción.
- ◆ Diseñar el modelo de la situación actual del proceso de mantenimiento realizado a equipos utilizados en Líneas de Producción.
- ◆ Analizar los requerimientos recopilados.
- ◆ Diseñar la arquitectura del software.
- ◆ Diseñar la estructura lógica de la base de datos.
- ◆ Diseñar el interfaz de usuario del Sistema de información propuesto considerando los requerimientos de los usuarios finales.
- ◆ Diseñar formatos de salida y reportes del sistema de mantenimiento a equipos utilizados en líneas de producción.

CAPITULO II MARCO TEÓRICO

2.1. ANTECEDENTES

Para el desarrollo del presente trabajo de investigación se utilizó la Metodología del Proceso Unificado de Desarrollo de Software y el Lenguaje Unificado de Modelado (UML), como herramientas para diseñar un sistema de información que permita automatizar las actividades relacionadas con el mantenimiento a equipos utilizados en líneas de producción en planta ensambladora de vehículos. En la Universidad de Oriente, anteriormente se han llevado a cabo trabajos de investigación conexos a la realización de sistemas automatizados dentro de empresas, en los que se han utilizado la metodología antes mencionada para realizar dichos estudios. Es por ello que a continuación se mencionan algunos de los trabajos relevantes que han servido como guía para la realización de este proyecto de investigación:

- ◆ Mata, C., (2007). **“Diseño de un Sistema de Información de los Parámetros de Funcionamiento de un Servidor de Almacenamiento del Centro de Cómputo de PDVSA-Refinación Oriente”**. Trabajo de Grado presentado en la Universidad de Oriente, Núcleo de Anzoátegui, para optar al título de Ingeniero de Sistemas. El trabajo de investigación consistió en un diseño de un sistema de supervisión que permitiera principalmente: recolectar y almacenar información sobre el funcionamiento del equipo, visualizar las alarmas generadas por las fallas del dispositivo y representar el desempeño del servidor a través del tiempo. Para realizar el Diseño del Sistema de Supervisión, se aplicaron las fases de Inicio y Elaboración del Proceso Unificado de Desarrollo de Software, todo esto, llevando a cabo los flujos de



trabajo: requisitos, análisis y diseño, mediante la utilización del Lenguaje Unificado de Modelado (UML). [2]

- ◆ Rojas A. y Prado L., (2007). **“Diseño de un Sistema de Información para el Seguimiento de las Actividades Asociadas con la Elaboración de Presupuestos de una Empresa Dedicada a la Fabricación de Productos de Aluminio”**. Trabajo de Grado presentado en la Universidad de Oriente, Núcleo de Anzoátegui, para optar al título de Ingeniero de Sistemas. Este trabajo de investigación se realizó con la finalidad de solucionar la pérdida de información en el Departamento de Ventas, cuando se realizaban las tareas de elaboración de presupuestos, facturas y actualización de precios, ocasionando efectos negativos para la empresa. Para solucionar este problema se propuso un Sistema de Información con el propósito de que la información técnica y económica de la empresa estén siempre disponibles. Para el proceso de análisis y diseño se utilizó el Lenguaje Unificado de Modelado (UML), para modelar el sistema a través de los diagramas que posee esta herramienta. Por último, se utilizó Microsoft Access para diseñar la Base de Datos y Visual 6.0 para la interfaz de usuario. [3]
- ◆ Marcano F. y Acevedo Y., (2006). **“Diseño de un Software de Balance de Materiales para Yacimientos de Crudo y Gas, para PDVSA Oriente”**. Trabajo de Grado presentado en la Universidad de Oriente, Núcleo de Anzoátegui, para optar al título de Ingeniero de Sistemas. Este trabajo tubo como principal objetivo el de realizar un estudio de los procedimientos llevados a cabo en el área, para diseñar un sistema automatizado el cual permita procesar, almacenar y analizar mediante el balance de materiales, toda la información relacionada a las propiedades y características de los yacimientos de petróleo negro. Todo esto realizado utilizando el Lenguaje Unificado de Modelado (UML), cubriendo sólo las fases de inicio y



elaboración del Proceso Unificado de Desarrollo de Software, para la representación y conceptualización del sistema. [4]

- ◆ Sánchez M., (2005). **“Diseño de un Sistema de Información para Automatizar Algunas de las Actividades Relacionadas con el Proceso de Producción de Crudo y Gas desde el Yacimiento hasta las Estaciones de Flujo, que se Realizan en una Empresa Petrolera, en Punta de Mata”**. Trabajo de Grado presentado en la Universidad de Oriente, Núcleo de Anzoátegui, para optar al título de Ingeniero de Sistemas. El estudio se basó en un Sistema de Información a través de la Metodología del Proceso Unificado y el Modelo Racional para el diseño de la Base de Datos, todo con la finalidad de diseñar un sistema automatizado que permitiera procesar, almacenar y generar la información necesaria referente a las actividades, costos de pozos y estaciones de flujos de la empresa. El diseño incluyó técnicas de entrevistas al personal de la empresa para el levantamiento de la información; además del análisis de las Bases de Datos involucradas, para determinar el comportamiento del sistema actual y así elaborar el modelo del sistema propuesto. [8]
- ◆ Chirinos A., (2005). **“Diseño de un Sistema de Información para la Automatización de las Actividades Administrativas del Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo de Anzoátegui, Basado en un Estudio de Sistemas Blandos”**. Trabajo de Grado presentado en la Universidad de Oriente, Núcleo de Anzoátegui, para optar al título de Ingeniero de Sistemas. El desarrollo de este proyecto estuvo basado en un sistema de información que pudiera automatizar las actividades administrativas más relevantes que se realizaban en el departamento. Todo esto bajo un estudio realizado mediante la Metodología de Sistemas Suaves de



Peter Checkland, seguidamente se utilizó las herramientas de Lenguaje Unificado de Modelado (UML) para representar gráficamente el sistema propuesto y mostrar con más detalle su comportamiento. [9]

2.2. MARCO TEORICO

2.2.1. Mantenimiento

Existen varias versiones sobre el significado del mantenimiento, sin embargo, hay un consenso de que el mantenimiento, es el conjunto de acciones permanentes y sistemáticas, destinadas a conservar y alargar en lo posible la vida de los activos y garantizar el funcionamiento normal y eficiente de los mismos. El mantenimiento consiste en asegurar el funcionamiento eficiente y continuado de los ambientes, las instalaciones, los equipos, mediante la prevención, la conservación y el mejoramiento de los mismos, a fin de lograr una mayor vida útil, seguridad de operación y economía en costos. El mantenimiento llega a tener connotación, cuando se conserva en operación continua, confiable, segura, económica, la totalidad de las instalaciones, inmuebles y equipos que la institución tiene la prestación de servicios.

Por ello la importancia del mantenimiento, por asegurar la disponibilidad de los equipos y otros bienes para que la organización logre sus objetivos con un rendimiento óptimo de inversión.

2.2.1.1. Objetivos del Mantenimiento

El diseño e implementación de cualquier sistema organizativo y su posterior información debe siempre tener presente que está al servicio de unos determinados objetivos. Cualquier sofisticación del sistema debe estar contemplada con gran



prudencia en evitar, precisamente, de que se enmascaren dichos objetivos o que se dificulte su consecución.

En el caso del mantenimiento su organización e informatización debe estar encaminada a la permanente consecución de los siguientes objetivos:

- ◆ Optimización de la disponibilidad del equipo productivo.
- ◆ Disminución de los costos de mantenimiento.
- ◆ Optimización de los recursos humanos.
- ◆ Maximización de la vida útil de las maquinas.

2.2.1.2. Funciones del Mantenimiento

- ◆ Planificar, programar y ejecutar las actividades de mantenimiento.
- ◆ Instalar y controlar equipos y sistemas.
- ◆ Registrar, controlar y evaluar sus actividades.
- ◆ Registrar, controlar y evaluar fallas.
- ◆ Desarrollar nuevas tecnologías de mantenimiento.
- ◆ Asesorar al departamento de la adquisición de materiales y repuestos.
- ◆ Formación y adiestramiento de su personal.
- ◆ Garantizar la seguridad y eficiencia operacional de los equipos al costo mas bajo.
- ◆ Análisis del proceso de producción y de la organización de mantenimiento existente.
- ◆ Inventarios de los equipos e instalaciones que integran la planta.
- ◆ Requerimiento de mantenimiento y sus frecuencias.
- ◆ Inspecciones basadas en una programación definida.
- ◆ El factor humano.



- ◆ Mantener bien informado a todo el personal directivo involucrado en el problema.

2.2.2. Sistemas de Información

Cada organización es única, tiene su propia combinación exclusiva de hombres, recursos económicos, máquinas, materiales y métodos. Esta singularidad hace necesario que cada organización desarrolle sus propias especificaciones de su sistema de información para la administración, mediante una evaluación sistemática de su propio ambiente externo e interno y de su punto de vista, de acuerdo con sus propias necesidades únicas.

Se puede esperar que la tecnología de la información continúe su rápida evolución. Bajo estas circunstancias, cuando se lleva a cabo el desarrollo de un nuevo sistema, su diseño debe tener en cuenta los progresos que se esperan en funcionamiento y costo que es probable que ocurran durante su desarrollo y en las primeras fases de su uso. A este respecto, el diseño de sistemas de información es un proceso similar a apuntar a un blanco móvil. Si no se tienen en cuenta estas tendencias se invertirá dinero en sistema que ya será obsoleto cuando estén listos para su implementación.

2.2.2.1. Definición

Un sistema de información es una colección de personas, procedimientos y equipos diseñados, construido, operado y mantenido para coleccionar, registrar, procesar, almacenar, recuperar y mostrar información. [21]

Por lo tanto, los sistemas de información, se definen como procesos que se realizan sobre un conjunto de datos. Dichos procesos consisten en recogerlos, agruparlos, analizarlos y difundirlos con el fin de realizar actividades de control y



toma de decisiones. El objetivo del proceso de Diseño de un Sistema de Información es la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del Sistema de Información.

2.2.2.2. Actividades que Realiza un Sistema de Información

- ◆ **Entrada de información:** Es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.

- ◆ **Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

- ◆ **Procesamiento de información:** Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una



proyección financiera a partir de los datos que contiene un estado de resultado o un balance general de un año base.

- ◆ **Salida de información:** La salida es la capacidad de un sistema de información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro sistema de información o modulo.

2.2.2.3. Elementos que Componen a un Sistema de Información [11]

Un sistema de información (SI) es un conjunto organizado de elementos, estos elementos son de 4 tipos:

- ◆ Personas.
- ◆ Datos.
- ◆ Actividades o técnicas de trabajo.
- ◆ Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todo ese conjunto de elementos interactúan entre si para procesar los datos y la información (incluyendo procesos manuales y automáticos) y distribuirla de la manera más adecuada posible en una determinada organización en función de sus objetivos. Normalmente el término es usado de manera errónea como sinónimo de sistema de información informático, estos son el campo de estudio de la tecnología de la información (IT), y aunque puedan formar parte de un sistema de información (como recurso material), por sí solos no se pueden considerar como sistemas de información, este concepto es más amplio que el de sistema de información



informático. No obstante un sistema de información puede estar basado en el uso de computadoras, según la definición de Langefors este tipo de sistemas son:

- ◆ Un medio implementado tecnológicamente para grabar, almacenar y distribuir expresiones lingüísticas, así como para extraer conclusiones a partir de dichas expresiones.
- ◆ Sistema de procesamiento de transacciones (TPS): Gestiona la información referente a las transacciones producidas en una empresa u organización.
- ◆ Sistemas de información gerencial (MIS): Orientados a solucionar problemas empresariales en general.
- ◆ Sistemas de soporte a decisiones (DSS): Herramienta para realizar el análisis de las diferentes variables de negocio con la finalidad de apoyar el proceso de toma de decisiones.
- ◆ Sistemas de información ejecutiva (EIS): Herramienta orientada a usuarios de nivel gerencial, que permite monitorizar el estado de las variables de un área o unidad de la empresa a partir de información interna y externa a la misma.
- ◆ Sistemas de automatización de oficinas (OAS): Aplicaciones destinadas a ayudar al trabajo diario del administrativo de una empresa u organización.
- ◆ Sistema experto (SE): Emulan el comportamiento de un experto en un dominio concreto.
- ◆ Sistema Planificación de Recursos (ERP): Integran la información y los procesos de una organización en un solo sistema.

En la **Figura 2.1** se muestran los elementos básicos que componen a un sistema de información.

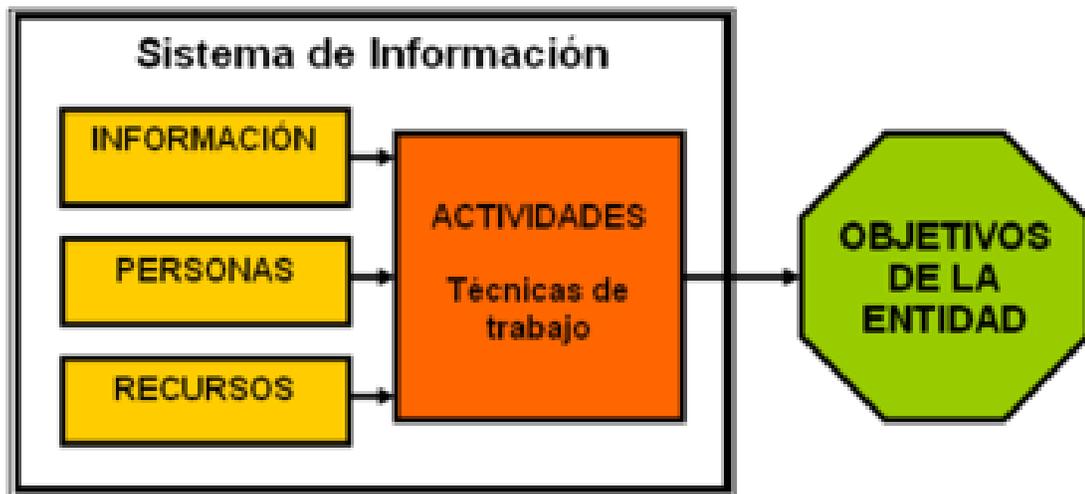


Figura 2.1. Elementos de un Sistema de Información.

Fuente: Ciborra, 2002.

Estos sistemas de información no surgieron simultáneamente en el mercado; los primeros en aparecer fueron los TPS, en la década de los 60, y los últimos fueron los SE, que alcanzaron su auge en los 90 (aunque estos últimos tuvieron una tímida aparición en los 70 que no cuajó, ya que la tecnología no estaba suficientemente desarrollada).

Otra clasificación, según el entorno de aplicación:

- ◆ Entorno transaccional: Una transacción es un suceso o evento que crea/modifica los datos. El procesamiento de transacciones consiste en captar, manipular y almacenar los datos, y también, en la preparación de documentos; en el entorno transaccional, por tanto, lo importante es qué datos se modifican y cómo, una vez ha terminado la transacción. Los TPS son los SI típicos que se pueden encontrar en este entorno.
- ◆ Entorno decisional: Este es el entorno en el que tiene lugar la toma de decisiones; en una empresa, las decisiones se toman a todos los niveles y en



todas las áreas (otra cosa es si esas decisiones son estructuradas o no), por lo que todos los SI de la organización deben estar preparados para asistir en esta tarea, aunque típicamente, son los DSS los que encargan de esta función. Si el único SI de una compañía preparado para ayudar a la toma de decisiones es el DSS, éste debe estar adaptado a todos los niveles jerárquicos de la empresa.

En la **Figura 2.2** se puede visualizar un esquema en donde se representa la clasificación de los sistemas de información.

2.2.3. Análisis y Diseño Orientado a Objetos

2.2.3.1. Definición

Es un enfoque de la ingeniería de software que modela un sistema como un grupo de objetos que interactúan entre sí. Este enfoque representa un dominio en términos de conceptos compuestos por verbos y sustantivos, clasificados de acuerdo a su dependencia funcional.

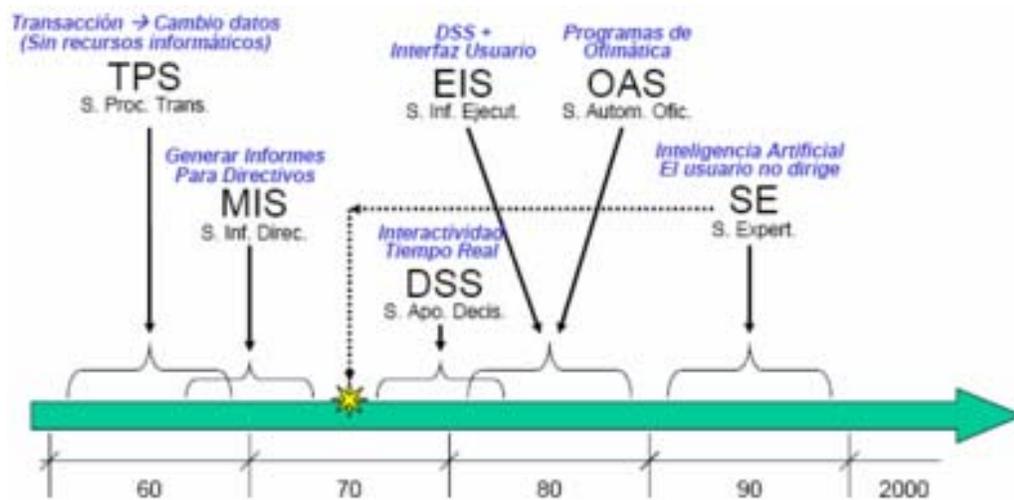


Figura 2.2. Clasificación de los Sistemas de Información.

Fuente: Ciborra, 2002.



En éste método de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada como, por ejemplo, el lenguaje unificado de modelado (UML). ADOO aplica técnicas de modelado de objetos para analizar los requerimientos para un contexto por ejemplo, un sistema de negocio, un conjunto de módulos de software - y para diseñar una solución para mejorar los procesos involucrados. No está restringido al diseño de programas de computadora, sino que cubre sistemas enteros de distinto tipo. Las metodologías de análisis y diseño más modernas son casos de uso guiados a través de requerimientos, diseño, implementación, pruebas, y despliegue.

El lenguaje unificado de modelado se ha vuelto el lenguaje de modelado estándar usado en análisis y diseño orientado a objetos.

2.2.4. Programación Orientada a Objetos [10]

2.2.4.1. Definición

La Programación Orientada a Objetos (POO u OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento. Su uso se popularizó a principios de la década de 1990. Actualmente son muchos los lenguajes de programación que soportan la orientación a objetos.

2.2.4.2. Conceptos Fundamentales

La programación orientada a objetos es una nueva forma de programar que trata de encontrar una solución a estos problemas. Introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:



- ◆ **Clase:** definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas, (de c a d), Es la facilidad mediante la cual la clase D ha definido en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D.
- ◆ **Objeto:** entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos). Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.
- ◆ **Método:** algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.
- ◆ **Evento:** un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.
- ◆ **Mensaje:** una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- ◆ **Propiedad o atributo:** contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y



esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

- ◆ **Estado interno:** es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.
- ◆ **Componentes de un objeto:** atributos, identidad, relaciones y métodos.
- ◆ **Representación de un objeto:** un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes.

2.2.4.3. Características de la POO

Hay un cierto desacuerdo sobre exactamente qué características de un método de programación o lenguaje le definen como "orientado a objetos", pero hay un consenso general en que las características siguientes son las más importantes:

- ◆ **Abstracción:** Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar *cómo* se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.
- ◆ **Encapsulamiento:** Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del



sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

- ◆ **Principio de ocultación:** Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una *interfaz* a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- ◆ **Polimorfismo:** comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.
- ◆ **Herencia:** las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia



organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en *clases* y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.

2.2.5. Proceso Unificado

Un proceso define quien está haciendo Qué, Cuándo y Cómo alcanzar un determinado objetivo. En la Ingeniería de Software el objetivo es construir un producto de software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad, captura y presenta las mejoras prácticas que el estado actual de la tecnología permite. En consecuencia, reduce el riesgo y hace el proyecto más predecible. [15]

Dentro del proceso unificado estas preguntas son presentadas en cuatro elementos:

- ◆ Trabajadores: El quién

Los trabajadores definen los puestos que las personas pueden optar, es decir, un trabajador es un papel que un individuo puede desempeñar durante el desarrollo del software.

Cada trabajador tiene un conjunto de responsabilidades y lleva a cabo un conjunto de actividades en desarrollo del software. Un trabajador puede representar a un conjunto de personas que trabajan juntas.



- ◆ **Actividades: El cómo**

Una actividad es una parte del trabajo que un trabajador realiza, produciendo con ello un trabajo significativo, lo que representa una unidad con límites bien definidos para facilitar la asignación de tareas.

- ◆ **Artefactos: El qué**

Un artefacto es una pieza de información que es producida, modificada o utilizada en un proceso, en sí, es cualquier tipo de información que los trabajadores pueden usar.

- ◆ **Flujos de Trabajo: El cuándo**

Un flujo de trabajo es un conjunto de actividades, y es el modo en que describe un proceso desarrollo. Ya se han mencionado los cinco flujos principales del proceso unificado.

2.2.5.1. Definición ^[15]

El Proceso Unificado (PU) es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el proceso unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños del proyecto.

El proceso unificado está basado en componentes lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

2.2.5.2. Principios Básicos del Proceso Unificado



◆ **Proceso Unificado Dirigido por Casos de Uso**

Al estar dirigido por casos de usos significa que el proceso se centra en lo que debe hacer el sistema, comenzando por especificar sus requisitos. Los casos de uso representan los requisitos funcionales el sistema y el conjunto de todos estos casos de usos forman por lo que se conoce como modelos de casos de usos, el cual contiene la descripción de todo el sistema. Los casos de usos además de ayudar a especificar los requisitos, sirven de guía durante todo el proceso, ya que se toman como base para el diseño, la implementación y las pruebas.

◆ **Un Proceso Centrado en la Arquitectura**

Al estar centrado en la arquitectura, el PU describe el sistema en varios puntos de vista. La arquitectura es la forma que tendrá el sistema y sirve como base para comprender como quedara terminado. Se utiliza como artefacto básico para conceptualizar, construir, gestionar y hacer evolucionar el sistema en desarrollo.

◆ **El Proceso es Iterativo e Incremental**

Un proceso iterativo es aquel que involucra la gestión de un flujo de versiones ejecutables. Un proceso incremental es aquel que implica la integración continua de la arquitectura del sistema para producir versiones, donde cada nuevo ejecutable incorpora mejoras incrementales sobre los otros.

Es un proceso iterativo e incremental ya que divide el proyecto en pequeños mini-proyectos, donde a cada uno de estos se le conoce como iteraciones, las cuales una vez terminada significa un incremento en el proyecto.



2.2.5.3. Vida del Proceso Unificado

El Proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema, los cuales constan de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a su vez en iteraciones que se repiten a lo largo de una serie de ciclos, donde el término de cada ciclo se obtiene una versión del software listo para ser entregado al cliente (**ver Figura 2.3**).

Una iteración puede pasar por los cinco flujos de trabajo fundamentales en el proceso unificado, los cuales son: requisitos, análisis, diseño, implementación y prueba.

Durante la fase de inicio, se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis de negocio para el producto. En esta fase se obtiene básicamente la descripción del problema a resolver mediante los casos de uso. Se describen las principales funciones que el sistema debe desempeñar, su arquitectura y un plan de tiempo y costo del proyecto.

Durante la fase de elaboración, se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. La arquitectura se expresa en forma de vista de todos los modelos del sistema, los cuales juntos representan el sistema entero. Durante esta fase del desarrollo, se realizan los casos de usos mas críticos que se identificaron en la fase de comienzo, dando como resultado una línea base de la arquitectura. Al término de esta fase se planifican las actividades y los recursos necesarios para la realización del proyecto.

Durante la fase de construcción se crea el producto, se añaden los músculos (software terminado) al esqueleto (la arquitectura). En esta fase, la línea base de la arquitectura crece hasta convertirse en el sistema completo. Al termino de esta fase, el sistema realizara las funciones especificadas por los casos de uso definidos en la fase

de elaboración, pero es probable que en esta fase el sistema tenga errores los cuales se irán detectando y resolviendo en la fase de transición.

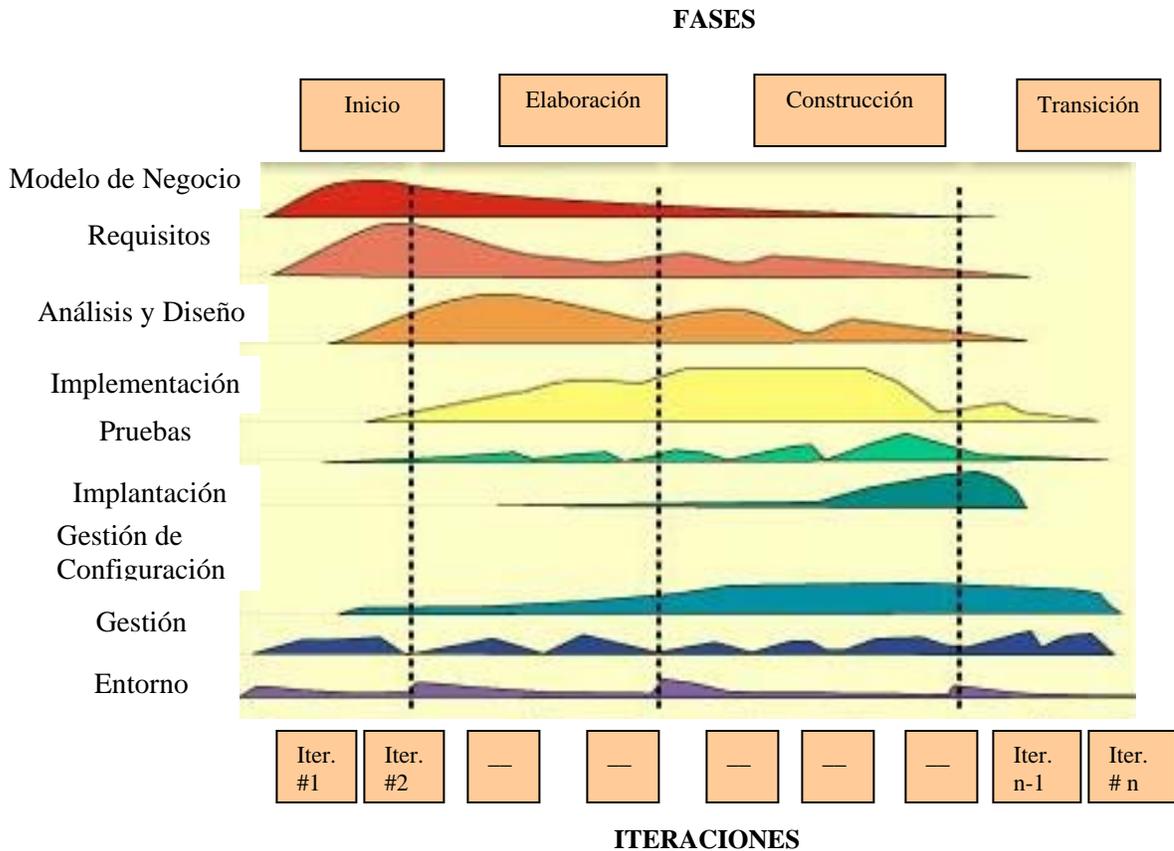


Figura 2.3. Fases y Flujos de Trabajos del Proceso Unificado de Desarrollo del Software.

Fuente: Jacobson, 2000.

La fase de transición cubre el periodo durante el cual el producto se convierte En la versión beta. En la versión beta un número reducido de usuarios con experiencia prueba el producto e informa de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas de las mejoras sugeridas en una versión general dirigida a la totalidad de la comunidad de usuarios. La fase de transición conlleva a



actividades como la fabricación, formación del cliente, el proporcionar una línea de ayuda y asistencia, y la corrección de los defectos que se encuentran tras la entrega.

2.2.6. Lenguaje Unificado de Modelado (UML) [12]

El proceso unificado utiliza el UML para expresar gráficamente todos los esquemas de un software.

El Lenguaje Unificado de Modelado emergió en los '90, luego de la búsqueda de un lenguaje de modelado que unificara, que siguió a la “guerra de métodos” de los '70 y '80. A pesar que UML evolucionó primeramente de varios métodos orientados al objeto de segunda generación UML no es simplemente un lenguaje para el modelado orientado a objetos de tercera generación. Su alcance extiende su uso más allá de sus predecesores. Y es la experiencia, experimentación y una gradual adopción del estándar lo que revelará su verdadero potencial y posibilitará a las organizaciones darse cuenta de sus beneficios.

2.2.6.1. Definición [13]

UML es un lenguaje de modelado unificado basado en una notación grafica que permite: especificar, construir, visualizar y documentar los objetos de un sistema.

Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.



UML es un lenguaje de propósito general para el modelado orientado a objetos; es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

Un modelo captura una vista de un sistema del mundo real. Es una abstracción de dicho sistema, considerando un cierto propósito. Así, el modelo describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle.

Un proceso de desarrollo de software debe ofrecer un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés. El código fuente del sistema es el modelo más detallado del sistema (y además es ejecutable). Sin embargo, se requieren otros modelos. Cada modelo es completo desde su punto de vista del sistema, sin embargo, existen relaciones de trazabilidad entre los diferentes modelos.

A través de un diagrama se puede realizar la representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo conexo de arcos (relaciones) y vértices (otros elementos del modelo). Un diagrama no es un elemento semántico, un diagrama muestra representaciones de elementos semánticos del modelo, pero su significado no se ve afectado por la forma en que son representados.

La mayoría de los diagramas de UML y algunos símbolos complejos son grafos que contienen formas conectadas por rutas. La información está sobre todo en la topología, no en el tamaño o la colocación de los símbolos. Hay tres clases importantes de relaciones visuales: conexión, contención y adhesión. Así como también existen cuatro clases de construcciones gráficas que se usan en la notación de UML: íconos, símbolos bidimensionales, rutas y cadenas.



2.2.6.2. Utilidad de UML

- ◆ UML es un lenguaje para modelamiento de propósito general evolutivo, ampliamente aplicable, debe de ser soportado por herramientas e industrialmente estandarizado. Se aplica a una multitud de diferentes tipos de sistemas, dominios, y métodos o procesos.
- ◆ Como lenguaje de propósito general, se enfoca en el corazón de un conjunto de conceptos para la adquisición, compartición, y utilización de conocimientos emparejados con mecanismos de extensión.
- ◆ Como un lenguaje de modelamiento ampliamente aplicable, puede ser aplicado a diferentes tipos de sistemas (software y no-software), dominios (negocios versus software) y métodos o procesos.
- ◆ Como un lenguaje para modelamiento soportable por herramientas, las herramientas ya están disponibles para soportar la aplicación del lenguaje para especificar, visualizar, construir y documentar sistemas.
- ◆ Como un lenguaje para modelamiento industrialmente estandarizado, no es un lenguaje cerrado, propiedad de alguien, sino más bien, un lenguaje abierto y totalmente extensible reconocido por la industria.

UML posibilita captura, comunicación y nivelación de conocimiento estratégico, táctico y operacional para facilitar el incremento de valor, aumentando la calidad, reduciendo costos y reduciendo el tiempo de presentación al mercado; manejando riesgos y siendo proactivo para el posible aumento de complejidad o cambio.

2.2.6.3. Bloques Básicos de UML

- ◆ Elementos: Estructurales, comportamiento, agrupación, anotación.
- ◆ Relaciones: Dependencia, asociación, generalización, realización.



- ◆ Diagramas: Clases, objetos, casos de uso, secuencia, colaboración, estados, actividades, componentes, despliegue, capas.

1. Elementos de Lenguaje

- ◆ **Elementos estructurales:** Son los nombres de los modelos de UML. En su mayoría son las partes estáticas de un modelo, y representan conceptos o cosas materiales. Colectivamente, los elementos estructurales se denominan *clasificadores*.

Entre los elementos estructurales tenemos:

- a) **Actores:** Un actor es "algo" o "alguien" que puede interaccionar con el sistema que se está desarrollando (ver **Figura 2.4**).



Figura 2.4. Representación de un Actor.

- b) **Casos de uso:** Un caso de uso es una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un actor particular (ver **Figura 2.5**).

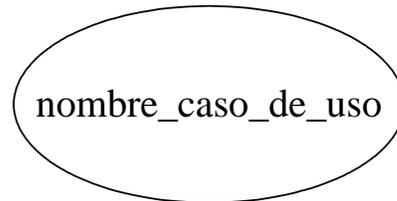


Figura 2.5. Representación de un Caso de Uso.

- c) **Clases:** Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica (ver **Figura 2.6**).

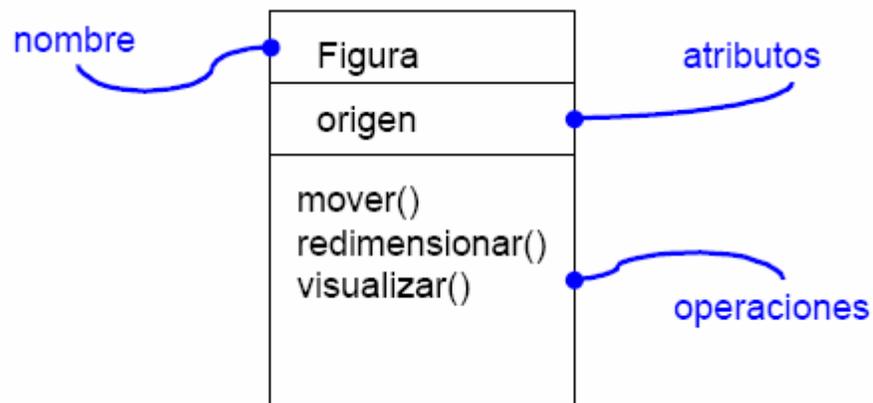


Figura 2.6. Representación de una Clase.

- d) **Objetos:** Un objeto es una instancia de alguna clase (ver Figura 2.7).

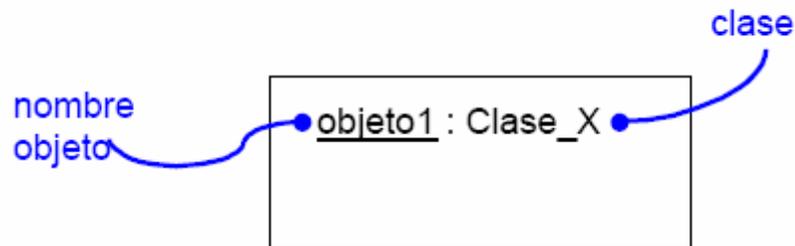


Figura 2.7. Representación de un Objeto.

- ◆ **Elementos de Comportamiento:** Son las partes dinámicas de los modelos UML. Éstos son los verbos de un modelo, y representan comportamiento en el tiempo y el espacio.
 - a) **Mensaje:** Los mensajes se usan para especificar una comunicación entre objetos (ver **Figura 2.8**). Se utilizan en los diagramas de secuencia.

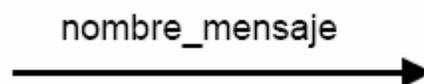


Figura 2.8. Representación de un Mensaje.

- ◆ **Elementos de Agrupación:** Son las partes organizativas de los modelos UML. Éstos son las cajas en las que puede descomponerse un modelo. Hay un tipo principal de elementos de agrupación: los paquetes.

- a) **Paquete:** Sirve para organizar elementos en grupos. Un paquete es puramente conceptual (sólo existe en tiempo de desarrollo), como se puede observar en la **Figura 2.9**.

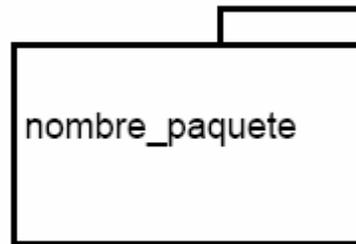


Figura 2.9. Representación de un Paquete.

2. Relaciones entre Elementos

- a) **Dependencia:**

Es una relación semántica entre dos elementos (o dos conjuntos de elementos), en la cual un cambio en un elemento puede afectar a la semántica de otro elemento (ver **Figura 2.10**).



Figura 2.10. Representación de la Relación de Dependencia entre dos Clases.

Existen varios tipos de dependencia predefinidas que se indican mediante estereotipos, por ejemplo: «extend», e «include» para casos de uso.

b) Asociación:

Es una relación estructural entre dos elementos, que describe las conexiones entre ellos (suele ser bidireccional). Es la única relación permitida entre los actores y los casos de uso (refleja la comunicación existente entre un actor y un caso de uso).



Figura 2.11. Representación de la Relación de Asociación entre Dos Clases.

c) Agregación:

Es una relación estructural entre un todo y sus partes. Se denota por una línea terminada en un "diamante" en el extremo de la clase que representa el todo.



Figura 2.12. Representación de una Relación de Agregación entre Dos Clases.

d) Generalización:

Es una relación taxonómica entre un elemento más general (el padre) y un elemento más específico (el hijo). Se usa tanto en diagramas de clases como en diagramas de casos de uso.



Figura 2.13. Representación de una Relación de Generalización entre Dos Clases.

3. Diagramas de UML^[16]

Los elementos de UML se muestran mediante diagramas que presentan múltiples vistas del sistema, ese conjunto de vistas son conocidos como modelos. UML presenta varios diagramas donde cada uno representa un aspecto del sistema, los cuales están conformados por Diagramas de estructura, Diagramas de comportamiento y Diagramas de interacción.

Diagramas de estructura, enfatizan en los elementos que deben existir en el sistema modelado:

- ◆ Diagrama de clases
- ◆ Diagrama de componentes
- ◆ Diagrama de objetos
- ◆ Diagrama de estructura compuesta
- ◆ Diagrama de despliegue
- ◆ Diagrama de paquetes

Diagramas de comportamiento, enfatizan en lo que debe suceder en el sistema modelado:



- ◆ Diagrama de actividades
- ◆ Diagrama de caso de usos
- ◆ Diagrama de estados

Diagramas de Interacción, un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- ◆ Diagrama de secuencia
- ◆ Diagrama de colaboración
- ◆ Diagrama de tiempos
- ◆ Diagrama de vista de interacción

1. **Diagrama de Casos de Uso:** El diagrama de casos de uso se emplea para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje (ver **Figura 2.14**). Los casos de uso no son artefactos orientados a objetos, es una técnica para captura de requisitos. Captura la funcionalidad del sistema vista por los usuarios.
2. **Diagrama de Secuencia:** El diagrama de secuencia muestra las interacciones entre los objetos organizadas en una secuencia temporal. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados (ver **Figura 2.15**). Representa una interacción, un conjunto de comunicaciones entre objetos organizadas visualmente por orden temporal. A diferencia de los diagramas de colaboración, los diagramas de secuencia incluyen secuencias temporales pero no incluyen las relaciones entre objetos. Pueden existir de forma de descriptor (describiendo todos los posibles escenarios) y en forma de instancia (describiendo un escenario real).

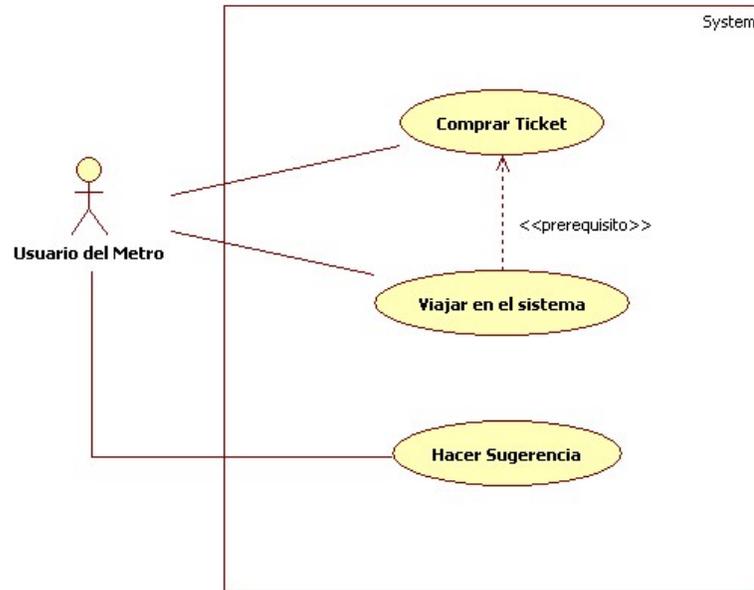


Figura 2.14. Diagrama de Casos de Uso.

Fuente: Martín y Kendall, 1999.

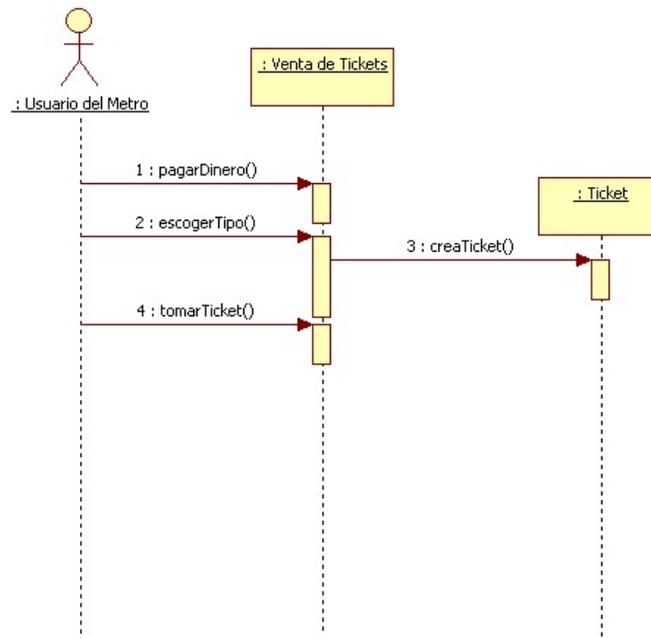


Figura 2.15. Diagrama de Secuencia.

Fuente: Martín y Kendall, 1999.

3. **Diagrama de Colaboración:** Un diagrama de colaboración muestra la implementación de la operación (ver **Figura 2.16**). La colaboración muestra los parámetros y las variables locales de la operación, así como operaciones más permanentes; modela los objetos y los enlaces significativos dentro de una interacción. Un rol describe un objeto, y un rol en la asociación describe un enlace dentro de una colaboración. Un diagrama de colaboración muestra los roles en la interacción en una disposición geométrica.

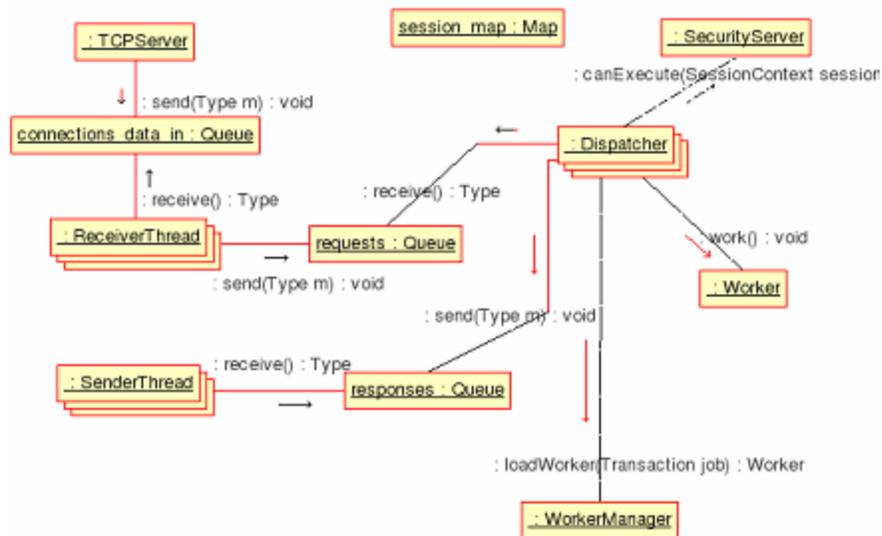


Figura 2.16. Diagrama de Colaboración.

Fuente: Martín y Kendall, 1999.

4. **Diagrama de Clases:** Es un diagrama que muestra un conjunto de clases y sus colaboraciones y relaciones. Estos diagramas sirven para visualizar las relaciones existentes entre las distintas clases y las formas en que colaboran unas con otras (ver **Figura 2.17**). Las relaciones entre las distintas clases son las relaciones comunes existentes en UML aunque con matices: Una asociación se traduce como que desde los objetos de una clase se puede acceder a los objetos de otra. Una dependencia se puede visualizar como que la clase utilizada es un parámetro de un método de la clase que la utiliza. Una generalización se traduce como una herencia entre clases. Además de las clases en el análisis, se encuentran tres estereotipos fundamentales a saber: de interfaz, de control y de entidad.

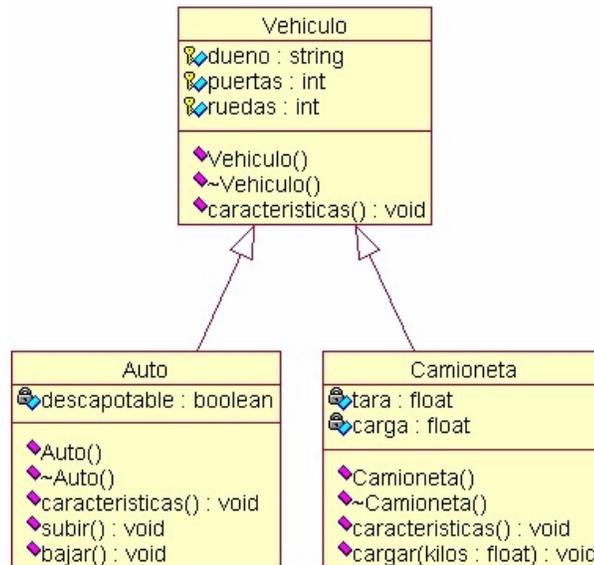


Figura 2.17. Diagrama de Clases.

Fuente: Martín y Kendall, 1999.

5. **Diagrama de Despliegue:** Los diagramas de despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos (ver **Figura 2.18**). La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

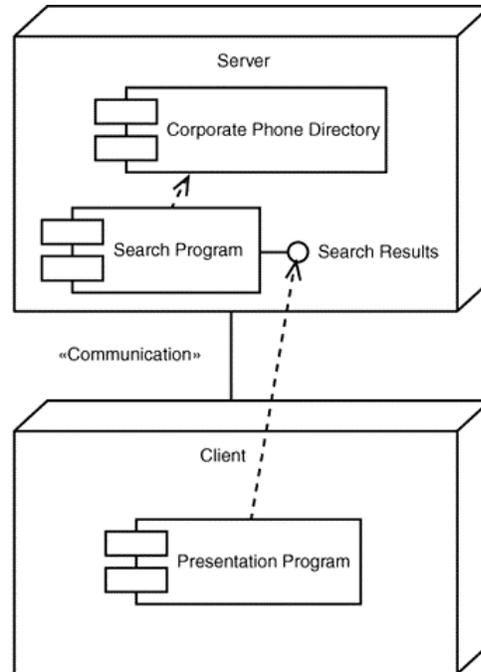


Figura 2.18. Diagrama de Despliegue.

Fuente: Martín y Kendall, 1999.

6. **Diagrama de Componentes:** Es un diagrama que muestra un conjunto de componentes y sus relaciones (ver **Figura 2.19**). Los diagramas de componentes muestran los componentes de un sistema desde un punto de vista estático.

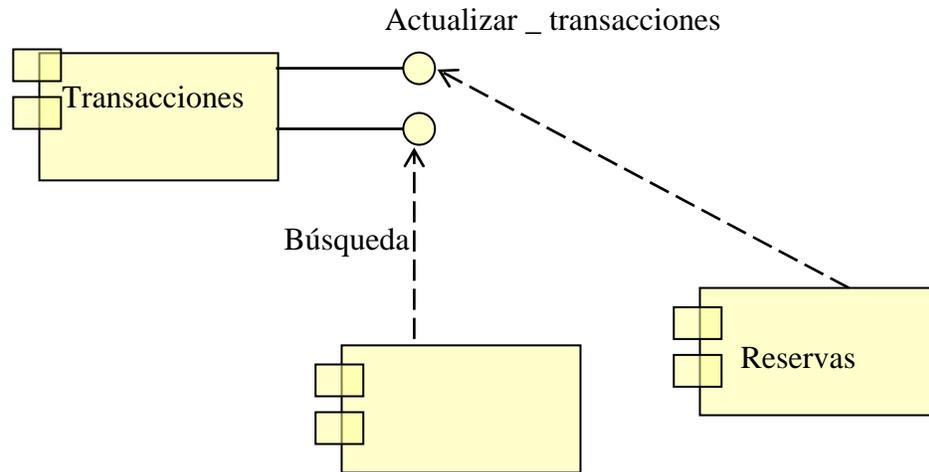


Figura 2.19. Diagrama de Componentes.

Fuente: Martín y Kendall, 1999.

7. **Diagrama de Estructura Compuesta:** Un diagrama de estructura compuesta muestra las partes internas, los conectores y los puertos que implementan un componente (ver **Figura 2.20**). Cuando se instancia el componente, también se instancias las copias de sus partes internas.

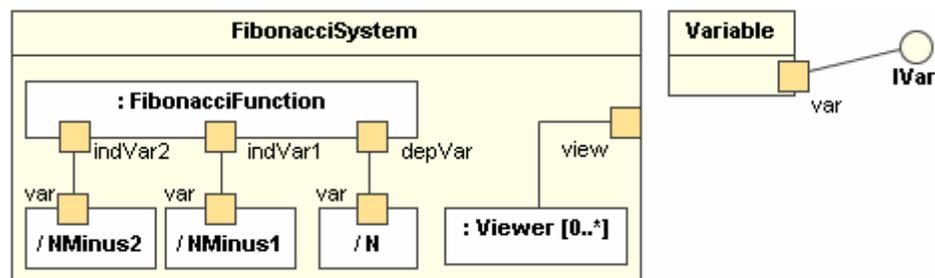


Figura 2.20. Diagrama de Estructura Compuesta.

Fuente: Martín y Kendall, 1999.



8. **Diagrama de Objetos:** un diagrama de objetos representa un conjunto de objetos y sus relaciones (ver **Figura 2.21**). Se utilizan para describir estructuras de datos, instantáneas estáticas de las instancias de los elementos existentes en los diagramas de clases. Los diagramas de objetos abarcan la vista de diseño estática o la vista de procesos estática de un sistema al igual que los diagramas de clases, pero desde la perspectiva de casos reales o prototípicos.
9. **Diagrama de Estados:** Un diagrama de estados representa una máquina de estados, constituida por estados, transiciones, eventos y actividades (ver **Figura 2.22**). Los diagramas de estados se utilizan para describir la vista dinámica de un sistema. Son especialmente importantes para modelar el comportamiento de una interfaz, una clase o una colaboración. Los diagramas de estados resaltan el comportamiento dirigidos por eventos de un objeto, lo que es especialmente útil al modelar sistemas reactivos.

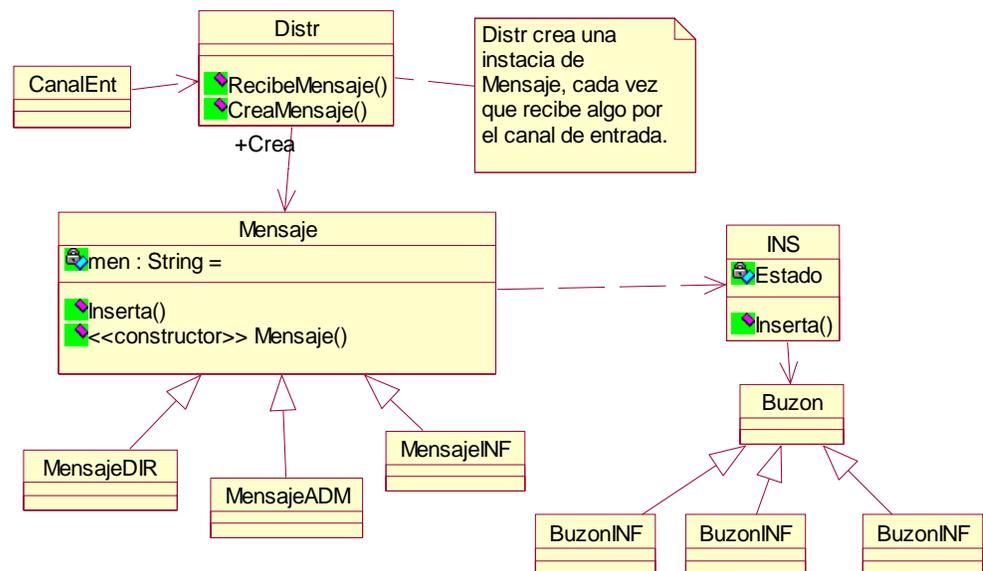


Figura 2.21. Diagrama de Objetos.



Fuente: Martín y Kendall, 1999.

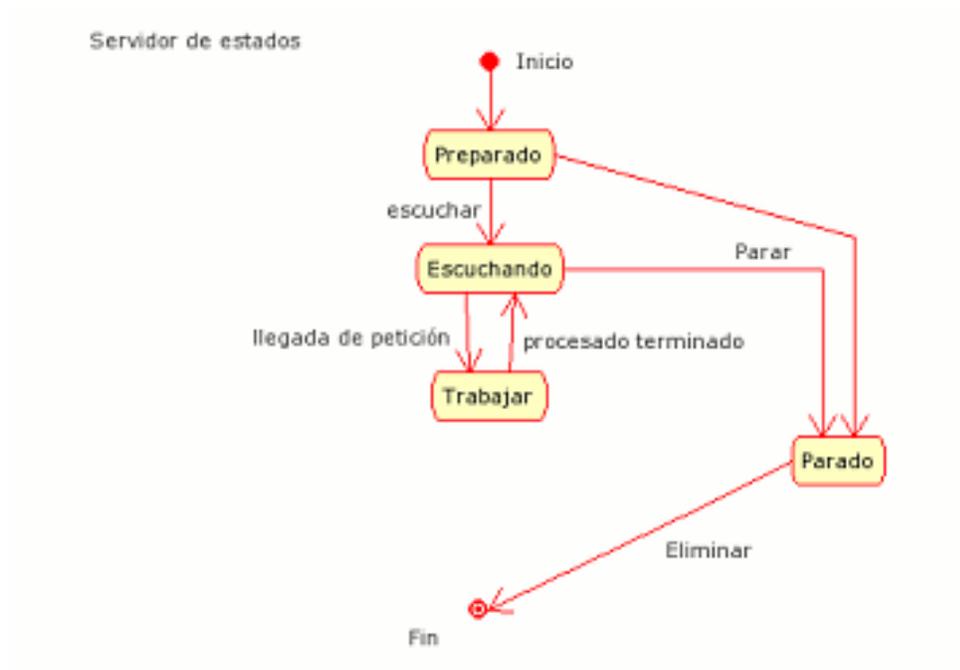


Figura 2.22. Diagrama de Estados.

Fuente: Martín y Kendall, 1999.

10. **Diagrama de Actividades:** un diagrama de actividades muestra el flujo paso a paso en una computación (ver **Figura 2.23**). Una actividad muestra un conjunto de acciones, el flujo secuencial o ramificado de acción en acción, y los valores que son producidos o consumidos por las acciones. Los diagramas de actividades se utilizan para ilustrar la vista dinámica de un sistema.

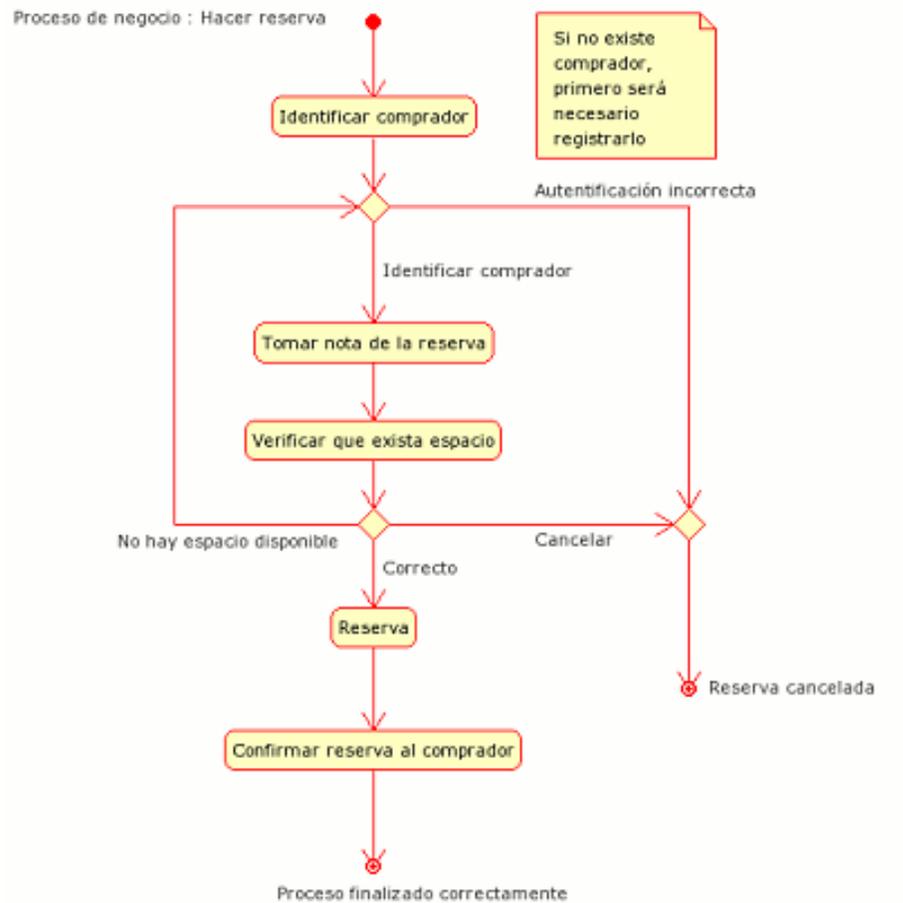


Figura 2.23. Diagrama de Actividades.

Fuente: Martín y Kendall, 1999.

2.2.7. Base de Datos

2.2.7.1. Definición

Una Base de datos es un conjunto de datos almacenados de forma integrada y compartida. Se entiende por integrada que la base de datos puede considerarse como un conjunto de varios archivos independiente, donde se elimina o se reduce al mínimo cualquier redundancia entre los mismos. Por compartida se entiende que



varios usuarios diferentes pueden acceder a la misma fracción de la base de datos, incluso al mismo tiempo, y utilizarla con fines diferentes. Por otro lado, un usuario determinado sólo tendrá acceso a algún sub-conjunto de la base completa. [18]

El análisis de requerimientos para una base de datos incorpora las mismas tareas que el análisis de requerimientos del software. Es necesario un contacto estrecho con el cliente; es esencial la identificación de las funciones e interfaces; se requiere la especificación del flujo, estructura y asociatividad de la información y debe desarrollarse un documento formal de los requerimientos. La base de datos es una disciplina organizacional, un método, más que una herramienta o una tecnología. Requiere de un cambio conceptual y organizacional.

Un sistema de manejo de datos (DBMS) es un conjunto de rutinas de software interrelacionadas, cada una de ellas es responsable de una tarea determinada. Ante cualquier pregunta de los usuarios, el DBMS responde utilizando los subprogramas apropiados para interpretar la consulta o localizar los datos deseados y presentarlos en el orden requerido. En definitiva, el DBMS permite el acceso a los datos en cualquier parte de la base de datos sin necesidad de conocer su organización dentro del dispositivo de almacenamiento.

2.2.7.2. Tipos de Base de Datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

- ❖ Según la variabilidad de los datos almacenados:

- ◆ **Bases de datos estáticas**



Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- ◆ **Bases de datos dinámicas**

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.

- ❖ Según el contenido:

- ◆ **Bases de datos bibliográficas**

Solo contienen un surrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias—ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

- ◆ **Bases de datos de texto completo**



Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

2.2.7.3. Modelo de Bases de Datos [5]

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

- ◆ **Bases de datos jerárquicas**

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.



Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

◆ **Base de datos de red**

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

◆ **Base de datos relacional**

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).



En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

Durante los años '80 (1980-1989) la aparición de dBASE produjo una revolución en los lenguajes de programación y sistemas de administración de datos. Aunque nunca debe olvidarse que dBase no utilizaba SQL como lenguaje base para su gestión.

◆ Bases de datos multidimensionales

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos multidimensional podría serlo también en una base de datos relacional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.



◆ Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.



Se está trabajando en SQL3, que es el estándar de SQL92 ampliado, que soportará los nuevos conceptos orientados a objetos y mantendría compatibilidad con SQL92.

◆ Bases de datos documentales

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesauro es un sistema de índices optimizado para este tipo de bases de datos.

◆ Base de datos deductivas

Un sistema de base de datos deductivas, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

◆ Gestión de bases de datos distribuida

La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etc.

2.2.8. Sistema de Gestión de Base de Datos (SGBD) [6]

2.2.8.1. Definición

Los sistemas de gestión de base de datos (SGBD), son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de base de



datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos.

2.2.8.2. Objetivos de SGBD

Existen distintos objetivos que deben cumplir los SGBD:

1. Independencia lógica y física de los datos: se refiere a la capacidad de modificar una definición de esquema en un nivel de la arquitectura sin que esta modificación afecte al nivel inmediatamente superior. Para ello un registro externo en un esquema externo no tiene por qué ser igual a su registro correspondiente en el esquema conceptual.
2. Redundancia mínima: se trata de usar la base de datos como repositorio común de datos para distintas aplicaciones.
3. Acceso concurrente por parte de múltiples usuarios: control de concurrencia mediante técnicas de bloqueo o cerrado de datos accedidos.
4. Distribución espacial de los datos: la independencia lógica y física facilita la posibilidad de sistemas de bases de datos distribuidas. Los datos pueden encontrarse en otra habitación, otro edificio e incluso otro país. El usuario no tiene por qué preocuparse de la localización espacial de los datos a los que accede.
5. Integridad de los datos: se refiere a las medidas de seguridad que impiden que se introduzcan datos erróneos. Esto puede suceder tanto por motivos



físicos (defectos de hardware, actualización incompleta debido a causas externas), como de operación (introducción de datos incoherentes).

6. Consultas complejas optimizadas: la optimización de consultas permite la rápida ejecución de las mismas.
7. Seguridad de acceso y auditoría: se refiere al derecho de acceso a los datos contenidos en la base de datos por parte de personas y organismos. El sistema de auditoría mantiene el control de acceso a la base de datos, con el objeto de saber qué o quién realizó una determinada modificación y en qué momento.
8. Respaldo y recuperación: se refiere a la capacidad de un sistema de base de datos de recuperar su estado en un momento previo a la pérdida de datos.
9. Acceso a través de lenguajes de programación estándar: se refiere a la posibilidad ya mencionada de acceder a los datos de una base de datos mediante lenguajes de programación ajenos al sistema de base de datos propiamente dicho.

2.2.9. Modelo Relacional de Base de Datos

2.2.9.1. Definición ^[19]

El modelo relacional representa la base de datos como una colección de relaciones. En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia; la información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Una relación representa una tabla o, hasta cierto punto, un archivo simple, en el que cada fila representa una colección de valores que describen una entidad del



mundo real. En la terminología del modelo relacional, una fila se denomina Tupla, una cabecera de columna es un Atributo y la tabla es una Relación. El tipo de datos que describe los tipos de valores que pueden aparecer en cada columna se llaman Dominio.

Una base de datos relacional es un conjunto de una o mas tablas estructuradas en registros y campos, que se vinculan entre si por un campo en común.

2.2.9.2. Descripción [1]

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene mayor relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de «consultas» que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional y el Cálculo relacional. El Álgebra relacional permite describir la forma de realizar una consulta, en cambio, el Cálculo relacional sólo indica lo que se desea devolver.



El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

◆ **Esquema**

Un esquema es la definición de una estructura (generalmente relaciones o tablas de una base de datos), es decir, determina la identidad de la relación y que tipo de información podrá ser almacenada dentro de ella; en otras palabras, el esquema son los metadatos de la relación. Todo esquema constará de:

- Nombre de la relación (su identificador).
- Nombre de los atributos (o campos) de la relación y sus dominios; el dominio de un atributo o campo define los valores permitidos para el mismo, es equivalente al tipo de dato por ejemplo character, integer, date, string, etc.

◆ **Instancias**

Una instancia de manera formal es la aplicación de un esquema a un conjunto finito de datos. En palabras no tan técnicas, se puede definir como el contenido de una tabla en un momento dado, pero también es válido referirnos a una instancia cuando trabajamos o mostramos únicamente un subconjunto de la información contenida en una relación o tabla, como por ejemplo:

- Ciertos caracteres y números (una sola columna de una sola fila).
- Algunas o todas las filas con todas o algunas columnas
- Cada fila es una tupla. El número de filas es llamado cardinalidad.
- El número de columnas es llamado aridad o grado.



2.2.9.3. Base de Datos Relacional

Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave. A esta manera de construir bases de datos se le denomina modelo relacional.

Estrictamente hablando el término se refiere a una colección específica de datos pero a menudo se le usa, en forma errónea como sinónimo del software usado para gestionar esa colección de datos. Ese software se conoce como SGBD (sistema gestor de base de datos) relacional o RDBMS (del inglés Relational Data Base Management System).

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización de una base de datos, el cual es entendido como el proceso necesario para que una base de datos sea utilizada de manera óptima.

Entre las ventajas de este modelo están:

1. Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves.
2. Garantiza la integridad referencial: Así al eliminar un registro elimina todos los registros relacionados dependientes.
3. Favorece la normalización por ser más comprensible y aplicable.

2.2.9.4. Formas Normales

Las relaciones que describe Codd al definir su modelo relacional, son transformadas en tablas al momento de trabajar con una base de datos. Las Formas Normales buscan



optimizar estas estructuras eliminando básicamente la redundancia utilizando como medio principal las dependencias funcionales. Las Formas Normales son un pequeño número de reglas que de cumplirse hacen que las estructuras posean la menor cantidad de redundancia posible.

- ◆ Primera Forma Normal (1NF): No hay campos múltiples (todo los campos son atómicos). Todas las filas deben tener el mismo número de columnas.
- ◆ Segunda Forma Normal (2NF): Todo campo que no sea clave debe depender por completo de toda la clave.
- ◆ Tercera Forma Normal (3NF): No hay dependencias transitivas. Un campo debe depender de la clave y no de otro campo.
- ◆ Forma Normal de Boyce-Codd (BCNF): Todos los determinantes de la tabla son clave candidata.
- ◆ Cuarta Forma Normal (4NF): Una fila no debe contener dos o más campos multi-valorados (aquellos que pueden contener más de un valor simultáneamente) sobre una entidad.
- ◆ Quinta Forma Normal (5NF): Una tabla puede almacenar atributos dependientes a la clave sólo por unión.
- ◆ Reglas de codd.

2.2.10 Interfaz de Usuario [7]

2.2.10.1. Definición

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.



2.2.10.2. Principales Funciones de las Interfaces de Usuario

Sus principales funciones son:

- ◆ Puesta en marcha y apagado
- ◆ Control de las funciones manipulables del equipo
- ◆ Manipulación de archivos y directorios
- ◆ Herramientas de desarrollo de aplicaciones
- ◆ Comunicación con otros sistemas
- ◆ Información de estado
- ◆ Configuración de la propia interfaz y entorno
- ◆ Intercambio de datos entre aplicaciones
- ◆ Control de acceso
- ◆ Sistema de ayuda interactivo.

2.2.10.3. Tipos de Interfaces de Usuario

1. Según la forma de interactuar del usuario

Atendiendo a como el usuario puede interactuar con una interfaz, nos encontramos con varios tipos de interfaces de Usuario:

- ◆ Interfaces alfanuméricas (intérpretes de mandatos) que solo presentan texto.
- ◆ Interfaces gráficas de usuario (GUI, Graphics User Interfaces), las que permiten comunicarse con el ordenador de una forma muy rápida e intuitiva representando gráficamente los elementos de control y medida.
- ◆ Interfaces táctiles, que representan gráficamente un "panel de control" en una pantalla sensible que permite interactuar con el dedo de forma similar a si se accionara un control físico.



2. Según su construcción

Pueden ser de hardware o de software:

- ◆ Interfaces hardware.- Se trata de un conjunto de controles o dispositivos que permiten la interacción hombre-máquina, de modo que permiten introducir o leer datos del equipo, mediante pulsadores, reguladores e instrumentos.
- ◆ Interfaces software.- Son programas o parte de ellos, que permiten expresar nuestros deseos al ordenador o visualizar su respuesta.

2.2.11. Reportes

El reporte se proyecta en el momento en que el autor del mismo define los datos y la manera de presentación de estos. En esta etapa, normalmente hay que definir conexiones a los distintos orígenes de datos para ver de dónde obtener los resultados que debe reflejar el reporte.

La administración del reporte está referida al hecho que en las organizaciones actuales tenemos distintas categorías de usuarios como por ejemplo los gerentes, los usuarios de servicio al cliente, entre otros. Por lo tanto, es importante definir quienes serán los usuarios del reporte.

CAPITULO III

FASE DE INICIO

En la fase de inicio se establecerá una visión del proyecto y su alcance, donde se obtendrá el concepto inicial del sistema, mediante la identificación de sus funcionalidades y requisitos; para conseguir ese esquema de funcionalidad se utilizarán los llamados casos de uso. En esta fase se identificarán todos los requisitos del proyecto y se elaborarán los modelos iniciales que capturen la trayectoria y desenvolvimiento del mismo, considerando los recursos disponibles y partiendo de diagramas que bosquejan los requisitos, los cuales son la base para la futura definición de la arquitectura del software.

Esta fase tendrá como propósito la determinación del ámbito del sistema y de sus principales funciones, identificar los riesgos asociados al proyecto, realizar el modelado del dominio del negocio, como abstracción o esquema del modelo de casos de uso, el cual captura los tipos mas importantes de objetos en el contexto del sistema, y proponer una visión general de la arquitectura candidata del sistema, asociado al sistema a desarrollar. En esta fase se detallarán todos los artefactos a generar por el proyecto. Se evaluará la viabilidad del proyecto.

Para llevar a cabo la fase de inicio se abarcarán los flujos de trabajo, requisitos, análisis y diseño, comenzando con la captura de requisitos mediante el conocimiento del contexto del sistema, dicho contexto se logra a través del establecimiento del diagrama de dominio del sistema, además se identificarán los riesgos que serán mitigados con la realización de las fases en el desarrollo del proyecto. A continuación, se realizará la captura de requisitos y se identificarán los actores que interactúan con el sistema por medio del modelamiento del diagrama de casos de uso. El siguiente flujo de trabajo es el de análisis, que permite modelar mediante los



diagramas de clase de análisis para la comprensión de la información obtenida del flujo anterior. Por último, se realizarán los diagramas de colaboración para representar las interacciones entre los objetos, y el diagrama de paquete de análisis, para encapsular los casos de usos que fueron definidos al realizar el análisis del sistema.

3.1. CONTEXTO DEL SISTEMA

Para determinar el contexto se debe realizar un acercamiento con el entorno del sistema que se desea desarrollar. Partiendo del conocimiento del Departamento de Mantenimiento, para luego analizar las actividades de control del proceso de mantenimiento realizado a equipos utilizados en líneas de producción en una planta ensambladora de vehículos, a través de la observación directa, con la finalidad de comprender los procesos de mayor relevancia dentro del área de interés, y ofrecer una visión representativa del contexto del sistema.

La Gerencia de Mantenimiento de MMC Automotriz, se encarga de llevar a cabo las labores de mantenimiento preventivo y correctivo de los equipos en sus instalaciones. Su labor principal es velar por el normal funcionamiento de los equipos productivos y mantener la infraestructura adecuadamente, para mantener la capacidad de producción de la planta, con bajos costos y alta seguridad.

El compromiso de ofrecer un servicio de mantenimiento a los equipos, sistemas e instalaciones utilizados por los diferentes departamentos de MMC Automotriz S.A., brindando una repuesta oportuna a sus requerimientos para conservar la disponibilidad y confiabilidad de los mismos al menor costo posible y lograr así la satisfacción de los clientes internos, utilizando eficientemente para ello los materiales y herramientas necesarias, cumpliendo con las normativas tanto de seguridad como



de protección ambiental con un personal calificado y comprometido con los valores de la organización.

La política del Departamento de Mantenimiento de MMC Automotriz S.A., esta enfocada en la calidad de sus servicios para obtener un rendimiento aceptable de las maquinarias durante más tiempo, así como de las satisfacción de sus cliente con la colaboración conjunta de gerentes, supervisores y obreros para el cumplimiento efectivo de sus funciones.

El proceso de análisis del mantenimiento a aplicarse en equipos utilizados en las líneas de producción comprendidas por: electropunto, latonería, pintura, vestidura, línea alta y línea final, para el ensamblaje de vehículos, se activa una vez que los programadores estudian los registros existentes de los equipos, actividad en la cual se realiza una selección por ubicación, para así distribuir los registros en grupos de trabajo, los cuales son gestionados para tener un control sobre todas las áreas de trabajo, presentando déficit en su aplicación, esto debido a que la información manejada es robusta, conllevando esto a que a pesar de los arduos trabajos, tiempos de trabajo destinados para ello, el mismo presenta un alto margen de error, ya que el dominio humano no logra tener un alcance completo para el satisfactorio desenvolvimiento del proceso bajo estudio.

Luego, se continua con la ejecución de la actividad, en la cual los programadores realizan el análisis de los registros de los equipos, los cuales contienen la bitácora de los equipos que intervienen en el proceso de mantenimiento aplicado a equipos utilizados en líneas de producción, considerando las especificaciones de los equipos, fecha y tipo de la última actividad realizada, insumos utilizados, así como personal disponible para la realización de las actividades, obteniendo los registros históricos de los mismos, y de esta manera tener un control del proceso de mantenimiento, con el objeto de generar las O.T. (Orden de Trabajo) para ejecutar la programación, y



obtener los listados de materiales a utilizar en las actividades a realizarse en dichos cronogramas. Los insumos requeridos para la ejecución de las actividades, deben ser solicitados con antelación al Almacén de Materiales no Productivos, por ello se debe manejar la información con eficiencia y eficacia, para lograr cumplir con los requerimientos de mantenimientos preventivos solicitados en las ordenes de trabajo, evitando de esta manera solventar dichas necesidades a través del mantenimiento correctivo, lo cual conlleva a paralizaciones de equipos, retrasando el proceso de ensamblaje en las líneas de producción.

El sistema **SYMP** (Sistema de Mantenimiento Preventivo) será el encargado de automatizar los procesos de recopilación y análisis de la información requerida para aplicar el mantenimiento a los equipos utilizados en el proceso de ensamblaje. Con lo cual se pretende reducir el tiempo de respuesta para obtener las O.T., las cuales contemplarán la programación de las actividades, de igual manera arrojará los insumos requeridos para efectuar las actividades, así como el personal calificado para ejecutarlas, realizando una distribución de acuerdo con el área en la cual este ubicado el equipo.

Una vez estudiado el área de interés, se procede a modelar todas sus actividades, para efectos del sistema **SYMP** se representará el contexto a través del Modelo de Dominio, el cual describe los conceptos importantes del contexto, como los objetos del dominio y enlaza unos con otros, con la finalidad de comprender los aspectos más importantes que se deben cubrir en este proyecto.

3.1.1. Modelo de Dominio

El Modelo de Dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan “las cosas” que existen o los eventos que suceden para así comprender y describir sus elementos relevantes.

Los objetos o clases de negocio se relacionan mediante asociaciones, agregaciones y composiciones, lo cual es la manera de modelar el desarrollo de las actividades. A continuación, se presenta el Modelo de Dominio (ver **Figura 3.1**) que representa las actividades relacionadas con el mantenimiento a equipos utilizados en líneas de producción, en una planta ensambladora de vehículos.

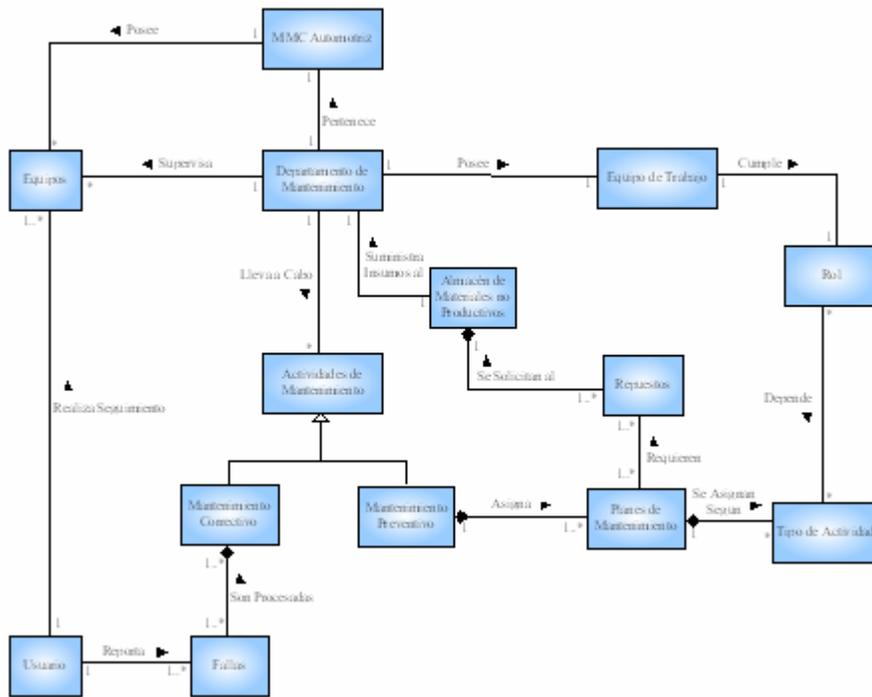


Figura 3.1. Modelo de Dominio.

En la **Figura 3.1**, se puede apreciar que la empresa MMC Automotriz posee Equipos los cuales son supervisados por el Departamento de Mantenimiento, que a su vez es el responsable de llevar a cabo las Actividades de Mantenimiento de los equipos e instalaciones, estas se encuentran divididas en dos tipos, el Mantenimiento Preventivo, por medio del cual son asignados Planes de Mantenimiento según el Tipo de Actividad, la cual va a depender del requerimiento que presente un determinado equipo en el transcurso de su ciclo de vida útil, y para efectuar estas actividades se



requieren Repuestos que son solicitados al Almacén de Materiales no Productivos, que se encuentra encargado de suministrar los insumos al Departamento de Mantenimiento que cuenta con un Equipo de Trabajo conformado por personal calificado, capaces de desempeñar los Roles necesarios para la ejecución de la actividad a efectuarse.

Durante el ciclo de vida de cada uno de los activos se van presentando fallas y para prevenirlas existen planes de mantenimiento, en caso de surgir fallas imprevistas que sean detectadas por el Usuario al realizar el seguimiento a los Equipos, este reporta las Fallas que son procesadas a través del Mantenimiento Correctivo para ser solventadas por medio del mismo, evitando la paralización o retardo de las actividades de producción.

3.1.2. Glosario de Términos

Es necesario contar con un lenguaje común para facilitar el intercambio de ideas. El glosario o diccionario modelo (ver **Tabla 3.1**) incluye y define todos los términos que son de vital importancia para la comprensión del análisis que se realizará en las fases de desarrollo del presente estudio, los cuales se mencionan a continuación:

Tabla 3.1. Glosario de Términos. (1/3)

TÉRMINO	DESCRIPCIÓN
Departamento de Mantenimiento	Unidad física en la que se realizan labores de mantenimiento a los equipos, sistemas e instalaciones, a la cual pertenece el trabajador.
Equipos	Colección de instrumentos y aparatos especiales para un trabajo.
Equipo de Trabajo	Se refiere a un grupo de personas, constituido por personal calificado para desempeñar una actividad.

**Tabla 3.1.** Glosario de Términos. (2/3)

TÉRMINO	DESCRIPCIÓN
Actividades de Mantenimiento	Se refiere al servicio que agrupa una serie de actividades cuya ejecución permite alcanzar un mayor grado de confiabilidad en los equipos.
Almacén de Materiales no Productivos	Unidad física en donde se almacenan materiales utilizados en actividades de soporte no relacionadas con el proceso productivo principal.
Rol	Representa los diferentes tipos de roles que cumplen los trabajadores, tales como, electricistas, mecánicos, entre otros.
Tipo de Actividad	Representa los diferentes tipos de labores de mantenimiento que se realizan a equipos.
Planes de Mantenimiento	Se refiere a las actividades de mantenimiento a aplicarse a los equipos que intervienen en el proceso productivo, las cuales consisten en la reparación de problemas que presenten éstos y en la adecuación de los mismos para trabajar sin ningún inconveniente durante todo el año.
Usuario	Representa a la persona que va a manejar el sistema.
Fallas	Es la terminación de la capacidad de un equipo para realizar una función requerida.
Mantenimiento Correctivo	Es el mantenimiento que se aplica cuando se ha producido el fallo y el paro súbito de la máquina o instalación, se ocupa de la reparación de los equipos, cuando se dispone del personal, repuesto, y documentos técnicos necesarios para efectuarlo.

**Tabla 3.1.** Glosario de Términos. (3/3)

TÉRMINO	DESCRIPCIÓN
Mantenimiento Preventivo	Es el mantenimiento que a través de rutinas de inspecciones periódicas y la renovación de los elementos dañados, disminuye el riesgo que el equipo presente fallas, detectándolas en su fase inicial.
Repuestos	Pieza o parte de un mecanismo que se tiene dispuesta para sustituir a otra, recambio.
Bitácora	Es un registro escrito de las acciones que se llevaron a cabo en cierto trabajo o tarea. Ésta incluye todos los sucesos que tuvieron lugar durante la realización de dicha tarea, las fallas que se produjeron, los cambios que se introdujeron y los costos que ocasionaron.
O.T.	Las Ordenes de Trabajo son una colección de actividades relacionadas entre si, a ser ejecutadas sobre un determinado equipo.

3.2. RIESGOS DEL SISTEMA

Los riesgos tienen como principal objetivo dirigir la viabilidad del proyecto a futuro, ya que estos pueden ocasionar pérdidas o daños, los cuales pongan en peligro el buen funcionamiento del sistema o en el peor de los casos que puedan afectar la culminación de dicho proyecto. En tal sentido, una vez identificados los posibles riesgos del sistema se priorizan y se define un plan de manejo para mitigarlos.



Una vez identificados pueden ser tratados de diversas maneras. Se cuenta fundamentalmente con cuatro elecciones: evitarlos, limitarlos, eliminarlos o controlarlos. Algunos riesgos pueden evitarse mediante una replanificación del proyecto o un cambio en los requisitos; otros deberían restringirse de modo que solo afecten una parte del proyecto; otros atenuarse ejercitándolos y observándolos; aunque existen riesgos que no pueden atenuarse, solamente pueden controlarse y observar si aparecen.

En el proyecto en desarrollo, los riesgos que fueron identificados para el Sistema de Mantenimiento Preventivo (**SYMP**) se presentan en la **Tabla 3.2**, que se muestra a continuación:

Tabla 3.2. Riesgos del Sistema.

RIESGO	DESCRIPCIÓN
Inadecuada interacción entre la empresa y los Analistas de Sistemas	Es de vital importancia tener una noción exacta del ámbito del sistema, a fin de manejar los aspectos fundamentales dentro del contexto, para el entendimiento de su funcionalidad.
Falla de operatividad	Fallas de energía eléctrica y problemas con el servidor, que no permitan la operatividad del sistema en la intranet.
Hardware y Software inadecuados	El hardware y software donde se ejecutará el sistema debe estar adaptado a los requerimientos de la aplicación. El software debe estar bien configurado para un buen desempeño.



3.3. REQUERIMIENTOS DEL SISTEMA

Los requerimientos establecen un contrato entre el sistema y su exterior, definen lo que se espera que realice el sistema, sin definir su funcionamiento interno. El objetivo final de cualquier diseño de la estructura de un sistema, es la de satisfacer los requerimientos de los usuarios para el sistema.

En la fase de inicio, la mayor parte del trabajo se lleva a cabo en el primer flujo de trabajo, el de requisitos, ya que en la captura de los mismos se confronta un proyecto desconocido y que no está claramente definido. Para obtener los requisitos, es necesario reunir los aspectos más interesantes que puedan aportar todos los participantes en el desarrollo, puesto que se trata de crear una lista que se tomará como punto de partida para la construcción del sistema; tomando en consideración que más adelante estos pueden cambiar o ser modificados de acuerdo a las necesidades y limitaciones que se presenten. En los requerimientos del sistema, se deben plasmar las funcionalidades que demandan los futuros usuarios de la aplicación.

Para la determinación de los requerimientos del Sistema de Mantenimiento Preventivo (**SYMP**), como se puede observar en la **Tabla 3.3** se consideraron los requerimientos funcionales y no funcionales del sistema que se desea. Los requerimientos funcionales describen la acción que debe realizar el usuario del sistema, especificando el comportamiento de entrada/salida, sin considerar restricciones físicas; mientras que los requerimientos no funcionales hacen referencia a aquellos aspectos no observables del sistema, pero que son de vital importancia para garantizar el mejor desempeño del sistema a diseñar, es decir, las propiedades que tienen que ver con rendimiento, velocidad, interfaz gráfica, uso de memoria, entre otros.

El sistema propuesto, que permitirá la automatización de las actividades relacionadas con el mantenimiento a equipos utilizados en líneas de producción en



una planta ensambladora de vehículos, debe cumplir con los siguientes requerimientos:

Tabla 3.3. Requerimientos del Sistema. (1/3)

REQUISITO	DESCRIPCIÓN	CATEGORÍA
Seguridad	El usuario debe iniciar una sesión haciendo uso de un identificador y una clave para permitir el acceso al sistema.	Funcional

Tabla 3.3. Requerimientos del Sistema. (2/3)

REQUISITO	DESCRIPCIÓN	CATEGORÍA
Configuraciones del Sistema	El sistema debe permitir definir las configuraciones sobre su funcionamiento e interacción con el usuario.	Funcional
Definición de las Características de la Actividad	Obtener la completa gestión de la actividad, esto incluye, características generales, específicas, data de operación en campo y manejo de la información asociada al ciclo de vida del equipo (fallas ocurridas, planes de mantenimiento y servicios realizados).	Funcional
Registro Histórico	El sistema debe interpretar la información extraída de las actividades programadas y las	Funcional



	ejecutadas para realizar el análisis, para así obtener el desempeño real.	
Arquitectura Robusta y Escalable	El sistema debe estar diseñado con una arquitectura que se adapte a los cambios y mejoras en su estructura.	No Funcional
Interfaz de Usuario	El sistema debe poseer una interfaz de usuario atractiva, amigable, de fácil manejo y acceso al contenido del mismo.	No Funcional

Tabla 3.3. Requerimientos del Sistema. (3/3)

REQUISITO	DESCRIPCIÓN	CATEGORÍA
Generación de Reportes	El sistema debe generar reportes impresos y digitales; y ordenes de trabajo con la programación de las actividades.	Funcional

3.3.1. Identificación de Actores

Un actor representa un conjunto coherente de roles que los usuarios de los casos de uso representan al interactuar con éstos. Normalmente, un actor representa un rol que es desempeñado por una persona, un dispositivo hardware o incluso otro sistema al interactuar con nuestro sistema. Por tanto, los actores simbolizan a terceros fuera del sistema que colaboran con el sistema. Una vez que se han identificado todos los actores del sistema, tenemos identificado el entorno externo al sistema.



Para el Sistema de Mantenimiento Preventivo (**SYMP**), se identificaron tres actores fundamentales como lo son: Administrador del Sistema, Supervisor de Planificación y Archivo de Ayuda. A continuación, se presenta la descripción de cada uno de los actores (ver **Tabla 3.4**).

Tabla 3.4. Descripción de Actores. (1/2)

ACTOR	DESCRIPCIÓN
Administrador del Sistema	Es el encargado de registrar y manipular la información contenida referente a los equipos, actividades, personal y fallas relacionadas con el mantenimiento.

Tabla 3.4. Descripción de Actores. (2/2)

ACTOR	DESCRIPCIÓN
Supervisor de Planificación	Es el actor principal del sistema, ya que representa a la persona encargada de llevar a cabo la aplicación así como del control de las actividades relacionadas al mantenimiento de equipos.
Archivo de Ayuda	Es el archivo en donde se permite manejar y codificar la ayuda del sistema.

3.3.2. Casos de Uso

Los casos de uso denotan un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso; el conjunto de casos de uso representa la totalidad



de operaciones desarrolladas por el sistema, y el cual va acompañado de un nombre significativo.

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto.

Los casos de uso identificados para el sistema **SYMP** se presentan en la **Tabla 3.5**, la cual se muestra a continuación:

Tabla 3.5. Casos de Uso del Sistema SYMP. (1/2)

CASO DE USO	DESCRIPCIÓN
Configurar sistema	Se refiere a las operaciones relacionadas con la configuración del sistema, relacionado con el mantenimiento periódico que se le debe efectuar al sistema.

Tabla 3.5. Casos de Uso del Sistema SYMP. (2/2)

CASO DE USO	DESCRIPCIÓN
Gestionar registros de falla	Permite procesar la información de la ocurrencia de la falla, haciendo seguimiento hasta que esta sea solventada.
Consultar registros	Representa las distintas consultas que el sistema puede realizar, en donde se muestra la información de las bitácoras de los equipos, la ubicación de los mismos, y el



	personal disponible, así como la información de las actividades que pueden ser programadas.
Registrar falla nueva	Se refiere al registro que realiza el usuario para hacer la notificación de las fallas que presentan los equipos en un determinado momento, permitiendo cargar al sistema las actividades no programadas que se realizan a los equipos.
Gestionar programación de mantenimiento preventivo	Permite generar la programación de acuerdo al período de tiempo establecido, hacer los registros de las actividades que se realizan a los equipos, así como originar el listado de materiales necesarios para la ejecución de la programación.
Procesar reportes	Permite visualizar los mantenimientos procesados, no procesados, los que están pendientes, así como las fallas que presentan los equipos en un período de tiempo determinado.
Procesar gráficas	Representa la visualización de gráficas que generan las estadísticas de los mantenimientos programados, ejecutados y lo programado vs. Ejecutado en un intervalo de tiempo.
Mostrar ayuda	Permite mostrar una ayuda para la utilización del sistema.

3.3.3. Modelo de Casos de Uso

El modelo de casos de uso tiene como finalidad la captura de todos o parte de los requisitos funcionales del sistema, ya que describe la funcionalidad propuesta del nuevo sistema. El recurso principal del cual se vale este modelo es el diagrama de casos de uso, que muestra las distintas operaciones que se esperan de una aplicación o sistema, y cómo se relaciona con su entorno (usuarios u otras aplicaciones).



El modelado de casos de uso es una técnica efectiva y a la vez simple para modelar los requerimientos del sistema desde la perspectiva del usuario. Presenta el sistema desde la perspectiva de su uso y esquematiza como proporcionará valor a sus usuarios. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores para limitar las funciones con que dispondrá el sistema luego de ser implementado, además proporciona la entrada fundamental para el análisis, el diseño, la implementación y las pruebas.

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Mediante la **Figura 3.2** se logra representar los requisitos del sistema **SYMP**, esto a través del modelo de casos de uso del mismo, en el cual se puede observar que el administrador del sistema es el encargado de ejecutar el proceso que hace posible la recolección de la información tanto de los equipos que intervienen en el proceso de ensamblaje, como de las actividades de mantenimiento que se deben aplicar a dichos equipos, y del personal necesario para la ejecución de las mismas, esto con el fin de procesar los datos, y luego proceder con su almacenamiento para su futura utilización.

La actividad principal del sistema es gestionar la programación de mantenimiento preventivo, donde el supervisor de planificación genera la programación de actividades a realizarse de acuerdo al período de tiempo establecido, asignando el mantenimiento a los equipos de acuerdo a los registros de las actividades que han sido efectuadas a los mismos (bitácoras), y en el caso de no llevarse a cabo la programación generada a determinados equipos, el supervisor puede generar un listado de mantenimiento pendiente por efectuarse a dichos equipos. También, origina los listados de insumos a utilizarse en cada una de las actividades, en el período de



tiempo establecido por el usuario, adecuando la programación a la jornada laboral, registrando los días no laborables.

El supervisor de planificación también puede hacer el registro de las fallas que presentan los equipos en un determinado momento, permitiendo cargar al sistema las actividades no programadas que se realizan a los equipos. El sistema **SYMP** a su vez permite procesar la información de la ocurrencia de la falla, haciendo seguimiento hasta que esta sea solventada.

El caso de uso consultar registros permite al supervisor de planificación realizar consultas sobre la información de las bitácoras de los equipos, los registros de su ubicación, ya sea por galpón, área o zona, así como del personal disponible, tanto para las O.T. generadas como para cada actividad, y las actividades de mantenimiento, incluyendo la descripción la cual contiene los insumos y personal necesario para ejecutarlas.

El supervisor de planificación a través del caso de uso generar reportes puede visualizar los mantenimientos procesados, no procesados, los que están pendientes, así como las fallas que presentan los equipos en un periodo de tiempo determinado, y obtener reportes impresos y electrónicos, en donde sea posible representar los datos obtenidos mediante gráficas del rendimiento de los mantenimientos realizados a los equipos a través del tiempo.



3.4. ANÁLISIS

Durante esta fase se analizan los requisitos que se describen en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión mas precisa de los requisitos y una descripción de los mismos que sea fácil de mantener, y que ayude a constituir el sistema, incluyendo su arquitectura. Además, en el análisis se pueden organizar los requisitos de manera que facilite su comprensión, su preparación, su modificación, y, en general, su mantenimiento. Esta estructura (basada en clases de análisis y paquetes) es independiente de la que se le da a los requisitos (basada en casos de uso). Sin embargo, existe una trazabilidad directa entre las mismas, de forma que se puede hacer la traza de diferentes descripciones, en diferentes niveles de detalle, del mismo requisito y mantener su consistencia mutua con facilidad.

El lenguaje que se utiliza en el análisis se basa en un modelo de objetos conceptual, que llamamos modelos de análisis. El modelo de análisis ayuda a refinar los requisitos y permite razonar sobre los aspectos internos del sistema, incluidos sus recursos compartidos internos, también ayuda a estructurar los requisitos.

Para el flujo de trabajo análisis se analizarán mas a fondo los casos de uso identificados en los requisitos de la fase de inicio, planteando las clases de análisis de ellos, y estableciendo sus diagramas de clases de análisis y los paquetes de análisis, lo cual permitirá precisar y fundamentar la línea base de la arquitectura del sistema.

A diferencia del lenguaje intuitivo utilizado en la captura de requisitos, en el flujo de trabajo análisis se utiliza un lenguaje basado en un modelo de objetos conceptual o modelo de análisis que permite refinar los casos de uso en el flujo anterior y comenzar a indagar en aspectos internos del sistema.



3.4.1. Análisis de los Casos de Uso

El análisis de los requisitos capturados en forma de casos de uso es un flujo de trabajo que permite de manera iterativa e incremental la construcción del modelo de análisis del sistema. Este modelo de análisis crece a medida que se analizan más y mas casos de uso, ofreciendo una comprensión mas precisa de los requisitos y una descripción de los mismos que sea fácil de mantener, y que ayude a estructurar el sistema entero.

3.4.1.1. Identificación de las Clases de Análisis

Las clases de análisis, también llamados objetos de análisis, son clases estereotipadas que representan un modelo conceptual para los elementos del sistema que tienen responsabilidad y comportamiento. Hay tres tipos de clases de análisis, las cuales son usadas en todo el modelo de análisis:

- ◆ Clases de interfaz
- ◆ Clases de entidad
- ◆ Clases de control

Una clase de interfaz es una clase estereotipada que modela la interacción entre uno o más actores (usuarios y sistemas externos) y el sistema. Esta interacción a menudo implica recibir (y presentar) información y peticiones de (y hacia) los usuarios y los sistemas externos. Las clases de interfaz modelan las partes del sistema que dependen de sus actores, lo cual implica que clarifican y reúnen los requisitos en los límites del sistema.

Las clases de entidad, por su parte, se utilizan para modelar información que posee una vida larga y que es a menudo persistente. Modela la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto del mundo real o un suceso del mundo real.



Las clases de control representan coordinación, secuencia, transacciones y control de otros objetos, y se usan con frecuencia para encapsular el control de un caso de uso en concreto.

Una clase de análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema, y constituye una colección de clases que describe los requisitos funcionales del sistema.

- **Clases de interfaz**

Una interfaz es una colección de operaciones que representan servicios ofrecidos por una clase o componente. La clase realiza (o suministra una realización de) varias interfaces. El prototipo de clase de interfaz se muestra en la **Figura 3.3**, a continuación se detallan las clases de interfaz que se identificaron en el modelo de análisis (ver **Tabla 3.6**):



Figura 3.3. Prototipo de Clase de Interfaz

Tabla 3.6. Clases de Interfaz. (1/3)

CLASE DE INTERFAZ	DESCRIPCIÓN
IU Gestionar Data Equipo	Permite al administrador del sistema crear, validar y eliminar los datos de los equipos.
IU Registrar Equipo	Permite al administrador del sistema registrar la data correspondiente al equipo.

**Tabla 3.6.** Clases de Interfaz. (2/3)

CLASE DE INTERFAZ	DESCRIPCIÓN
IU Actualizar Equipo	Permite al administrador del sistema realizar modificaciones con respecto a la data de los equipos.
IU Eliminar Equipo	Permite al administrador del sistema eliminar datos de los equipos.
IU Gestionar Registros de Falla	Permite al supervisor de planificación realizar el seguimiento a las fallas registradas.
IU Consultar Registros	Permite al supervisor de planificación efectuar las consultas de la información almacenada de las actividades, personal y equipos.
IU Consultar Actividad	Permite al supervisor de planificación efectuar consultas sobre la data almacenada de las actividades.
IU Consultar Personal	Permite al supervisor de planificación efectuar consultas sobre la data almacenada del personal que labora en el Dpto. de Mantenimiento.
IU Consultar Equipos	Permite al supervisor de planificación efectuar consultas sobre la data almacenada de los equipos.
IU Registrar Falla Nueva	Permite al supervisor de planificación cargar los datos de las fallas imprevistas.
IU Registrar Días no Laborables	Permite al supervisor de planificación cargar los días en los que no se va a laborar en el Dpto. de Mantenimiento.
IU Asignar Mantenimiento	Permite al supervisor de planificación seleccionar el período de tiempo en el que se va a efectuar la programación.

**Tabla 3.6.** Clases de Interfaz. (3/3)

CLASE DE INTERFAZ	DESCRIPCIÓN
IU Gestionar Programación de Mantenimiento Preventivo	Permite al supervisor de planificación emitir la programación de mantenimiento preventivo.
IU Ver Cronograma de Mantenimiento por Equipo	Permite al supervisor de planificación generar en pantalla el cronograma de las actividades programadas por cada equipo.
IU Generar Listado de Materiales	Permite al supervisor de planificación emitir el listado de los materiales requeridos para llevar a cabo las órdenes de trabajo.
IU Listar Mantenimiento Pendiente de Equipos	Permite al supervisor de planificación generar el listado de las actividades que están pendientes en sistema por efectuarse.
IU Generar Reporte de Mtto. Pendiente	Permite al supervisor de planificación emitir el reporte con el estatus de los mantenimientos que aún no han sido efectuados.
IU Procesar Reportes	Permite al supervisor de planificación generar reportes sobre la información almacenada relacionada con los mantenimientos.
IU Generar Reporte de Mtto. Efectuado	Permite al supervisor de planificación emitir el reporte con el estatus de los mantenimientos efectuados.
IU Generar Reporte de Fallas Registradas	Permite al supervisor de planificación emitir el reporte de las fallas ocurridas en un período de tiempo determinado.
IU Mostrar Ayuda	Permite al supervisor de planificación consultar la información general sobre el manejo del sistema SYMP.



- **Clase de Control**

Una operación es un servicio que una instancia de la clase puede realizar. En la **Figura 3.4** se puede observar el estereotipo de la clase de control, a continuación se especificarán las clases que se encontraron en el diseño de análisis (ver **Tabla 3.7**):



Figura 3.4. Prototipo de Clase de Control

Tabla 3.7. Clases de Control. (1/3)

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Gestionar Data Equipo	Actualiza los registros de los equipos en el sistema.
Gestor Registrar Data Equipo	Registra datos de los equipos en el sistema.
Gestor Actualizar Data Equipo	Modifica datos de los equipos en el sistema.
Gestor Eliminar Data Equipo	Elimina datos de los equipos en el sistema.
Gestor Generar Reporte	Emite los reportes de las consultas hechas al sistema de las fallas detectadas.
Gestor Consultar Actividad	Procesa las interacciones del supervisor de planificación con la información de las actividades de mantenimiento.
Gestor Consultar Personal	Procesa las interacciones del supervisor de planificación con la información del personal que intervienen en el mantenimiento efectuado a los equipos.



Gestor Consultar Equipos	Procesa las interacciones del supervisor de planificación con la información de los equipos que intervienen en el proceso productivo de la empresa.
--------------------------	---

Tabla 3.7. Clases de Control. (2/3)

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Consultar Equipos por Área	Procesa las interacciones del supervisor de planificación con la información de los equipos existentes en un área determinada, que intervienen en el proceso productivo de la empresa.
Gestor Consultar Equipos por Galpón	Procesa las interacciones del supervisor de planificación con la información de los equipos según su ubicación.
Gestor Consultar Equipos por Zona	Procesa las interacciones del supervisor de planificación con la información de los equipos mediante su ubicación por una determinada zona dentro de la empresa.
Gestor Registrar Falla Nueva	Gestiona el proceso de registro de fallas nuevas.
Gestor Programar Mantenimiento	Procesa las actividades de asignación y visualización de los mantenimientos.
Gestor Registrar Días no Laborables	Procesa la operación de registro de días no laborables establecidos en el calendario laboral.
Gestor Listar Mantenimiento Pendiente de Equipos	Gestiona el proceso de salida de listados de mantenimientos que no hayan sido efectuados en determinados equipos.
Gestor Generar Listado de Materiales	Gestiona el proceso de verificación de los insumos necesarios para la ejecución de una determinada



	programación de mantenimiento.
Gestor Ver Cronograma de Mantenimiento por Equipo	Procesa el cronograma de mantenimiento para visualizar la programación de mantenimiento de un equipo específico.

Tabla 3.7. Clases de Control. (3/3)

CLASE DE CONTROL	DESCRIPCIÓN
Gestor Asignar Mantenimiento	Procesa la asignación de los mantenimientos que se deben realizar a los equipos en un determinado período de tiempo.
Gestor Generar Reporte de Mtto. Pendiente	Procesa la salida de la información relacionada con el mantenimiento pendiente de los equipos.
Gestor Generar Reporte de Mtto. Efectuado	Procesa la salida de la información relacionada con el mantenimiento efectuado de los equipos.
Gestor Generar Reporte Fallas Registradas	Procesa la salida de la información relacionada con la incidencia de las fallas que presentan los equipos en período de tiempo determinado.
Gestor Mostrar Ayuda	Gestiona los procesos relacionados con la consulta de información sobre el manejo del sistema SYMP.

- **Clases de Entidad**

La clase de entidad permite modelar la información. La **Figura 3.5** muestra el prototipo de la clase entidad, a continuación se especifican las clases encontradas en el modelo de análisis (ver **Tabla 3.8**).

**Figura 3.5.** Prototipo de Clase de Entidad**Tabla 3.8.** Clases de Entidad. (1/2)

CLASE DE ENTIDAD	DESCRIPCIÓN
Componente	Representa las partes en las cuales está dividida un equipo, para luego asignar las actividades de mantenimiento que deben ser ejecutadas sobre ese componente.
Equipo	Representa un determinado equipo ubicado por galpón, área y zona dentro de la empresa.
Datos_Equipo	Representa todas las especificaciones de un equipo y descripción técnica del mismo, así como también su año de instalación y costo histórico.
Correctivo	Representa la información de las actividades que se realizan una vez ocurrida una falla; indica el tiempo estimado, la duración y la fecha de ejecución de la falla.
Correctivo _ interrupción	Representa la duración en la cual estuvo detenido un equipo.
Personal	Representa a cada uno de los empleados que intervienen en las actividades de mantenimiento dentro de la empresa.
Falla	Representa el registro que se genera cuando ocurre una falla en un determinado período de tiempo.
Tarea	Representa toda la información de las labores de mantenimiento que deben ser aplicadas cuando ocurre una falla.



Actividad	Representa toda la información de las actividades que se realizan a los equipos y el tiempo de ejecución en la cual se realiza la misma.
-----------	--

Tabla 3.8. Clases de Entidad. (2/2)

CLASE DE ENTIDAD	DESCRIPCIÓN
Calendario	Representa los días laborables y no laborables dentro de la empresa.
Orden de trabajo	Representa las órdenes de trabajo emitidas para un determinado equipo.
Materiales	Representa el listado de los materiales que son requeridos para llevar a cabo alguna labor de mantenimiento.
Ayuda	Representa la información a la cual puede acceder el usuario en caso de necesitar conocer en detalle el funcionamiento del sistema.

- **Diagrama de Clases de Análisis para el Caso de Uso Gestionar Programación de Mantenimiento Preventivo**

En el diagrama de clases de análisis para el caso de uso gestionar programación de mantenimiento preventivo, como se puede observar en la **Figura 3.6**, agrupa las principales actividades relacionadas con la programación de mantenimiento a los equipos que intervienen en el proceso productivo, es iniciado por la instancia del actor Supervisor de Planificación quien selecciona la clase de interfaz denominada Interfaz Gestionar Programación de Mantenimiento Preventivo, que se encarga de la interacción con el usuario, y comunicar las opciones seleccionadas al sistema, además



se tienen las clases de control Gestor Registrar Días no Laborables, que se encarga del control del calendario laboral de la empresa, otra clase de control es definida como Gestor Asignar Mantenimiento, a través de la cual el Supervisor de Planificación solicita la asignación del mantenimiento a los equipos, seleccionando un período de tiempo determinado. La clase de control Ver Cronograma de Mantenimiento por Equipo, permite obtener el cronograma de mantenimiento a efectuarse por cada equipo, de igual manera al seleccionar la clase de control Generar Listado de Materiales, se obtiene el listado de los materiales y herramientas solicitados para llevar a cabo cada una de las actividades que se encuentren dentro de una programación específica. La clase de control Gestor Listar Mantenimiento Pendiente de Equipos, maneja la información referente a los mantenimientos que no fueron efectuados en una determinada programación, para llevar un control de las incidencias en un período de tiempo finito.

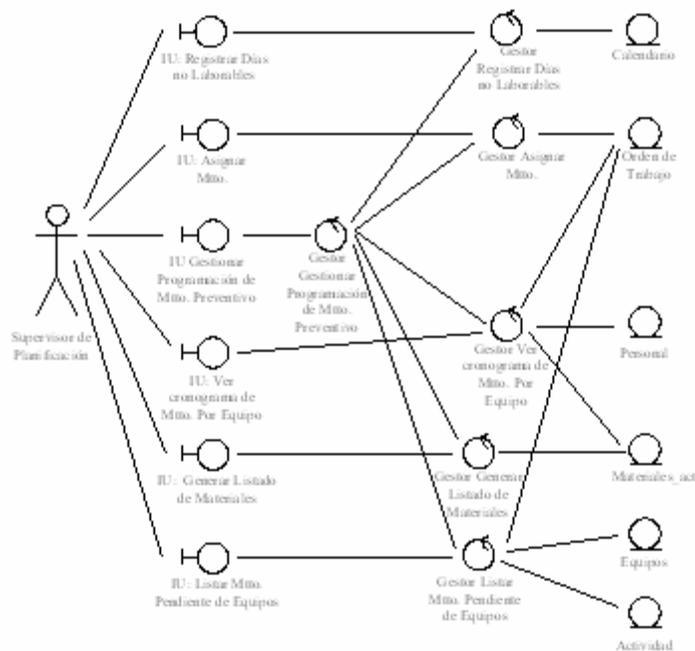


Figura 3.6. Diagrama de Clases de Análisis para el Caso de Uso Gestionar Programación de Mantenimiento Preventivo.

- **Diagrama de Clases de Análisis para el Caso de Uso Consultar Registros**

La **Figura 3.7** representa el diagrama de clases de análisis del caso de uso Consultar Registros, donde se especificó una clase de interfaz denominada Interfaz Consultar Registros, encargada de la interacción con el usuario, que le permite a Supervisor de Planificación realizar las diferentes consultas que el sistema permite.

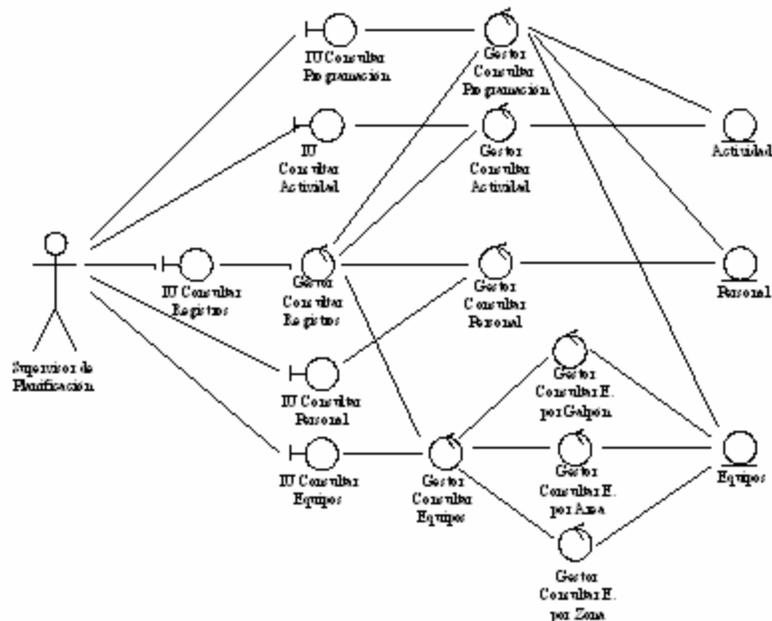


Figura 3.7. Diagrama de Clases de Análisis para el Caso de Uso Consultar Registros.

Para efectuar las consultas del sistema es necesario que el Supervisor de Planificación haga uso de la Interfaz Consultar Registros, ésta se encuentra conformada por cuatro procesos, el Gestor Consultar Programación que permite obtener la información relacionada a una determinada programación, la clase de control Gestor Consultar Actividad permite realizar la consulta de la información que



contiene la actividad a seleccionar, la clase de control Gestor Consultar Personal permite adquirir la información almacenada del personal que labora en el departamento, así como la disponibilidad del mismo, y la clase de control Gestor Consultar Equipos, que a su vez permite realizar la consulta de un equipo según su ubicación, ya sea por galpón, por área ó por zona, respectivamente, para obtener el detalle de información correspondiente al mismo.

3.4.1.2. Diagramas de Colaboración

En los diagramas de colaboración, se muestran las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a esos enlaces. El nombre de un mensaje debe denotar el propósito del objeto invocante en la interacción con el objeto invocado.

- **Diagrama de Colaboración para el Caso de Uso Gestionar Data Equipo**

En el diagrama de colaboración del caso de uso Gestionar Data Equipo, como se muestra en la **Figura 3.8**, el Supervisor de Planificación solicita la Interfaz Gestionar Data Equipo, en la cual puede acceder a la información de los equipos, y solicitar a través del Gestor Gestionar Data Equipo el ingreso, modificación o eliminación del registro de los equipos; así como ingresar a la información de las actividades de mantenimiento que se programan para estos, de igual forma para el caso de uso Gestionar Data Actividad, el cual solicita a través del Gestor Gestionar Data Actividad el ingreso, modificación o eliminación del registro de las actividades, y por último, para acceder a la información del personal por medio del Gestor Gestionar Data de Personal, almacenar las modificaciones hechas a los registros del personal que intervienen en las actividades de mantenimiento, y la data registrada es almacenada en las entidades componente, equipo y datos_equipo; aclarando la funcionalidad de cada proceso mediante el archivo de ayuda. La **Tabla 3.9** contiene



los mensajes del Diagrama de Colaboración para el Caso de Uso Gestionar Data Equipo, que describirán la secuencia de acciones.

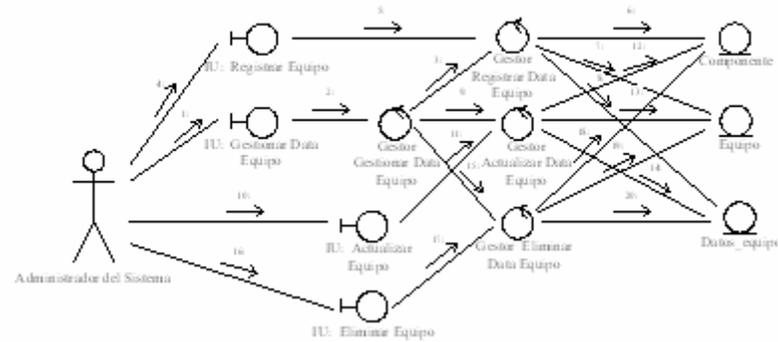


Figura 3.8. Diagrama de Colaboración para el Caso de Uso Gestionar Data Equipo.

- **Diagrama de Colaboración para el Caso de Uso Gestionar Registros de Falla**

La **Figura 3.9**, representa el diagrama de colaboración del caso de uso Gestionar Registros de Falla, en donde el Supervisor de Planificación se encarga de seleccionar el caso de uso que esta representado por la Interfaz Gestionar Registros de Falla, esto se hace una vez que la falla reportada no es solventada de manera inmediata, el Supervisor de Planificación debe realizar seguimiento a la misma verificando los recursos disponibles para su ejecución. Luego que la falla sea resuelta, invoca al Gestor Generar Reporte, este gestor almacena toda la información obtenida a través de las entidades correctivo, personal, falla, y equipo. La **Tabla 3.10** contiene los mensajes del Diagrama de Colaboración para el Caso de Uso Gestionar Registros de Falla, que contienen la secuencia de acciones.



LEYENDA

1: Realizar gestión de data equipo	11: Procesar actualización de data de equipo
2: Procesar data equipo	12: Obtener nueva data de componente
3: Solicitar registros de data equipo	13: Obtener nueva data de equipo
4: Activar registro de equipo	14: Obtener nuevos parámetros de datos
5: Procesar registro de data de equipo	15: Solicitar eliminación de data de equipo
6: Cargar datos de componente	16: Activar eliminación de equipo
7: Cargar datos de equipo	17: Procesar eliminación de data de equipo
8: Cargar parámetros de datos	18: Procesar eliminación de data de componente
9: Solicitar actualización de data de equipos	19: Procesar eliminación de data de equipo
10: Activar actualización de equipo	20: Procesar eliminación de parámetros de datos

Tabla 3.9. Mensajes del Diagrama de Colaboración del Caso de Uso Gestionar Data Equipo.

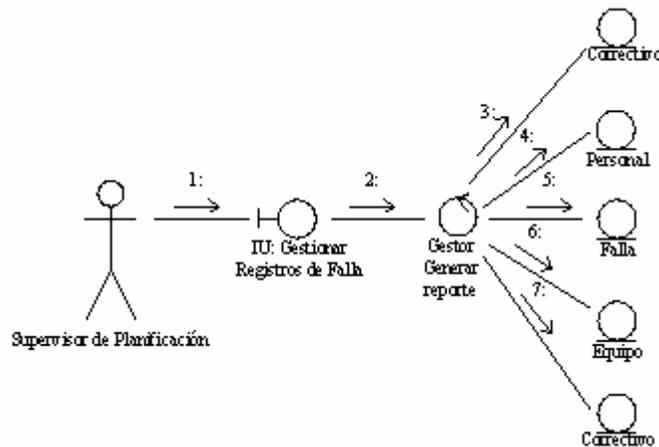


Figura 3.9. Diagrama de Colaboración para el Caso de Uso Gestionar Registros de Falla.



<u>LEYENDA</u>
1: Activar gestión de registro de falla
2: Mostrar reporte de falla
3: Obtener estatus de correctivo
4: Obtener datos de personal
5: Obtener datos de falla
6: Obtener datos de equipo
7: Obtener datos de tarea

Tabla 3.10. Mensajes del Diagrama de Colaboración del Caso de Uso Gestionar Registros de Falla.

- **Diagrama de Colaboración para el Caso de Uso Consultar Registros**

El diagrama de colaboración del caso de uso Consultar Registros, agrupa las principales actividades relacionadas con la consulta de los registros necesarios para el funcionamiento del sistema como se aprecia en la **Figura 3.10**, lo que permite consultar datos generales relacionados con la gestión de mantenimiento en la empresa; realizando esto mediante la Interfaz Consultar Registros, la cual se encuentra conformada por tres procesos, el Gestor Consultar Actividad que sería el encargado de consultar registros de las actividades, el Gestor Consultar Personal por medio del cual el Supervisor de Planificación puede consultar los registros del personal disponible, y por último, éste puede buscar la data de los equipos a través del Gestor Consultar Equipos, que posee las opciones de consulta por galpón, por área y por zona, obteniendo la consultas que fueron requeridas. La **Tabla 3.11** contiene los mensajes del Diagrama de Colaboración para el Caso de Uso Consultar Registros, que describirán la secuencia de acciones.

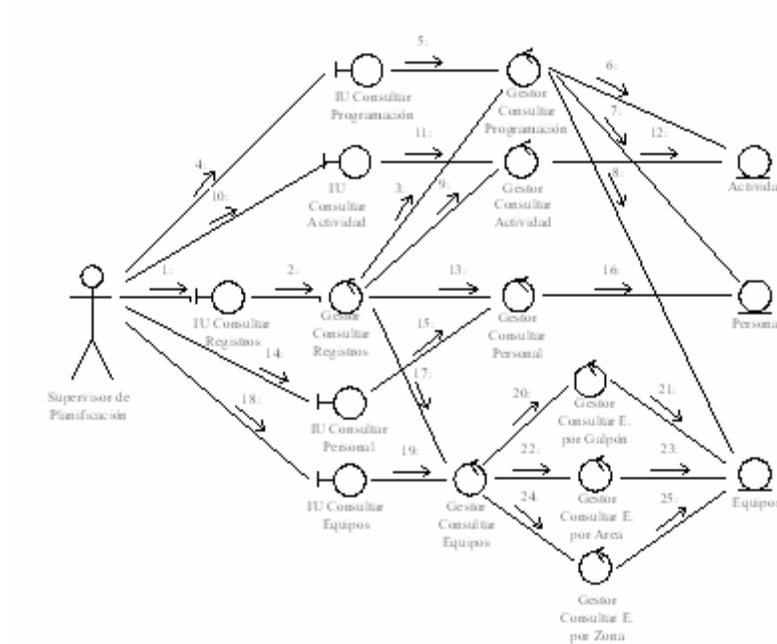


Figura 3.10. Diagrama de Colaboración para el Caso de Uso Consultar Registros.

- **Diagrama de Colaboración para el Caso de Uso Registrar Falla Nueva**

Una vez que se le sea informado al Supervisor de Planificación alguna falla inesperada solicita la Interfaz Registrar Falla Nueva para hacer la gestión del registro de la falla nueva, esto con la finalidad de almacenar el registro de la falla ocurrida como se puede observar en la **Figura 3.11**. La **Tabla 3.12** contiene los mensajes del Diagrama de Colaboración para el Caso de Uso Registrar Falla Nueva.



LEYENDA

- | | |
|--|--|
| 1: Realizar consulta de registros | 14: Activar consulta de personal |
| 2: Procesar consulta de registros | 15: Seleccionar personal a consultar |
| 3: Solicitar consulta de programación | 16: Obtener registro de personal |
| 4: Activar consulta de programación | 17: Solicitar consulta de equipo |
| 5: Seleccionar programación a consultar | 18: Activar consulta de equipo |
| 6: Obtener datos de actividad | 19: Procesar consulta de equipo |
| 7: Obtener datos de personal | 20: Seleccionar consulta de equipo por galpón |
| 8: Obtener datos de equipo | 21: Obtener datos de equipos |
| 9: Solicitar consulta de actividad | 22: Seleccionar consulta de equipo por área |
| 10: Activar consulta de actividad | 23: Obtener datos de equipo |
| 11: Seleccionar actividad a consultar | 24: Seleccionar consulta de equipo por zona |
| 12: Obtener registro de actividad | 25: Obtener datos de equipo |
| 13: Solicitar consulta de personal | |

Tabla 3.11. Mensajes del Diagrama de Colaboración del Caso de Uso Consultar Registros.

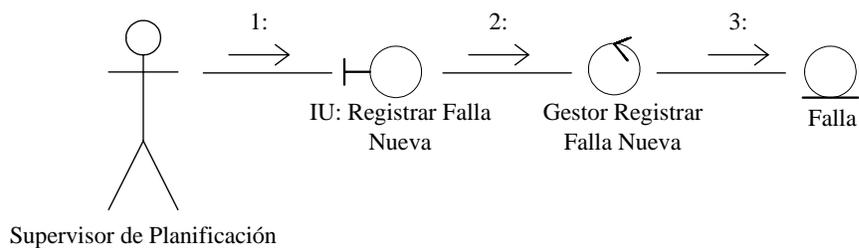


Figura 3.11. Diagrama de Colaboración para el Caso de Uso Registrar Falla Nueva.



<u>LEYENDA</u>
1: Solicita registrar falla nueva
2: Procesa el registro de la falla
3: Almacena registro de falla

Tabla 3.12. Mensajes del Diagrama de Colaboración del Caso de Uso Registrar Falla Nueva.

- **Diagrama de Colaboración para el Caso de Uso Gestionar Programación de Mantenimiento Preventivo**

En el diagrama de colaboración del caso de uso Gestionar Programación de Mantenimiento Preventivo de la **Figura 3.12**, el procedimiento se inicia cuando el Supervisor de Planificación selecciona la interfaz de usuario gestionar programación de mantenimiento preventivo, en donde la actividad principal se lleva a cabo mediante el Gestor Programar Mantenimiento, que a su vez permite asignar el mantenimiento, según el recurso humano y material disponible, para un período de tiempo determinado, obteniendo de esta manera la programación general de mantenimiento a ejecutarse, así como ver el cronograma de mantenimiento por cada equipo.

Para obtener el listado de materiales es necesario que el Supervisor de Planificación active el Gestor Generar Listado de Materiales requerido para una determinada programación de mantenimiento, el cual debe ser solicitado a otro ente para poder ejecutar los mantenimientos programados.



El Supervisor de Planificación también puede generar un listado de mantenimiento pendiente por efectuar a los equipos de manera tal que pueda realizar la programación para su futura ejecución a través del Gestor Listar Mantenimiento Pendiente de Equipos. Adicionalmente, con el Gestor Registrar Días no Laborables, el Supervisor de Planificación efectúa el registro de los días no laborables establecidos por la empresa.

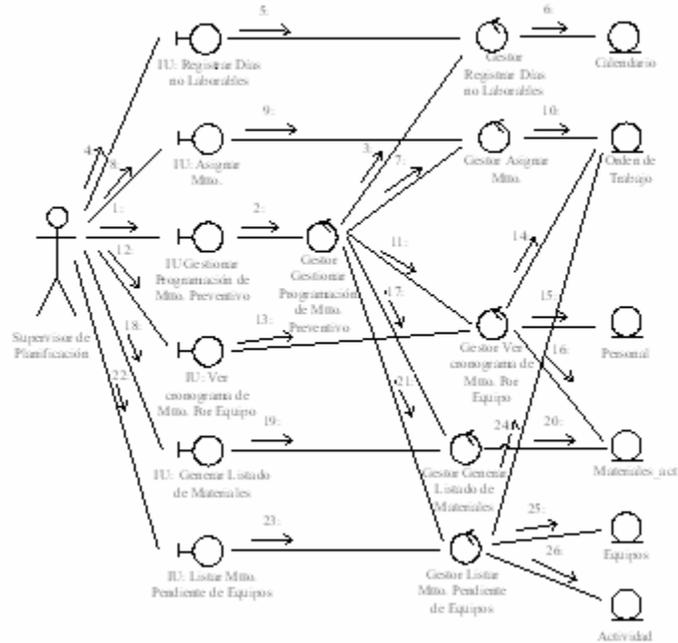


Figura 3.12. Diagrama de Colaboración para el Caso de Uso Gestionar Programación de Mantenimiento Preventivo.

La **Tabla 3.13** contiene los mensajes del Diagrama de Colaboración para el Caso de Uso Gestionar Programación de Mantenimiento Preventivo.



LEYENDA

1: Realizar gestión de programación de Mtto. Preventivo	14: Obtener datos de orden de trabajo
2: Solicitar programación de Mtto.	15: Obtener datos de personal
3: Solicitar registro de días no laborables	16: Obtener datos de materiales por actividad
4: Activar registro de días no laborables	17: Solicitar listado de materiales
5: Agregar registro de días no laborables	18: Activar generación de listado de materiales
6: Almacenar calendario laboral	19: Procesar listado de materiales
7: Definir asignación de Mtto.	20: Mostrar materiales por actividad
8: Activar asignación de Mtto.	21: Solicitar listado de Mtto. Pendiente por equipo
9: Procesar asignación de Mtto.	22: Activar la interfaz listar Mtto. Pendiente de equipo
10: Obtener datos de orden de trabajo	23: Procesar listado de Mtto. Pendiente de equipo
11: Solicitar ver cronograma de Mtto. Por equipo	24: Obtener datos de orden de trabajo
12: Activar ver cronograma de Mtto. Por equipo	25: Obtener datos de equipo
13: Mostrar cronograma de Mtto. Por equipo	26: Obtener datos de actividad

Tabla 3.13. Mensajes del Diagrama de Colaboración del Caso de Uso Gestionar Programación de Mantenimiento Preventivo.

- **Diagrama de Colaboración para el Caso de Uso Procesar Reportes**

En la **Figura 3.13**, tal como se observó en el diagrama del caso de uso Procesar Reportes (**ver Figura 3.2**), se describe en detalle el procedimiento que se realiza a través de la Interfaz Procesar Reportes. Este procedimiento comienza cuando el actor Supervisor de Planificación accede a la Interfaz Procesar Reportes, mediante la cual



podrá hacer la solicitud a las clases de control Generar Reporte de Mantenimiento Pendiente, Generar Reporte de Mantenimiento Efectuado y Generar Reporte de Fallas Registradas, con la finalidad de obtener los informes de los mantenimientos que no han sido efectuados, que ya fueron ejecutados y de las fallas que fueron registradas durante un período de tiempo establecido.

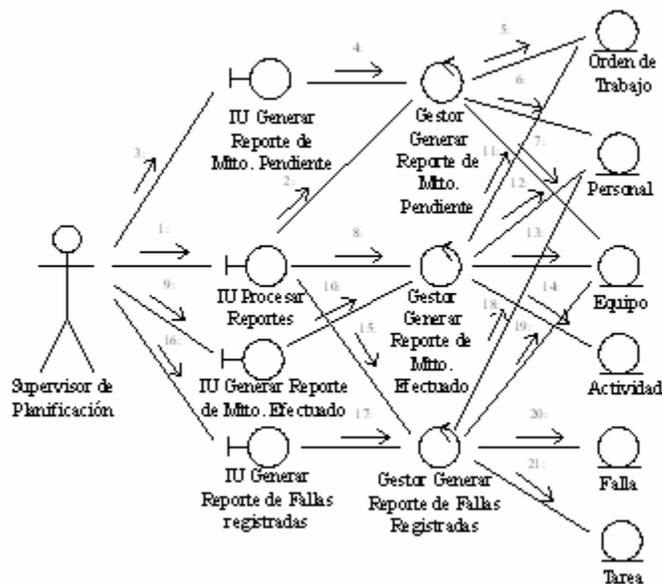


Figura 3.13. Diagrama de Colaboración para el Caso de Uso Procesar Reportes.

La **Tabla 3.14** contiene los mensajes del Diagrama de Colaboración para el Caso de Uso Procesar Reportes, que describirán la secuencia de acciones.



LEYENDA

1: Activar la interfaz procesar reportes	11: Obtener datos de orden de trabajo
2: Solicitar reporte de Mtto. Pendiente	12: Obtener datos de personal
3: Activar generación de reporte de Mtto. Pendiente	13: Obtener datos de equipo
4: Mostrar reporte de Mtto. Pendiente	14: Obtener datos de actividad
5: Obtener datos de orden de trabajo	15: Solicitar reporte de fallas registradas
6: Obtener datos de personal	16: Activar generación de reportes de fallas registradas
7: Obtener datos de equipo	17: Mostrar reporte de fallas registradas
8: Solicitar reporte de Mtto. Efectuado	18: Obtener datos de personal
9: Activar generación de reporte de Mtto. Efectuado	19: Obtener datos de equipo
10: Mostrar reporte de Mtto. Efectuado	20: Obtener datos de falla
	21: Obtener datos de tarea

Tabla 3.14. Mensajes del Diagrama de Colaboración del Caso de Uso Procesar Reportes.

- **Diagrama de Colaboración para el Caso de Uso Mostrar Ayuda**

En el diagrama de colaboración para el caso de uso Mostrar Ayuda de la **Figura 3.14**, el Supervisor de Planificación a través de la Interfaz de Usuario Mostrar Ayuda puede consultar información general sobre el manejo del sistema **SYMP**, y por medio del Gestor Mostrar Ayuda coordinar los procesos relacionados con dichas consultas, accediendo a esta información mediante el manejador de ayuda. La **Tabla 3.15** contiene los mensajes del Diagrama de Colaboración para el Caso de Uso Mostrar Ayuda, que describirán la secuencia de acciones.

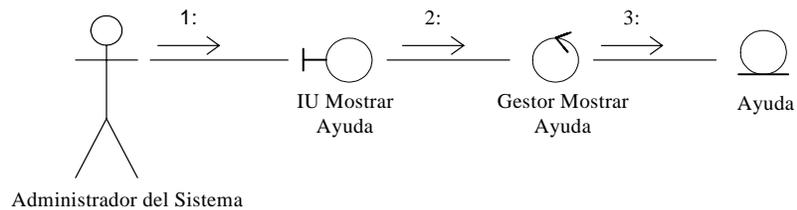


Figura 3.14. Diagrama de Colaboración para el Caso de Uso Mostrar Ayuda.

LEYENDA

- 1:** Activa interfaz mostrar ayuda
- 2:** Solicita mostrar ayuda
- 3:** Visualiza manejador de ayuda

Tabla 3.15. Mensajes del Diagrama de Colaboración del Caso de Uso Mostrar Ayuda.

3.4.2. Análisis de la Arquitectura

El propósito del análisis de la arquitectura es esbozar el modelo de análisis y la arquitectura mediante la identificación de paquetes de análisis. La descripción de la arquitectura contiene una vista de la arquitectura del modelo de análisis, que muestra sus artefactos significativos para la arquitectura como la descomposición del modelo de análisis en paquetes de análisis y sus dependencias.



3.4.2.1. Identificación de los Paquetes de Análisis

Los paquetes de análisis proporcionan un medio para organizar los artefactos del modelo de análisis en piezas manejables. Un paquete de análisis puede constar de clases de análisis, de realizaciones de casos de uso, y de otros paquetes de análisis (recursivamente).

Los paquetes de análisis deben ser cohesivos (fuertemente relacionados), y deben ser débilmente acoplados (con mínima dependencia).

Un paquete es una forma de agrupar clases en modelos grandes, pueden tener asociaciones de dependencias o de generalizaciones entre ellos.

La identificación de paquetes de análisis permitirá organizar el modelo de análisis en unidades que, básicamente, englobarán los requerimientos funcionales del sistema definidos con anterioridad como casos de uso en el modelo de casos de uso.

- **Paquete Datos del Sistema**

El paquete de análisis Datos del Sistema (ver **Figura 3.15**) contiene las clases de análisis (clases de entidad) que son consideradas básicas en el sistema, debido a que representan los objetos y datos que conforman la información del mismo.

Siguiendo las directrices del Proceso Unificado de Desarrollo de Software, se realizan una serie de actividades cuyos resultados constituyen la versión inicial del modelo de análisis (ver **Figura 3.16**), el cual contiene los paquetes Configurar Sistema, Gestionar Registro de Falla, Consultar Registros, Gestionar Programación de Mto. Preventivo, Registrar Falla Nueva, Procesar Gráficas, Procesar Reportes y Mostrar Ayuda.

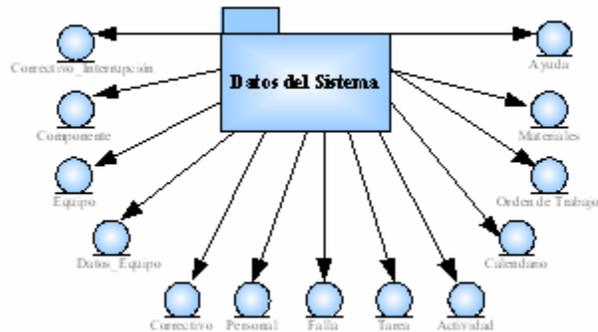


Figura 3.15. Paquete de Análisis de Datos del Sistema.

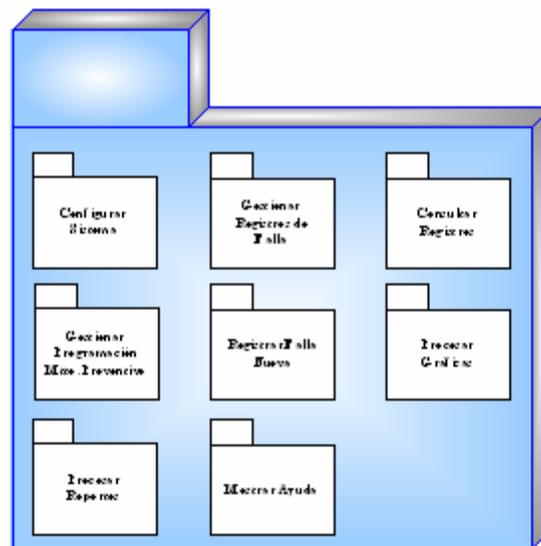


Figura 3.16. Diagrama de Paquetes de Análisis del Sistema SYMP.

3.5. DISEÑO

En la etapa de diseño se busca modelar el sistema y encontrar su forma (incluida la arquitectura) para que soporte los requisitos que se le suponen al sistema. Una entrada en el diseño es el resultado del análisis, representado por el modelo de análisis, el cual proporciona una comprensión detallada de los requisitos.



En esta fase, se dejan de un lado representaciones conceptuales como las utilizadas durante el análisis. En la fase de inicio, el objetivo principal es esbozar un modelo de diseño de la arquitectura candidata, es por ello que se dejará su desglose para la siguiente fase.

3.6. EVALUACIÓN DE LA FASE DE INICIO

En este capítulo se llevó a cabo la fase de inicio, la cual es primera fase del Proceso Unificado de Desarrollo de Software, en donde se estableció el concepto inicial del sistema, creando un esquema general del mismo, se identificaron y describieron los requisitos de sistema bajo estudio mediante los llamados casos de uso, con el fin de elaborar un conjunto de modelos iniciales que permitieron capturar la semántica y comportamiento del sistema.

La fase de inicio es la que reviste más importancia en todo el proceso, dado que es en ella donde se efectuó el análisis del sistema propuesto, se describió el contexto del sistema, representado a través del modelo de dominio, lo que permitió identificar los riesgos críticos que pondrían en peligro el desarrollo del proyecto y proponer una arquitectura candidata factible. Los casos de uso identificados con el flujo de trabajo de los requisitos fueron estudiados en detalle con diagramas de colaboración en el flujo de análisis, para representar las interacciones entre los objetos. Además, se identificaron y describieron las clases de análisis, haciendo la refinación de los casos de uso, mediante el uso de diagramas de clases de análisis, y se construyó el diagrama de paquetes de análisis para encapsular los casos de uso.

CAPITULO IV

FASE DE ELABORACIÓN

En el presente capítulo se contempla la fase de elaboración, la cual refleja una perspectiva del sistema completo, describe los elementos más importantes del mismo, y tiene como objetivo principal construir una arquitectura sólida que sirva de base para edificar el sistema.

La fase de elaboración tiene por objeto construir el núcleo central de la arquitectura, resolver los elementos de alto riesgo, definir los requerimientos y estimar los recursos necesarios. En esta fase, la arquitectura del sistema identifica las capacidades del sistema con las componentes de software, así como las relaciones entre estos componentes, se construye el núcleo central de la arquitectura incluyendo técnicas de diseño, mas bien abarcativas que de detalle, se identifican los procesos, capas de software, paquetes, subsistemas, estableciendo sus responsabilidades, se efectúa la clarificación de las interfaces internas (entre componentes) y externas (con los actores), refinándolas e incluyendo parámetros y valores de retorno.

Los casos de uso seleccionados para desarrollarse en esta fase permiten definir la arquitectura del sistema, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar del problema y comienza la ejecución del plan de manejo de riesgos, según las prioridades definidas en él. Al final de la fase, se determina la viabilidad de continuar el proyecto, dado que la mayor parte de los riesgos han sido mitigados, se obtiene una visión refinada, implementación iterativa del núcleo central de la aplicación, resolución de los riesgos más altos, identificación de nuevos requisitos y nuevos alcances, estimaciones más ajustadas.



4.1. ANALISIS

Para el flujo de trabajo análisis se analizarán mas a fondo los casos de uso identificados en los requisitos de la fase de inicio, planteando las clases de análisis de ellos, y estableciendo sus diagramas de colaboración y los paquetes de análisis, lo cual permitirá precisar y fundamentar la línea base de la arquitectura del sistema.

4.1.1. Diagrama de Clases de Diseño

Una clase es un artefacto de modelado que describe un conjunto de objetos que comparten los mismos atributos (conocimiento), operaciones (responsabilidad), relaciones (entrelazamiento), y semántica (relevancia).

Para modelar los aspectos estáticos de un sistema se hace uso del diagrama de clases, el cual es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

El diagrama de clases es el principal diagrama de diseño para un sistema. Los diagramas de clases de diseño muestran la funcionalidad del sistema mediante clases relacionadas entre sí por medio de líneas, haciendo la especificación de las instancias que participan en la relación.

Las clases de diseño definen los atributos y operaciones con que debe contar cada clase de análisis para llevar a cabo sus responsabilidades, así como las relaciones existentes entre ellas. Los atributos y operaciones se encuentran en su mayoría inmersos dentro de los diagramas y artefactos obtenidos en los diferentes flujos de trabajo, por lo que se necesita realizar una revisión exhaustiva de estos y agrupar las



distintas responsabilidades de las clases para ofrecer soporte a todos sus roles. Los diagramas de clases muestran la estructura estática de los casos de uso.

Una vez obtenido el resultado del flujo de análisis en la fase de inicio, en la **Figura 4.1** se ilustra el nivel de abstracción del sistema representado a través del Diagrama de Clases de Diseño del Sistema SYMP, definiendo las clases presentes en el sistema. La clase IU Principal SYMP representa toda la aplicación como un conjunto, la clase IU Configuración del Sistema es agregada a la clase IU Principal SYMP, y constituye una ventana de interfaz de usuario que se usa para iniciar la configuración, las clases Gestión de Data de Personal, Data de Actividad, y Data de Equipo son agregadas a la clase IU Configuración del Sistema, y representan el instrumento que permite tener actualizada la data del personal, actividades y equipos, respectivamente. La clase IU Programación de Mantenimiento Preventivo es agregada a la clase IU Principal SYMP, y constituye una ventana de interfaz de usuario que se usa para generar la programación de mantenimiento preventivo. Las clases Gestión Días no Laborables, Gestión Programación de Mantenimiento, Gestión Listar Mantenimiento Pendiente, y Gestión Generar Listado de Materiales son agregadas de la clase IU Programación de Mantenimiento Preventivo, que implican el procesamiento de los parámetros para obtener la programación de mantenimiento. La clase Gestión Programación de Mantenimiento es una abstracción de las subclases Gestión asignar Mantenimiento y Gestión ver Cronograma de Mantenimiento. Estas subclases permiten al usuario seleccionar asignar un mantenimiento o ver el cronograma de una programación asignada.

La clase IU Falla es agregada a la clase IU Principal SYMP, y constituye una ventana de interfaz de usuario que se usa para realizar el ingreso de la data de las fallas que se producen en los equipos, mediante el Gestor Registro de Falla Nueva, que es agregado a la clase IU Falla. La clase IU Correctivo es agregada a la clase IU Principal SYMP, y constituye una ventana de interfaz de usuario que se usa para



hacer un seguimiento a una determinada falla. La clase Gestión de Reporte de Falla es agregada a la clase IU Correctivo, y representa el ingreso, control y registro de la solvencia de la falla a consultar. La clase IU Consulta es agregada a la clase IU Principal SYMP, y constituye una ventana de interfaz de usuario que se usa para realizar las diversas consultas que sean requeridas por el usuario, las clases Gestión Consulta de Programación, Personal, Actividad, y Equipo son agregadas a la clase IU Consulta, y representan el instrumento que permite consultar la data de las programaciones, personal, actividades y equipos. La clase IU Reporte es agregada a la clase IU Principal SYMP, y constituye una ventana de interfaz de usuario que se usa para generar los diferentes reportes estipulados para llevar el control de los mantenimientos. Las clases Gestión de Reporte de Mtto. Pendiente, Fallas Registradas, y Mtto. Efectuado son agregadas a la clase IU Reporte, y representan las salidas del control sobre las programaciones.

La clase IU Gráficas es agregada a la clase IU Principal SYMP, y constituye una ventana de interfaz de usuario que se usa obtener las estadísticas del desempeño de la programación de mantenimiento. Las clases Gestión Procesar Programado, Ejecutado, y Prog. Vs Ejec. son agregadas a la clase IU Gráficas. La clase IU Ayuda es agregada a la clase IU Principal SYMP, y constituye una ventana de interfaz de usuario que el usuario utiliza a través de la clase Gestión de Ayuda para obtener el detalle de la información que solicite de un determinado proceso.

4.1.1.1. Diseño de las Clases

El las **Figuras 4.2, 4.3, 4.4 y 4.5** se muestran algunas de las clases establecidas y presentadas en la **Figura 4.1**, con la finalidad de identificar y presentar las operaciones y atributos que conforman dichas clases.

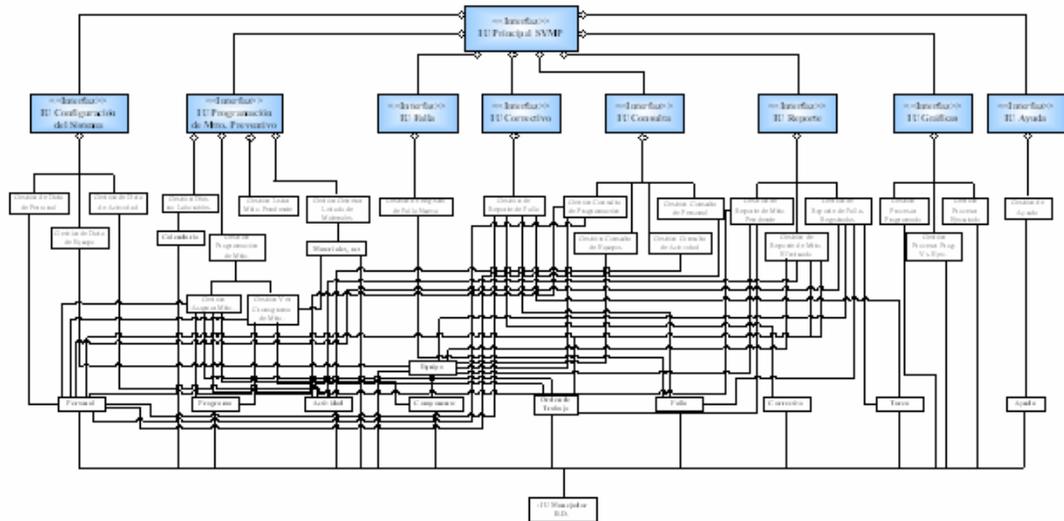


Figura 4.1. Diagrama de Cases de Diseño del Sistema SYMP.

En la **Figura 4.2** se muestra en detalle el diagrama de clases de diseño de la interfaz principal del sistema SYMP.

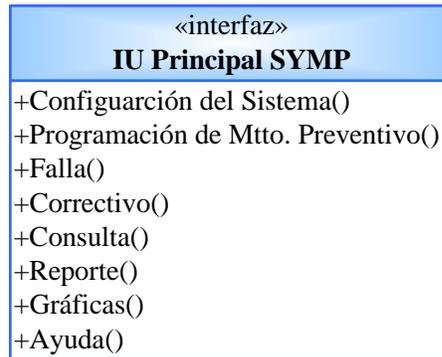


Figura 4.2. Diagrama de Clases de Diseño de la Interfaz Principal SYMP.

Mediante la **Figura 4.3** se puede observar el diagrama de clases de diseño detallado en la realización del caso de uso Gestionar Programación de Mantenimiento



Preventivo y la relación de las clases que forman parte del mismo, en el cual su clase principal es IU Programación de Mantenimiento Preventivo, a la cual se le es agregada las clases Gestión Días no Laborables, Gestión Listado de Materiales, Gestión Listar Mantenimiento Pendiente, Gestión Asignar Mantenimiento y Gestión Ver Cronograma de Mantenimiento, esta clase tiene funciones de filtrado de búsqueda de información, de manera que el usuario pueda ver los registros solicitados.

La **Figura 4.4** muestra la clase IU Falla, la cual tiene como función permitir al usuario efectuar el registro de la información correspondiente a las diversas fallas que ocurren a los componentes de los equipos, lo que permite llevar el control de la ocurrencia de las mismas.

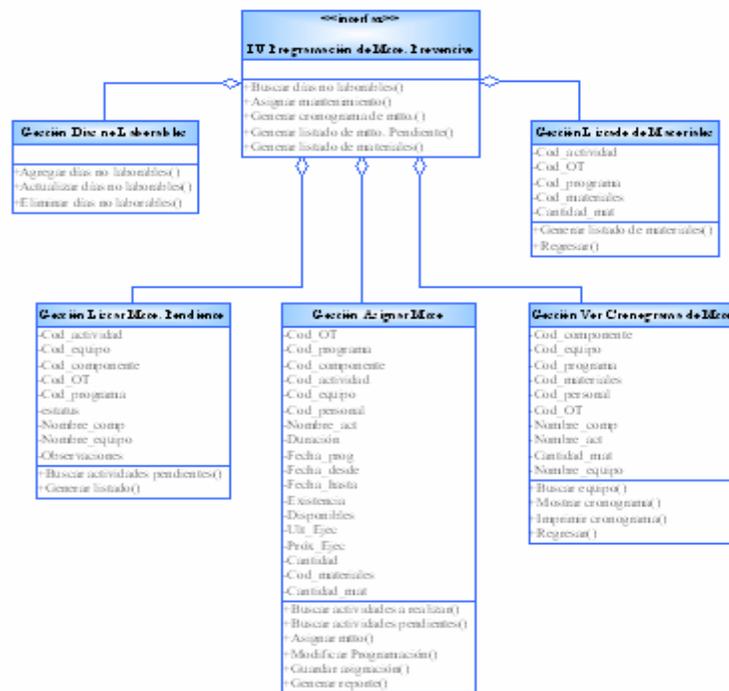


Figura 4.3. Diagrama de Clase de Diseño para el Caso de Uso Gestionar Programación de Mantenimiento Preventivo.

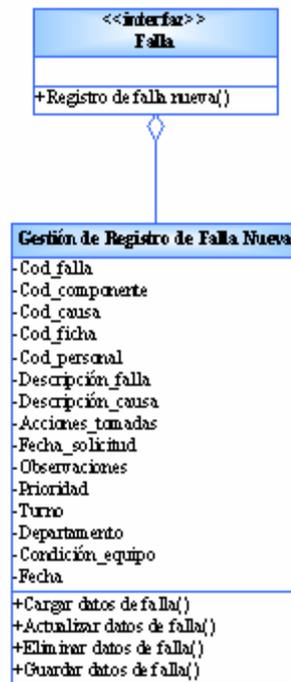


Figura 4.4. Diagrama de Clase de Diseño para el Caso de Uso Falla.

La clase IU Correctivo como se puede observar en la **Figura 4.5**, tiene funciones de búsqueda del correctivo aplicado a una falla antes registrada, lo cual permite conocer al usuario el estatus de una determinada falla, de igual manera permite al usuario modificar o completar la información, de acuerdo al avance del mismo.

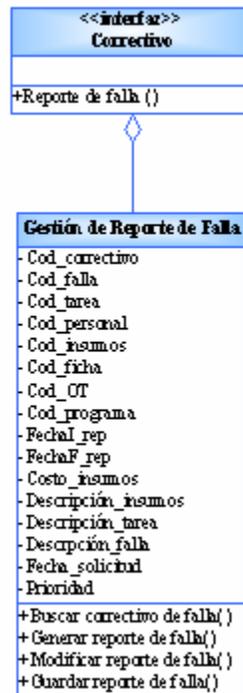


Figura 4.5. Diagrama de Clase de Diseño para el Caso de Uso Correctivo.

4.1.2. Diagramas de Secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase; contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos. Modelan la secuencia lógica, a través del tiempo, de los mensajes entre las instancias.

Los diagramas de secuencia representan una interacción, un conjunto de comunicaciones entre objetos organizados visualmente por orden temporal. Estos ayudan a conocer el desarrollo de cada proceso.



A continuación se presentan los diagramas de secuencia para dos de los casos de uso fundamentales para SYMP, **Figura 4.6** y **Figura 4.7**, en ellos se muestran los diferentes objetos representados por rectángulos y líneas punteadas verticales que indican la vida del objeto. Los objetos se comunican entre ellos mediante el envío de mensajes, representados por flechas horizontales orientadas desde el emisor hasta el receptor o destinatario de dicho mensaje.

La **Figura 4.6** muestra el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado, haciendo énfasis en el orden y el momento en que se envían mensajes a los objetos, correspondiente al caso de uso Gestionar Programación de Mantenimiento Preventivo, el cual procesa la programación de mantenimiento requerida por el usuario

Mediante la **Figura 4.7**, se muestra el diagrama de secuencia para el caso de uso Gestionar Data de Equipo , donde se puede observar los objetos que intervienen en dicho caso de uso, que forman parte de la arquitectura base del sistema SYMP.



4.1.3. Diagrama de Capas

El hardware y el software no son irreconciliables, sino que se complementan para definir el sistema informático. Las aplicaciones de software facilitan a los usuarios comunicar sus necesidades al computador sin tener que aprender a programar.

La **Figura 4.8** muestra un diagrama de capas donde se describe el sistema informático en términos de varios niveles desde el más externo (donde se encuentra el usuario) hasta el más interno (donde se encuentra el hardware). La información fluye entre las diferentes capas a nivel real y a nivel lógico.

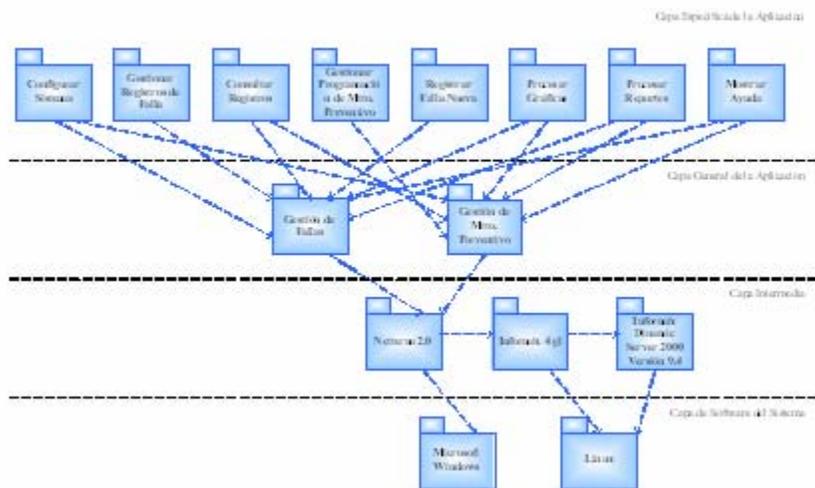


Figura 4.8. Diagrama de Capas del Sistema SYMP.

4.2. DISEÑO

En la fase de elaboración el diseño tiene como objetivo la obtención de la vista de la arquitectura del Modelo de Diseño, la realización física de los casos de uso el cual es un modelo de objetos que describe. Los diagramas de clases representan la estructura



estática del modelo de las clases, tanto la estructura interna como las relaciones con otros elementos.

4.2.1. Base de Datos

Uno de los objetivos principales de un sistema de información es contar no sólo con los recursos de información, sino también con los mecanismos necesarios para poder encontrar y recuperar estos recursos.

Una base de datos es un conjunto de datos organizados y estructurados pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

4.2.1.1. Diseño de las Tablas

Una tabla es una estructura lógica, que sirve para almacenar los datos de un mismo tipo, (desde el punto de vista conceptual). Almacenar los datos de un mismo tipo no significa que se almacenen sólo datos alfanuméricos. Desde el punto de vista conceptual, esto significa que cada entidad se almacena en estructuras separadas.

A continuación se explica brevemente el contenido de las tablas principales de la base de datos de **SYMP**, que fueron diseñadas siguiendo el modelo relacional y las formas normales, las cuales son resultantes del análisis de los requerimientos, estas son:

◆ TABLA EQUIPOS



La tabla equipos permite almacenar información sobre el código del equipo, su nombre, y el código de la zona, el cual es el vínculo de conexión con la tabla ubicación, para con ello obtener la ubicación exacta dentro de las instalaciones de la empresa. La descripción de los datos que forman parte de esta tabla se muestra mediante la **Tabla 4.1**.

Tabla 4.1. Descripción de la Tabla Equipos.

EQUIPOS		
Nombre Campo	Tipo	Descripción
Cod_equipo	Char 10	Identificador único del Equipo
Cod_zona	Char 30	Referencia a la Tabla Ubicación
Nombre_equipo	Char 50	Nombre del Equipo

◆ **TABLA DATOS_EQUIPO**

La tabla datos_equipo permite almacenar la información descriptiva de los diferentes equipos ubicados en las líneas de producción de una planta ensambladora de vehículos, tales como marca, descripción, modelo, entre otros. La descripción de los datos que forman parte de esta tabla se muestra mediante la **Tabla 4.2**.

Tabla 4.2. Descripción de la Tabla Datos_Equipo. (1/2)

DATOS_EQUIPO		
Nombre Campo	Tipo	Descripción
Cod_Datos	Integer	Identificador único del Código de Datos



Tabla 4.2. Descripción de la Tabla Datos_Equipo. (2/2)

DATOS_EQUIPO		
Nombre Campo	Tipo	Descripción
Marca	Char 20	Marca del Equipo
Descripción	Char 200	Descripción del Equipo
Modelo	Char 20	Modelo del Equipo
Serial	Char 20	Serial del Equipo
Catálogo	Char 5	Catálogo donde se encuentra el equipo
Operatividad	Char 15	Operatividad
Voltaje	Integer	Voltaje
Amperaje	Integer	Amperaje
Revoluciones	Integer	Revoluciones
Potencia	Char 20	Potencia
Descripción técnica	Char 200	Descripción técnica proporcionada por el fabricante
Observación	Char 100	Observaciones o comentarios sobre hechos relevantes
Ano_Instalación	Date	Año de instalación del equipo
Costo_Histórico	Decimal 15,2	Costo de adquisición del equipo



◆ TABLA COMPONENTE

La tabla componente permite almacenar la información descriptiva referente a los componentes de los equipos ubicados en las líneas de producción de una planta ensambladora de vehículos, tales como `cod_componente`, `nombre_componente`, entre otros. La descripción de los datos que forman parte de esta tabla se muestra mediante la **Tabla 4.3**.

Tabla 4.3. Descripción de la Tabla Componente.

COMPONENTE		
Nombre Campo	Tipo	Descripción
<code>Cod_componente</code>	Integer	Identificador único del Código del componente
<code>Cod_equipo</code>	Char 10	Referencia a la Tabla Equipos
<code>Cod_datos</code>	Integer	Referencia a la Tabla Datos_equipo
<code>Nombre_componente</code>	Char 50	Nombre asignado al componente

◆ TABLA DATOS_ACT

La tabla `datos_act` permite almacenar la información descriptiva de las diferentes actividades que pueden realizarse a los componentes de los equipos ubicados en las líneas de producción de una planta ensambladora de vehículos, tales como



nombre_act, frecuencia, duración, entre otros. La descripción de los datos que forman parte de esta tabla se muestra mediante la **Tabla 4.4**.

Tabla 4.4. Descripción de la Tabla Datos_act.

DATOS_ACT		
Nombre Campo	Tipo	Descripción
Cod_actividad	Integer	Identificador único de la actividad
Nombre_act	Char 20	Nombre de la actividad
Observaciones	Char 200	Observaciones o comentarios relevantes
Frecuencia	Smallint	Frecuencia de ocurrencia
Duración	Decimal 7,2	Especifica la unidad de duración de la actividad
Ult_Ejec	Date	Fecha de última ejecución de la actividad
Prox_Ejec	Date	Fecha de próxima ejecución de la actividad

◆ TABLA PROGRAMACION_ACT

La tabla programacion_act permite almacenar la información descriptiva acerca de la programación gestionada, para llevar a cabo las actividades a los componentes de los equipos ubicados en las líneas de producción de una planta ensambladora de vehículos, tales como cod_programa, estatus, fecha_ejec, entre otros. La descripción de los datos que forman parte de esta tabla se muestra mediante la **Tabla 4.5**.

**Tabla 4.5.** Descripción de la Tabla Programacion_act.

PROGRAMACION_ACT		
Nombre Campo	Tipo	Descripción
Cod_programa	Integer	Identificador único de la programación
Cod_actividad	Integer	Referencia a la Tabla Actividad
Estatus	Char 50	Especifica el estatus de la programación
Fecha_ejec	Date	Fecha de ejecución de la programación
Fecha_prog	Date	Fecha de programación
H\H_real	Decimal 7,2	Horas hombre reales utilizadas para la ejecución de la actividad
Observaciones	Char 200	Observaciones o comentarios

◆ **TABLA PERSONAL**

La tabla personal permite almacenar la información descriptiva del personal que lleva a cabo los mantenimientos, tales como descripcion_pers, existencia, disponibles, entre otros. La descripción de los datos que forman parte de esta tabla se muestra mediante la **Tabla 4.6**.

**Tabla 4.6.** Descripción de la Tabla Personal.

PERSONAL		
Nombre Campo	Tipo	Descripción
Cod_personal	Char 6	Identificador único del personal
Descripcion_pers	Char 50	Descripción del personal
Existencia	Smallint	Existencia del personal en el departamento
Disponibles	Smallint	Personal disponible en el departamento
H\laborables	Decimal 5,2	Horas que laboró el personal
Costo_unidad	Decimal 5,2	Costo por unidad d tiempo
Observaciones	Char 200	Observaciones o comentarios

◆ **TABLA FALLA GENERAL**

La tabla falla_general permite almacenar la información descriptiva de las distintas fallas que ocurren en los equipos ubicados en las líneas de producción de una planta ensambladora de vehículos, tales como descripcion_falla, cod_causa, condicion_equipo, entre otros. La descripción de los datos que forman parte de esta tabla se muestra mediante la **Tabla 4.7**.



Tabla 4.7. Descripción de la Tabla Falla_general.

FALLA_GENERAL		
Nombre Campo	Tipo	Descripción
Cod_falla	Integer	Identificador único de la falla
Cod_componente	Integer	Referencia a la Tabla Componente
Cod_ficha	Integer	Referencia a la Tabla Ficha
Cod_causa	Integer	Referencia a la Tabla Causa
Departamento	Char 30	Departamento donde ocurrió la falla
Descripcion_falla	Char 50	Detalle de la falla
Condicion_equipo	Char 25	Condición en la que quedó el equipo con ocurrencia de la falla
Observaciones	Char 200	Observaciones o comentarios

◆ **TABLA CORRECTIVO**

La tabla correctivo permite almacenar los datos referentes a los correctivos aplicados a las fallas, cuando se efectúa un mantenimiento correctivo, tales como fechaI_rep,



fechaF_rep, costo, entre otros. La descripción de los datos que forman parte de esta tabla se muestra mediante la **Tabla 4.8**.

Tabla 4.8. Descripción de la Tabla Correctivo.

CORRECTIVO		
Nombre Campo	Tipo	Descripción
Cod_falla	Integer	Identificador único de la falla
Cod_correctivo	Integer	Identificador único del correctivo
Cod_insumos	Integer	Referencia a la Tabla Correctivo_insumos
Cod_tareas	Integer	Referencia a la Tabla Tarea
FechaI_rep	Date	Fecha de inicio de la reparación
FechaF_rep	Date	Fecha de culminación de la reparación
Costo	Decimal 15,2	Costo de afectación por estar inactivo el equipo
Observaciones	Char 200	Detalles de la actividad

4.2.1.2. Modelo Relacional

El modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente, es el modelo relacional, utilizado en el proyecto bajo estudio;



mediante la base de datos relacional se fundamenta el uso de relaciones, que pueden considerarse en forma lógica como conjunto de datos llamados “Tuplas”, se concibe en cada relación una tabla compuesta por registros (las filas de una tabla), que representan las tuplas, y campos (las columnas de las tablas).

En este modelo, el lugar y la forma en la que se almacenen los datos no tienen relevancia, teniendo como ventaja que es fácil de entender y manejar para un usuario esporádico de la base de datos. La información puede ser recuperada y almacenada mediante “consultas”, que ofrecen una amplia flexibilidad para administrar la información.

Para almacenar la información ingresada a través de las aplicaciones que conforman el sistema **SYMP**, es necesario el empleo de una base de datos, en la cual se debe organizar dicha información.

Para el diseño de la base de datos se han tomado como referencia los diagramas de análisis y los diagramas de clases, como punto de partida para el diseño de las tablas y el manejo de la información requerida por el sistema, así como la relación existente entre dichas tablas.

Cabe destacar que el buen diseño de una base de datos permitirá entender las entidades de manera sencilla, así como realizar consultas y reportes desde diferentes puntos de vista, de una manera rápida y eficiente.

Para el diseño de la base de datos se hizo uso del modelo relacional, mediante asociaciones de entidades e interrelaciones. Al tener todas la tablas que representan todas las entidades que utilizará el sistema, se procede a realizar las relaciones exigiendo la integridad referencial y la actualización en cascada de los campos



En un sentido más amplio, puede definirse interfaz como el conjunto de comandos y/o métodos que permiten la intercomunicación del programa con cualquier otro programa o entre partes (módulos) del propio programa o elemento interno o externo.

El diseño de la interfaz de usuario parte de los requisitos que requieren los usuarios para su interacción con la aplicación. En el diseño de las interfaces, se debe tomar en consideración la homologación del funcionamiento de las diferentes interfaces, para facilitar el aprendizaje de manera rápida al usuario. A continuación se presentan los diseños de las interfaces principales **SYMP**.

- **Diseño de la Interfaz Principal del Sistema**

La **Figura 4.10** muestra la interfaz de usuario de la pantalla principal del Sistema de Mantenimiento Preventivo (**SYMP**), que permitirá al usuario iniciar la interacción con el sistema en todo momento.



Figura 4.10. Interfaz Principal SYMP.

- **Diseño de la Interfaz de Acceso al Sistema**

La **Figura 4.11** muestra el prototipo de diseño de la interfaz de usuario que le permitirá al usuario ingresar sus datos para con ello obtener el acceso al sistema, esta ventana permite introducir el nombre de usuario (login), así como la contraseña, tiene como función permitir el control de acceso al sistema.



Figura 4.11. Interfaz de Acceso al Sistema.

- **Diseño de la Interfaz de Usuario Principal**

La **Figura 4.12** muestra el diseño de la pantalla principal del sistema. Esta representa la pantalla inicial donde se ofrecen las funcionalidades, a las cuales el usuario puede acceder, tiene como función servir de menú del sistema.

La pantalla principal del sistema, muestra en el encabezado una barra de opciones en la esquina superior derecha, que le permite al usuario adaptar la el tamaño de la pantalla.



Figura 4.12. Interfaz de Usuario Principal de SYMP.

- **Diseño de la Interfaz Configurar Sistema**

En la **Figura 4.13** se puede observar la estructura que contiene la opción Configurar Sistema, presente en el menú principal del sistema **SYMP**, este se desglosa en tres opciones, las cuales permiten al usuario seleccionar si desea realizar la configuración de un equipo, una actividad o algún personal.

Estas opciones se desglosan en tres opciones adicionales, como lo muestra la **Figura 4.13**, donde se representa la opción Equipo, esta a su vez posee las opciones de registro, actualización y eliminación de data.



Figura 4.13. Despliegue de la Opción Configurar Sistema de Equipo en la Interfaz de Usuario del Sistema.

- **Diseño de la Interfaz de Programación**

El menú principal del sistema contiene la opción Programación, como se puede observar en la **Figura 4.14**, que muestra las diferentes opciones que se desglosan con la selección de esta opción., en donde se indica si se desea generar la programación de mantenimiento preventivo, los materiales a utilizarse en una determinada programación, ver los mantenimientos no han sido llevados a cabo, o visualizar el calendario laboral.



Figura 4.14. Interfaz de Usuario Mostrando la Opción de Programación de Mto. Preventivo.

- **Diseño de la Interfaz de Falla**

El sistema cuenta con una opción de registro de falla, que permite al usuario efectuar el ingreso de la data de la misma, con la selección de la opción falla, como lo muestra la **Figura 4.15**.



Figura 4.15. Interfaz de Usuario Mostrando la Opción Falso del Sistema.

- **Diseño de la Interfaz de Consulta**

La **Figura 4.16** muestra la estructura de la opción consulta, una vez que haya sido seleccionada en el panel de opciones del menú principal del sistema, que como se puede observar se despliega en cuatro opciones, las cuales indican si se desea realizar la consulta de alguna programación, equipo, actividad o personal.

Estas opciones se desglosan en opciones adicionales, como se observa en la **Figura 4.16**, en donde se despliega la opción Equipo, la cual indica si se desea efectuar la consulta del equipo ya sea por galpón, área o zona.



Figura 4.16. Despliegue de la Opción Consulta de Equipo en la Interfaz de Usuario del Sistema.

- **Diseño de la Interfaz de Reporte**

El sistema debe proyectar un conjunto de información organizada y simplificada, la cual se obtiene a través de la opción principal llamada reporte, como se puede observar en la **Figura 4.17**, cuyo despliegue indica las opciones de selección para generar el reporte de mantenimientos pendientes, el reporte de fallas registradas, así como el reporte de mantenimientos efectuados.



Figura 4.17. Interfaz de Usuario Mostrando la Opción de Reporte.

- **Diseño de la Interfaz de Gráficas**

La **Figura 4.18** muestra la estructura de la opción principal gráficas, que como se puede observar se despliega en tres opciones, las cuales indican si se puede generar la gráfica de los mantenimientos programados, la gráfica de los mantenimientos ejecutados o la gráfica de los mantenimientos programados vs. ejecutados.



Figura 4.18. Interfaz de Usuario Mostrando la Opción de Gráficas.

- **Diseño de la Interfaz Ayuda**

En la **Figura 4.19** se puede observar el despliegue del menú principal llamado ayuda, el cual indica la opción de Acerca de SYMP 1.0. Como se puede observar en la **Figura 4.20** se muestra el despliegue de la opción Acerca de SYMP 1.0, el cual contiene los autores del sistema, así como los derechos de autoría del mismo.



Figura 4.19. Interfaz de Usuario Mostrando la Opción de Ayuda.



Figura 4.20. Despliegue de la Opción Acerca de SYMP 1.0 en la Interfaz de Usuario del Sistema.

- **Mensaje de Error**

Los mensajes de error forman parte del diseño de una interfaz de usuario, para efectos del sistema SYMP se muestran sobre la pantalla en uso y de acuerdo a la funcionalidad, permitiendo al usuario del sistema hacer las correcciones del error, sin necesidad de ventanas adicionales de información al usuario.

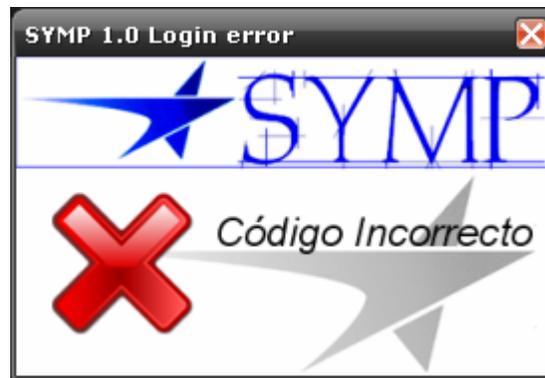


Figura 4.21. Interfaz de Usuario Mostrando Error del Sistema.

En la **Figura 4.21** se puede observar el mensaje de error emitido por el sistema cuando se introduce un código incorrecto.

4.2.3. Diseño de los Reportes Impresos

El reporte es el formato mediante el cual se definen los datos y la manera de presentación de los mismos, con el objeto de obtener los resultados que este deba reflejar, documentando las actividades y resultados relevantes.

El reporte es un resumen que identifica, para efectos de la presentación de la información exacta y oportuna, un formato útil a partir de la información que se maneja.

El objetivo principal de los reportes es permitir obtener de manera simplificada la información requerida por el usuario de manera periódica. Para garantizar que los reportes contengan sólo datos relevantes, se deben personalizar según las necesidades del usuario.



Los modelos de los reportes de la información almacenada, relacionados a los mantenimientos programados, se pueden observar en las **Figuras 4.22, Figura 4.23 y Figura 4.24**, donde se realizan los resúmenes específicos filtrando por características específicas la información requerida.

La **Figura 4.25** muestra el detalle de la salida del caso de uso Generar Listado de Materiales que se va a obtener una vez seleccionada la opción programación, y a su vez ver materiales.

		REPORTE DE MANTENIMIENTO PENDIENTE						
		SUPERVISOR:				FECHA:		
Item.	Programa	OT	Fecha	Actividad	Tipo	Estatus	Eficiencia	Observaciones
Tipo: P: Preventivo C: Correctivo O: Otros		Estatus: E: En curso N: No iniciado						

Figura 4.22. Reporte de Mantenimiento Pendiente.



		REPORTE DE MANTENIMIENTO EFECTUADO								
		SUPERVISOR:					DESDE: / /			HASTA: / /
Item.	Fecha	Descripción	Equipo	Tipo	Personal	Ficha	Tiempo	Materiales y Repuestos	Observaciones	
Tipo: P: Preventivo C: Correctivo O: Otros		Personal: M: Mecánico E: Electricista		Tiempo: h: Hora m: Minuto			Fecha de Emisión:			

Figura 4.23. Reporte de Mantenimiento Efectuado.

		REPORTE DE FALLAS REGISTRADAS							
		SUPERVISOR:					DESDE: / /		
Item.	Cod. falla	Descripción	Equipo	Personal	Ficha	Descripción Causa	Materiales y Repuestos	Observaciones	
Personal: M: Mecánico E: Electricista				Fecha de Emisión:					

Figura 4.24. Reporte de Fallas Registradas.



CONCLUSIONES

En base a los resultados obtenidos en el desarrollo de SYMP (Sistema de Mantenimiento Preventivo), una vez realizado el análisis y posterior diseño, se llegó a las siguientes conclusiones:

- ◆ El empleo del Proceso Unificado de Desarrollo de Software permitió efectuar el diseño de la arquitectura del sistema, de una manera robusta, sencilla y adaptable a cambios, es decir, las funcionalidades del sistema pueden crecer y el sistema diseñado podrá tolerarlo.
- ◆ La fase de inicio del proyecto jugó un papel fundamental, puesto que permitió obtener una noción clara del contexto del sistema, a través del Modelo de Dominio, logrando con ello la obtención de los requisitos necesarios para llevar a cabo el diseño.
- ◆ Los casos de uso fueron de gran importancia para representar las condiciones y requerimientos del sistema, logrando esto a través de una buena documentación y análisis, lo cual conllevó al esbozo de la visión general del funcionamiento del SYMP, mediante el diagrama de casos de uso.
- ◆ Los modelos de análisis permitieron obtener de una forma más específica la realización de la arquitectura inicial del sistema, partiendo de los casos de uso se generaron las clases de análisis y se modelaron las interacciones entre varios objetos mediante el flujo de mensajes.



- ◆ En la fase de elaboración se estableció la línea base de la arquitectura del SYMP, desarrollada a través de los modelos de casos de uso, análisis y diseño, y la elaboración de la base de datos, mitigando el impacto de los riesgos sobre el desempeño del sistema.

- ◆ Los diagramas de secuencia permitieron mostrar las interacciones entre los objetos que forman parte del SYMP, logrando establecer la secuencia que debe llevar cada caso de uso en función del tiempo.

- ◆ La base de datos diseñada permitirá el almacenamiento y recuperación, de manera eficiente, de toda la información relacionada con el proceso de mantenimiento a los equipos que intervienen en el proceso productivo, lo cual se logró haciendo uso del modelo relacional.

- ◆ Las interfaces del Sistema de Mantenimiento Preventivo (SYMP) fueron diseñadas tomando en consideración al usuario final, a fin de que interactúe con el sistema de forma rápida sencilla y amigable.



RECOMENDACIONES

- ◆ Se recomienda implementar a futuro el diseño del Sistema de Mantenimiento Preventivo expuesto en el presente proyecto, ya que solventaría la demora que presenta el Departamento de Mantenimiento para dar respuesta a la programación y manejo de los mantenimientos.
- ◆ Adiestrar a los usuarios que harán uso del SYMP, sobre las operaciones del mismo así como al personal que dará soporte técnico y realizará el mantenimiento.
- ◆ En caso de implementar el sistema propuesto, efectuar frecuentemente el respaldo de la base de datos del sistema, para garantizar la consistencia de los datos.
- ◆ El Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo Anzoátegui, debe generar planes que favorezcan la integración entre la industria y la universidad, con la finalidad de contribuir en la solución e innovación de problemas en la industria, y con ello promover la interacción con el estudiante, para que en conjunto con la industria lleve a cabo proyectos que nutran el conocimiento, y lo ayuden a enfrentarse con éxito al ámbito laboral.

BIBLIOGRAFÍA

1. Colaboradores de Wikipedia, (2008). *“Modelo relacional”*. Wikipedia, La Enciclopedia Libre,

http://es.wikipedia.org/w/index.php?title=Modelo_relacional&oldid=21444113.

2. MATA, C., (2007). *“Diseño de un Sistema de Información de los Parámetros de Funcionamiento de un Servidor de Almacenamiento del Centro de Cómputo de PDVSA-Refinación Oriente”*. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui. Venezuela.
3. ROJAS, A. y PRADO, L., (2007). *“Diseño de un Sistema de Información para el Seguimiento de las Actividades Asociadas con la Elaboración de Presupuestos de una Empresa Dedicada a la Fabricación de Productos de Aluminio”*. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui. Venezuela.
4. MARCANO, F. y ACEVEDO, Y., (2006). *“Diseño de un Software de Balance de Materiales para Yacimientos de Crudo y Gas, para PDVSA Oriente”*. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui. Venezuela.
5. WALES, J. y SANGER, L., (2006). *“Bases de Datos”*, Enciclopedia Libre, http://es.wikipedia.org/wiki/Base_de_datos.



6. WALES, J. y SANGER, L., (2006). “*Sistema Manejador de Bases de Datos DBMS*” Enciclopedia Libre, <http://es.wikipedia.org/wiki/DBMS>.
7. WALES, J. y SANGER, L., (2006). “*Interfaces de Usuarios*”. Enciclopedia Libre, http://es.wikipedia.org/wiki/Interfaz_de_usuario.
8. SÁNCHEZ, M., (2005). “*Diseño de un Sistema de Información para Automatizar Algunas de las Actividades Relacionadas con el Proceso de Producción de Crudo y Gas Desde el Yacimiento Hasta las Estaciones de Flujo, que se Realizan en una Empresa Petrolera, en Punta de Mata*”. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui. Venezuela.
9. CHIRINOS, A., (2005). “*Diseño de un Sistema de Información para la Automatización de las Actividades Administrativas del Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo de Anzoátegui, Basado en un Estudio de Sistemas Blandos*”. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui. Venezuela.
10. ROJAS, J., (2003). “*Programación Orientada a Objetos*” http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos.
11. CIBORRA, C. (2002). “*Labyrinths of Information*”, Oxford University Press http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n.
12. MORA, F., (2002). “*Lenguaje Unificado de Modelado*”, Universidad de Alicante. <http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF>.



13. WESLEY, A. BOOCH, G. RUMBAUGH, J. JACOBSON, I., (2002). “*El Lenguaje Unificado de Modelado*”, Guía básica de UML, <http://usuarios.lycos.es/oopere/uml.htm>.
14. JACOBSON, I., (2000). “*El Proceso Unificado de Desarrollo de Software*”, Editorial Pearson Educación, S.A., Madrid, España.
15. JACOBSON, I. BOOCH, G. RUMBAUGH, J., (2000). “*El Proceso Unificado de Desarrollo de Software*”. Pearson Addison-Wesley, http://es.wikipedia.org/wiki/Proceso_Unificado.
16. MARTIN, F. y KENDALL, S., (1999). “*UML Gota a Gota*”, <http://es.wikipedia.org/wiki/UML>.
17. MONTILVA, J., (1991). “*Desarrollo de Sistemas de Información*”, Universidad de los Andes, Consejo de publicaciones, Mérida, Venezuela.
18. RODRÍGUEZ, M., (1992). “*Base de Datos*”, 1^{era} Edición, Editorial McGraw-Hill, Madrid, España.
19. ELMASRI, R. y NAVATHE, S., (1997). “*Sistemas de bases de datos*”, 2^{da} Edición, Editorial Addison-Wesley Iberoamericana, Wilmington, Delaware, E.U.A.
20. ALHIR, S. (1998). UML in a nutshell: A Desktop Quick Referente. O'Really & Associates, Inc.
21. MONTILVA, J., (1991). “*Desarrollo de Sistemas de Información*”, Universidad de los Andes, Consejo de publicaciones, Mérida, Venezuela.



22. FÁBREGAS, LL., (1991). “*Sistemas de Información Planificación, Análisis y Diseño*”, Editorial Miro C.A., Caracas, Venezuela.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

TÍTULO	“Diseño de un Sistema de Información que Permita la Automatización de las Actividades Relacionadas con el Mantenimiento a Equipos Utilizados en Líneas de Producción en una Planta Ensambladora de Vehículos”
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Benevento Trinchese Mayerlin C.	CVLAC: 16.252.046 E MAIL: b_mayerlin@hotmail.com
Sánchez Aguilar Carmen Virginia	CVLAC: 16.479.157 E MAIL: virgisanchez2@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

Sistema de Información

Mantenimiento

Equipos

Líneas de Producción

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÁREA	SUBÁREA
Ingeniería de Sistemas	Sistemas de Información
	Software
	Base de Datos

RESUMEN (ABSTRACT):

El presente proyecto de investigación consiste en el diseño de un sistema de información que permite la automatización de las actividades relacionadas con el mantenimiento a equipos utilizados en líneas de producción en una planta ensambladora de vehículos. Este trabajo fue realizado para satisfacer los requerimientos del departamento de mantenimiento de dicha ensambladora; con el principal objetivo de realizar un estudio de las actividades llevadas a cabo en el área, se logró el diseño de un sistema automatizado que permita procesar, almacenar y generar la información referente al mantenimiento de los equipos utilizados en las líneas de producción de la empresa. El proceso de diseño incluyó el levantamiento de información de interés, mediante entrevistas realizadas al personal de la empresa; así como el diseño de sus interfaces y de su base de datos. Todo esto se realizó cubriendo las fases de inicio y elaboración del Proceso Unificado de Desarrollo de Software, mediante la utilización del Lenguaje Unificado de Modelado (UML), en la cual se formó un concepto general del sistema y se llevo a cabo el análisis del negocio; finalmente se desarrollaron los diagramas correspondientes.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Zulirais García	ROL	CA	AS	TU x	JU
	CVLAC:	10.299.576			
	E_MAIL	zzzuliii@hotmail.com			
	E_MAIL				
Manuel Carrasquero	ROL	CA	AS x	TU	JU
	CVLAC:	7.374.987			
	E_MAIL	manuelscm@hotmail.com			
	E_MAIL				
Héctor moisés	ROL	CA	AS	TU	JU x
	CVLAC:	8.277.670			
	E_MAIL	mbhenrique@hotmail.com			
	E_MAIL				
	ROL	CA	AS	TU	JU x
	CVLAC:				
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	02	26
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
TESIS.DISEÑO DE UN SISTEMA DE INFORMACIÓN QUE PERMITA LA AUTOMATIZACIÓN DE LAS ACTIVIDADES RELACIONADAS CON EL MANTENIMIENTO A EQUIPOS UTILIZADOS EN LÍNEAS DE PRODUCCIÓN EN UNA PLANTA ENSAMBLADORA DE VEHÍCULOS.doc	application/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H
I J K L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u
v w x y z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

_____ Ingeniero de Sistemas _____

NIVEL ASOCIADO CON EL TRABAJO:

_____ Pre-grado _____

ÁREA DE ESTUDIO:

_____ Departamento de Computación y Sistemas _____

INSTITUCIÓN:

_____ Universidad de Oriente, Núcleo Anzoátegui _____

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

De acuerdo al Artículo 44 del reglamento de Trabajos de Grados de la
Universidad de Oriente:

“Los trabajos de grado son de exclusiva propiedad de la universidad y
solo podrán ser utilizados a otros fines con el consentimiento del consejo
de núcleo respectivo, quien lo participará al consejo universitario”

AUTOR

AUTOR

AUTOR

TUTOR

JURADO

JURADO

POR LA SUBCOMISION DE TESIS