



UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACION

ESCUELA DE INGENIERIA DE SISTEMAS

Desarrollo de un Sistema de Control de No Conformidades

Tesis previa a la obtención del título de Ingeniero de Sistemas

Autor: Jackson Marcelo Serrano Tinitana

Director: Ing. Sist. Lenin Erazo Garzón.

Cuenca – Ecuador

2011

Dedicatoria

Dedico esta Tesis a Marcelo y Luisa Mis padres
Stalin y Dalton Mis Hermanos.
Mariela mi esposa y
A mí querida hija
Ayleen

Agradecimientos

Ing. Lenin Erazo Garzón, por compartirme sus conocimientos y dirigirme en este trabajo,
Ing. Gina Motta por darme la oportunidad de desarrollar este proyecto en la Institución.

Autoría

Todas las ideas expresadas en el presente trabajo son de exclusiva responsabilidad de su autor.

J. Marcelo Serrano T.

Resumen

La presente Tesis engloba el proceso de construcción de una aplicación, denominada Gestión de No Conformidades, definimos por No Conformidad el no cumplimiento de un requisito específico, según lo detallado en ISO 9001, 14001 y OSHAS 18001. Para realizar este proyecto hemos tomado como objeto de estudio a la Industria Cartonera Palmar (IN.CAR.PALM.).

El objetivo es construir una solución en entorno web basada en una arquitectura Tres Capas, apoyándose en:

- Especificación de Requisitos de Software (ERS).
- Los diagramas de Procesos.
- Las Herramientas Forms y Reports de Oracle con sus diferentes plataformas de desarrollo.
- El Gestor de Base de Datos Oracle 10g Release 10.1.2.
- La utilización de Recursos de Red
- La utilización del Servidor de Impresión

Al final se tendrá como resultado una aplicación que aproveche al máximo el conjunto de recursos utilizados y que permitirá el control y registro de las No Conformidades.

Abstract

This thesis deals with the building up of an application called Management of Non Conformances, taking as the subject of study the Palmar Paperboard Mill (IN.CAR.PALM.).

The goal of this work is the building up of a solution based on Triple architecture, supported by:

- Specification of Software Requisites (SSR)
- The process diagrams
- The Forms tools and Oracle reports in their different development platforms
- The Oracle 10 g Release 10.12 database agent
- The use of Network resources
- The use of the printing net server

At the end, the expected result will be an application that takes the maximum advantage of the set of resources used, which in turn will let the control and registration of Non Conformances.

Jackson Marcelo Serrano Tinitana

Translated by,




DEDICATORIA	2
AGRADECIMIENTOS	3
AUTORÍA	4
RESUMEN.....	5 ;ERROR! MARCADOR NO DEFINIDO.
ABSTRACT	5
CAPITULO 1.....	9
1.1 INTRODUCCIÓN	9
Definición de una No Conformidad.....	10
1.2 OBJETIVOS.....	11
1.2.1 Objetivo General.....	11
1.2.2 Objetivos Específicos.....	11
1.3 METODOLOGÍA ORACLE	12
1.4 CONCEPTOS DE PROGRAMACIÓN CON DEVELOPER SUITE.....	21
1.5 CONCEPTOS DE MODELOS DE BASE DE DATOS.....	22
1.6 INTRODUCCIÓN A LA ARQUITECTURA DE APLICACIÓN TRES CAPAS.....	28
CAPITULO 2.....	31
2.1 ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE	31
2.1.1 Introducción.....	31
2.1.2 Propósito.....	31
2.1.3 Ámbito del Sistema	31
2.1.4 Definiciones.....	32
2.1.5 Acrónimos.	32
2.1.6 Referencias.	32
2.1.7 Responsables del documento.....	32
2.2 Descripción General	33
2.3 Requisitos Específicos.....	33
2.3.1 Requisitos Funcionales.....	33
2.3.2 Descripción de Funcionalidades	33
2.3.3 Casos de Usos.....	34
CAPITULO 3.....	57
3.1 MODELO DE LOS DATOS	57
3.1.1 Modelo Entidad Relación	57
3.1.2 Diccionario de Datos.....	59
3.2 MODELO DE PROCESOS.....	73
3.2.1 Diagrama de Contexto	73
3.2.2 Diagrama de Descomposición	74
3.2.3 Diagrama de Flujo de Datos	75
CAPITULO 4.....	77
4.1 CREACIÓN DE LAS DEFINICIONES DE LA BASE DE DATOS.....	77
4.2 PROGRAMACIÓN CON HERRAMIENTAS DEVELOPER SUITE DE ORACLE.....	85
4.2.1 Objetos y Propiedades.....	87
4.2.2 Trigger	87
4.2.3 Elementos de Diseño.	89

4.2.4	Funciones Reglas y Comandos para el manejo de Fechas, Cadenas y Elementos de Control..	92
CAPITULO 5		95
5.1	CONFIGURACION DE SERVIDOR DE APLICACIONES.	95
5.2	INSTALACIÓN DE LA APLICACIÓN	106
5.3	PRUEBAS DE LA APLICACIÓN.	107
5.3.1	Tipos de Pruebas.....	107
5.4	CORRECCIÓN DE LOS ERRORES ENCONTRADOS	109
5.5	SISTEMA EN PRODUCCIÓN.	109
CAPITULO 6		110
6.1	MANUAL TÉCNICO.....	110
6.2	MANUAL DE USUARIO	110
6.2.1	Gestión de No Conformidades.....	110
6.2.2	Indicadores de Gestión de No Conformidad.....	114
6.2.2.1	Historial de No Conformidades.....	114
6.2.2.2	Consulta de No Conformidades	114
6.2.2.3	Indicador de No Conformidades por Rango de Fechas.....	115
6.2.2.4	Indicador de No Conformidades por Area	115
CAPITULO 7		¡ERROR! MARCADOR NO DEFINIDO.
	CONCLUSIONES Y RECOMENDACIONES	116
	BIBLIOGRAFIA.....	118
	ANEXOS.....	120

CAPITULO 1

INTRODUCCION

1.1 Introducción

Hoy en día muchas empresas, se han extendido en la adquisición de herramientas de desarrollo rápido, con el fin de automatizar los aspectos claves de todo el proceso de desarrollo para un sistema, desde el principio hasta el final e incrementar su posición en el mercado competitivo.

Oracle proporciona un conjunto de herramientas semi automatizadas y automatizadas que están desarrollando una cultura de ingeniería nueva para muchas empresas. Uno de los objetivos más importantes de este tipo de herramientas (a largo plazo) es conseguir que los programas tengan una generación automática desde el nivel de diseño.

Ahora bien con la aparición de las redes de ordenadores en empresas y universidades ha surgido en el mundo de la informática la tecnología Tres Capas. Son muchas de las organizaciones que ya cuentan con un número considerable de aplicaciones Tres Capas en operación.

El sistema a desarrollar está destinado para la Industria Cartonera INCARPALM que se encuentra en la ciudad de Machala. Esta institución se dedica a la fabricación de cajas, pads, tapas, fondos todos estos productos tienen como materia prima base el cartón.

El Sistema a realizar permitirá a la institución automatizar y controlar los registros manuales de las No Conformidades que se realizan en las Áreas de la institución antes mencionada.

Definición de una No Conformidad

De acuerdo a la definición de la Norma ISO 900:2000 en su literal (3.6.2), establece que una No Conformidad es el **incumplimiento de un requisito**.

Hay tres partes en una No Conformidad bien documentada:

- La **evidencia de auditoría** que soporta los rasgos del auditor.
- Un **registro del requerimiento** contra el cual una No Conformidad se detecta.
- **Enunciado de una No Conformidad**.

Aunque todas ellas necesitan ser abarcadas, en la práctica real, es la evidencia de auditoría la que primeramente debe ser identificada y documentada. Esto es porque un Auditor competente observará situaciones que **siente** que pudieren ser una No Conformidad potencial durante la auditoría aunque no este 100% seguro en ese momento. El auditor competente entonces documentará la evidencia de auditoría de la No Conformidad en sus **notas de auditoría**, antes de seguir las pistas adicionales para confirmar si realmente se trata de una No Conformidad.

Si **no** hay evidencia entonces **no** hay una No Conformidad. Si hay evidencia entonces esta debe ser documentada como una No Conformidad en vez de ser suavizada con otra clasificación, por ejemplo (observaciones, recomendaciones).

La evidencia de auditoría debe ser documentada con suficiente detalle para que la organización auditada pueda encontrar y confirmar exactamente lo que el auditor ha observado.

La siguiente cosa que el auditor debe hacer es identificar y registrar el requisito específico que no ha sido cubierto, por lo que si el auditor **no puede** identificar el requisito que se incumple, entonces el auditor **no puede** levantar una No Conformidad.

1.2 Objetivos.

1.2.1 Objetivo General.

Desarrollar un Sistema de Control de No Conformidades aplicado a la Industria Cartonera INCARPALM para automatizar el registro y seguimiento de las mismas, en las diferentes áreas de la Institución.

1.2.2 Objetivos Específicos.

- Adquirir nuevos conocimientos sobre las áreas tecnológicas desconocidas por parte del desarrollador del proyecto que sirvan de complemento para la formación académica y profesional.
- Identificar la información de los formularios utilizados para el registro de las No Conformidades, a fin de construir un modelo sólido para el desarrollo de la aplicación.
- Adquirir conocimientos sobre la configuración de servidores y los lenguajes de programación, teniendo la facilidad de probar en diferentes instancias.
- Lograr experiencia en el manejo de herramientas Oracle.
- Proporcionar al auditor una fuente de consulta siempre disponible y a toda hora, con la manipulación de la información en tiempo real.
- Proporcionar mecanismos de almacenamiento seguro de información, mediante técnicas de backup's y respaldos.

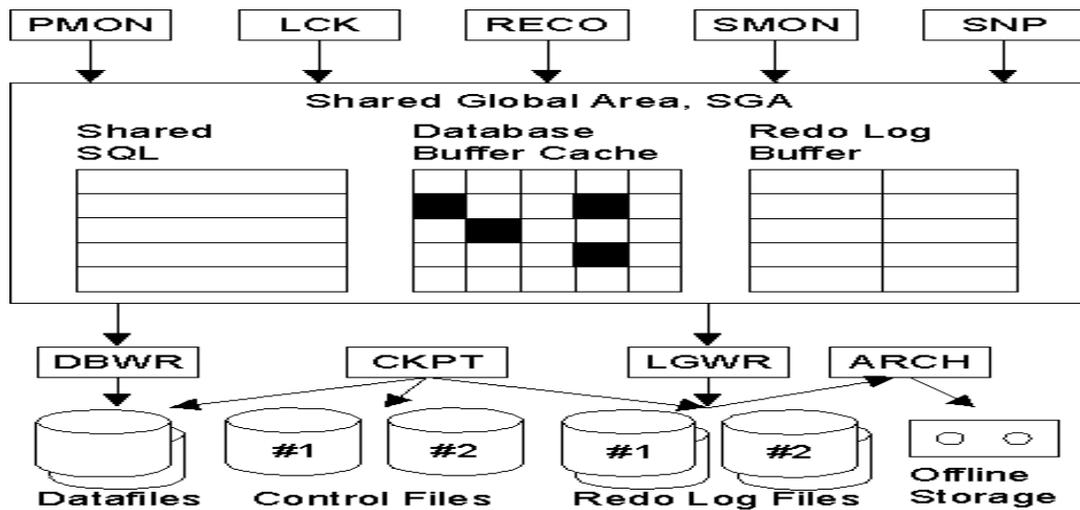
1.3 Metodología Oracle

Oracle es un sistema de gestión de base de datos objeto-relacional desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando propiedades como:

- Soporte de transacciones
- Estabilidad
- Escalabilidad
- Soporte Multiplataforma

Arquitectura Servidor Oracle



La Arquitectura de Oracle tiene tres componentes que son:

a) Estructuras de Memoria b) Archivos. c) Procesos.

Estructuras de Memoria.

Existen dos clases de memoria, una de ella compartida por todos los usuarios conectados y otra dedicada al trabajo de cada uno de ellos.

El Area Global del Sistema o SGA(System Global Area), es el área compartida por todos los usuario y se divide en tres partes:

- **Fondo Común Compartido (Shared Pool)** en ella se mantiene el diccionario de datos y las áreas compartidas de todas las ordenes SQL que se solicitan para su procesamiento.
- **Area de Memoria Rápida (DataBase Buffer Cache)** donde se mantiene los datos traídos por las ordenes SQL de los usuarios conectados a la Base de Datos.
- **Area de Registros de Rehacer (Redo Log Buffer)** aquí se registran los cambios hechos a la base de datos.

Por cada sesión de usuario, se crea también en memoria un área específica llamada Area Global de Programa PGA; esta área no se comparte con las otras sesiones de usuario.

Archivos

Los archivos que maneja Oracle se dividen en cuatro grupos:

- Archivos de Datos.
- Archivos de Control.
- Archivos de Rehacer.
- Archivos Fuera de Línea.

Archivos de datos (Data Files): Sirven para el almacenamiento físico de las tablas, índices, o agrupamientos (clúster) y procedimientos. Estos archivos son los únicos que contienen los datos de los usuarios de la base datos así como el diccionario de datos.

Las unidades lógicas más grandes manejadas por ORACLE para el almacenamiento de los datos son llamadas **Tablespace**, que le permiten manejar o controlar el espacio en los discos.

Tablespace: Pertenecen sólo a una base de datos y sirven para agrupar los datos de la base de datos. Cada Tablespace está formado físicamente por uno o más archivos de datos. Están divididos en 0 o más segmentos. Se pueden visualizar en línea o fuera de línea y pueden ser activados en sólo lectura o en lectura / escritura.

Segmento: Sirve para almacenar las estructuras lógicas de la base de datos (tablas, índices). Un Tablespace se compone de uno o más segmentos. Pero el mismo segmento no puede estar en más de un Tablespace.

Extensiones: División que se hace a cada segmento. El DBA puede añadir o quitar extensiones a los segmentos a fin de hacer que ganen o pierdan espacio.

Archivos de control (Control Files): Tienen la descripción física y dirección de los archivos de la base de datos y de los archivos de rehacer para el arranque correcto de la base de datos, en estos archivos se especifican cuales datafiles conforman la base de datos para poder tener acceso a los datos o para poder recuperar la base de datos ante una falla.

Archivos Rehacer (Redo Log Files): Tienen los cambios hechos a la Base de Datos para la recuperación ante fallas, es decir almacenan datos recuperables en caso de error grave. Poseen los valores antes de la ejecución los valores, la orden ejecutada y opcionalmente el valor después de la transacción. El principal propósito de estos archivos es servir de respaldo de los datos en la memoria RAM.

Además se utilizan otros archivos de forma auxiliar

Archivos Fuera de Línea (Archived Files): Son archivos temporales donde se guarda información vieja de los archivos de rehacer, muy convenientes para los respaldos de base de datos.

Archivos de parámetros (Parameter Files): Que definen algunas características de una instancia Oracle.

Archivos de contraseñas: Que sirven para autenticar a los usuarios.

Copias de archivos rehacer: Utilizadas para la recuperación de datos.

En un servidor se pueden crear varias instancias, pero se recomienda solo una porque cada instancia consume muchos recursos. Por cada instancia de Oracle se recomienda que se tenga una sola Base de Datos.

Procesos

Los procesos son programas que se ejecutan para permitir el acceso a los datos. Los procesos se cargan en memoria y son transparentes para los usuarios, estos se clasifican en tres grupos:

Procesos de Base, Procesos de Usuario y Procesos Servidores.

Procesos de Base o de Soporte: Los procesos de Base (**background**), son los que se encargan de traer los datos desde y hacia el SGA; mejorando el desempeño al consolidar las tareas que son impartidas por todos los usuarios. Cada proceso de base tiene su propia área de memoria. Los procesos de base son los siguientes:

DBWR (DataBase Writer): Se encarga de manejar los buffers de memoria cache para que los procesos del usuario siempre encuentren uno disponible. Es un proceso obligatorio que además escribe los bloques de datos modificados por los usuarios, en los archivos de datos que componen la B.D. cuando el proceso LGWR le envía el mensaje de hacerlo.

LGWR (Log Writer): Este proceso escribe datos desde el SGA a los archivos rehacer(Redo Log Files) que sirven en caso de fallas en la instancia, este proceso es obligatorio y es el único capaz de escribir y leer en estos archivos. El proceso de llenado de estos archivos es circular, por lo tanto antes de empezar a sobrescribir en uno de ellos, se marca un punto de verificación y LGWR envía una orden de escritura en los datafiles al proceso DBWR.

LCKn, lock (Lock Process): Este tipo de bloqueo es un proceso opcional, efectúa los bloqueos entre instancias, en caso de ambientes con servidores paralelos (hasta con 10 servidores).

CKPT (Check Point): El punto de comprobación es un proceso opcional que ocurre cuando los usuarios conectados a la base de datos, hacen solicitudes de exámenes de datos.

SNPn (Snapshot Process): Se encarga de refrescar los snapshots o réplicas de tablas que se usan principalmente en ambientes distribuidos.

SMON (System Monitor): Permite recuperar la instancia de la B.D. en caso de una caída fatal, Además limpia los segmentos temporales no utilizados y compacta los huecos libres contiguos en los ficheros de datos. Este proceso se despierta regularmente para comprobar si debe intervenir.

PMON (Process Monitor): Este proceso restaura las transacciones no validadas de los procesos de usuario que abortan, liberando los bloqueos y los recursos de la SGA. Asume la identidad del usuario que ha fallado, liberando todos los recursos de la BD que estuviera utilizando, y anula la transacción cancelada. Este proceso se despierta regularmente para comprobar si su intervención es necesaria.

RECO (Recovery): Recupera ante fallas en ambientes distribuidos.

ARCH (Archive): Copia los registros de rehacer de la RAM en archivos de datos que permitan la recuperación cuando se presentan fallas en los medios magnéticos.

Procesos de Usuario

Cuando un usuario se conecta a la base de datos, se crea un proceso de usuario que se encarga de ejecutar el código de aplicación del usuario y manejar el perfil del usuario con sus variables de ambiente. Los procesos de usuarios no se pueden comunicar directamente con la base de datos, únicamente lo hacen a través de procesos servidores.

Procesos Servidores

Estos procesos ejecutan las ordenes SQL de los usuarios y llevan los datos del **Database Buffer Cache**, para que los procesos del usuario puedan tener acceso a los datos.

Instancia de Oracle

Se denomina instancia al conjunto de estructuras de memoria y procesos de fondo que acceden los archivos de base de datos.

El sistema de Base de Datos Oracle cuando inicia sigue los siguientes pasos que se detallan a continuación.

- **Iniciar la Instancia:** Oracle lee el archivo de parámetros y configura la instancia con base en ellos, en ese momento se crea el SGA y se activan los procesos de base; pero aún no se puede hacer nada. Es como encender un vehículo en neutro, listo para empezar a trabajar.
- **Montar la Base de Datos:** Consiste en preparar el sistema para su uso, trayendo a la RAM el diccionario de datos; es como poner el sistema en primera, listo para recibir algunas órdenes del DBA.
- **Abrir la Base de Datos:** En este momento se abren los archivos y los usuarios ya pueden tener acceso a los datos.

Los parámetros de la instancia se definen en el archivo INIT.ORA, entre ellos se pueden mencionar:

Db_block_buffers: Numero de bloques de base de datos en el SGA. Existirá un buffer por cada bloque.

Db_block_size: Tamaño del bloque de la base de datos.

Shared_pool_size: Tamaño del área compartida **shared pool**, en bytes.

Además allí se especifica el número de usuarios concurrentes y los nombres de Archivos de Control para la base de datos.

Estos parámetros se pueden ajustar durante la etapa de afinamiento, porque ellos inciden en el desempeño del sistema. Algunos parámetros son específicos a una base de datos y por lo tanto deben ser cambiados antes de crear una base de datos.

Se incluyen en estos:

Database_Name: Nombre de la Base de Datos

Db_block_size: Tamaño del bloque.

Conexión con una instancia de Oracle: De todos los servicios creados durante la instalación Oracle solo interesa básicamente dos:

- El servicio relacionado con la instancia y la Base de Datos, cuyo nombre tiene la siguiente estructura **OracleServiceXXX**, donde **XXX** identifica el nombre de la instancia.
- El servicio relacionado con la disponibilidad del servidor para el acceso remoto, el nombre de este servicio es **Listener**, Este servicio reside en memoria y se encarga de recibir las llamadas que llegan a la base de datos desde la red, y de pasárselas a esta.

Estos dos servicios deben estar ejecutándose y su verificación se puede ver en la ventana de servicios.

Conexión Local Utilizando SQL*Plus: Es la conexión que permite al usuario comunicarse con el servidor.

Una vez instalado Oracle en la Capa de Datos se trabaja con procedimientos, funciones, desencadenantes y paquetes que hacen más eficiente el desarrollo de las aplicaciones.

Transacciones

Los cambios en la base de datos no son guardados hasta que tras una serie de instrucciones se decide llevar a cabo esos cambios. Hasta ese momento todo lo realizado se toma como provisional. Un fallo en la máquina permitiría invertir los cambios.

Una transacción es un conjunto de operaciones SQL que forman una unidad de trabajo.

Comienza cuando una persona se conecta y de ahí hasta que ejecuta la instrucción commit (ejecutar la transacción) o rollback (anular la transacción). La anulación deja la base de datos en el estado anterior al comienzo de la transacción. Tras un commit o un rollback comienza la siguiente transacción.

En Oracle se admite además el uso de puntos de ruptura (checkpoints) para almacenar valores intermedios y volver a cualquier de ellos si interesa. Pero esto ralentiza excesivamente el sistema.

Usuarios

Los usuarios son las personas que acceden de una forma o de otra a la base de datos. Cada usuario tiene una vista determinada de la base de datos. Hay varios conceptos sobre los usuarios a tener en cuenta.

Privilegios

Controlan el permiso que posee un usuario de ejecutar una determinada instrucción SQL.

Un usuario que quiera crear una tabla, deberá tener el privilegio (o permiso) adecuado para ello.

Además se pueden colocar privilegios en los objetos, es decir, un usuario propietario de una tabla puede otorgar privilegios a esa tabla (se trataría de un privilegio a nivel de objeto) para que haya otros usuarios que la puedan usar.

Rol

Son agrupaciones de privilegios que facilitan la tarea de gestionar a los usuarios. Así cuando una serie de usuarios van a tener los mismos privilegios, se crea un rol que contenga esos privilegios y a esos usuarios se les asigna el rol.

Oracle proporciona varios roles ya preparados, por ejemplo el rol DBA da privilegio absoluto a un usuario.

Esquemas

Los esquemas están asociados a los usuarios. Agrupan los objetos lógicos que pertenecen al usuario. Es decir es el conjunto de tablas, vistas, sinónimos, instantáneas, enlaces de base de datos, procedimientos y funciones, paquetes.

Cada usuario tiene su propio esquema y, en principio, un usuario no tiene acceso a los elementos de otro usuario, salvo que sea un administrador o que otro usuario ceda el privilegio de utilización de uno o más de sus objetos al resto de usuarios.

1.4 Conceptos de Programación con Developer Suite

Entre las Herramientas de Oracle para desarrollar aplicaciones tenemos:

Oracle Forms. Permite crear aplicaciones visuales sobre bases de datos de Oracle

Oracle Reports. Asistente para la producción de informes

Entre los principales objetivos y acciones buscadas e implementadas debemos señalar algunas características como:

- Aumentar la productividad de las áreas de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la calidad del software desarrollado.
- Reducir tiempos y costos de desarrollo y mantenimiento del software.
- Mejorar la gestión de dominio sobre el proyecto en cuanto a su planificación ejecución y control.
- Mejorar el diccionario de datos, reduciendo la dependencia de uso.
- Automatizar:
 - El desarrollo del software.
 - La documentación
 - La generación de código.
 - La revisión de Errores.
 - La Gestión del Proyecto.
- Permitir
 - La reutilización del Software.
 - La portabilidad del Software.
 - La Estandarización de la Documentación.

1.5 Conceptos de Modelos de Base de Datos.

Desde su aparición en la década de 1950, estas aplicaciones se han hecho imprescindibles para sociedades industriales. La primera base de datos para Pc data de 1980; era el dBase II, desarrollado por el Ingeniero estadounidense Wayne Ratliff. Desde entonces su evolución ha seguido paralela a la que ha experimentado el software, y hoy existen desde bases de datos para una utilización personal hasta bases de datos corporativas, soportadas por grandes sistemas informáticos.

Para definirlo de otra manera tenemos que: Una Base de Datos es un conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su uso posterior. En este sentido una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

Otra definición: Sistema distribuido entre múltiples procesadores donde hay clientes que solicitan servicios y servidores que los proporcionan.

En informática existen los Sistemas Gestores de Base de Datos(SGBD) (DBMS Data Base Manager System), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los Sistemas Gestores de Bases de Datos se estudian en informática. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

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

- **Base de datos estáticas**
Estas 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**
Estas 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 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 al contrario estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias).
- **Bases de datos numéricas**
Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio.

- Base 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.
- Directorios
Un ejemplo son las guías telefónicas en formato electrónico.
- Banco de imágenes, audio, video, multimedia, etc.
Almacenan información en distintos formatos.
- Bases de datos o “bibliotecas” de información Biológica
Son bases de datos que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas.

Modelos de Bases de Datos

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 donde se guarda la información), así como 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 bases de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos son:

- Bases de Datos Jerárquicas
Estas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo de 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 conocen como hojas.
Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

- Bases de Datos de Red

Este 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 a significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

- Bases de Datos Relacionales

Este 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 considerarse como un nuevo Paradigma en los modelos de bases 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 almacenan 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 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 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 específica puede modificarse separadamente sin afectar a 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.

- Bases de Datos Documentales

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

- Base de Datos Deductivas

Tiene la diferencia 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 bases de datos lógicas, a raíz de que se basan en lógica matemática.

Gestión de Bases de Datos Distribuidas

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.”

Ventajas de las Bases de Datos

- Independencia de datos y tratamiento.
- Cambio en datos no implica cambio en programas y viceversa
- Coherencia de resultados.
- Reduce redundancia.
- Acciones lógicamente únicas.
- Se evita inconsistencia de los datos.
- Mejora en la disponibilidad de datos.
- Guardamos descripción (Idea de catálogos).
- Cumplimiento de ciertas normas.
- Restricciones de seguridad.
- Accesos(Usuarios a datos).
- Operaciones (Operaciones sobre datos).
- Más eficiente la gestión de almacenamiento.

1.6 Introducción a la Arquitectura de Aplicación Tres Capas.

Una arquitectura es un conjunto de reglas, definiciones, términos y modelos que se emplean para producir un producto.

En la arquitectura a tres capas (three-tier) añadimos una nueva capa entre el cliente y el servidor donde se implementa la lógica de la aplicación. De esta forma el cliente es básicamente una interface, que no tiene por qué cambiar si cambian las especificaciones de la base de datos o de la aplicación; queda aislado completamente del acceso a los datos.

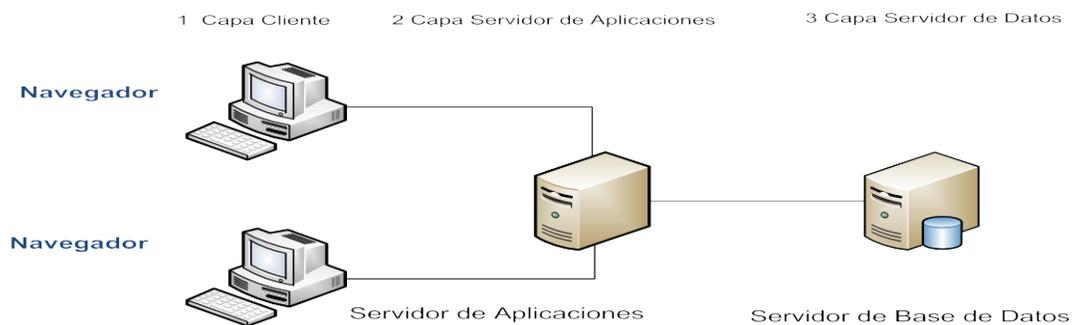
Así un applet de Java se carga en el navegador del cliente y se comunica con un servlet que corre en la máquina servidor; o bien accedemos a la base de datos a través de un formulario Web. El servlet establece una conexión a la base de datos mediante JDBC.

Capas y Niveles

- La **Capa de Cliente**: es la que ve el usuario (También se la denomina Capa de Usuario), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser “amigable” (entendible y fácil de usar) para el usuario.
- La **Capa Servidor de Aplicaciones**: (También denominada Capa de Negocio) es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- La **Capa Servidor de Datos**: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que reside la capa de negocio.

Lo importante es comprender que para la arquitectura existen varias perspectivas.



Tipos Comunes de Servidores:

- Servidor de Archivos (FTP, Novell).
- Servidores de Base de Datos (ORACLE,SQL,CBASE,INFORMIX).
- Servidor de Comunicaciones.
- Servidor de Impresión.
- Servidor de Terminal.
- Servidor de Aplicaciones (Windows NT, Novell).

Funciones Comunes del Servidor:

- Acceso, almacenamiento y organización de datos.
- Actualización de datos almacenados.
- Administración de Recursos Compartidos.
- Ejecución de toda la lógica para procesar un transacción.
- Procesamiento común de elementos del servidor (Datos, capacidad de CPU, almacenamiento en disco, capacidad de impresión, manejo de memoria y comunicación).

Red de Comunicación

Es todo aquel conjunto de elementos basados en hardware y software que permite establecer un enlace de comunicación entre:

Capa de Cliente – Capa Servidor de Aplicaciones – Capa de Servidor de Datos

Se clasifican por su tamaño LAN, MAN y WAN.

Características de Comunicación

- No se utiliza la metodología de compartición de archivos ya que todos los accesos a la información se llevan a cabo a través de peticiones por medio de comunicación.
- Debido a que los programas de manejo y control de información (Archivos y Bases de Datos solo se envían y reciben los resultados de las operaciones. (Tráfico Igual a Datos leídos o escritos).
- Debido a la flexibilidad de establecer sesiones con múltiples y manejo de información en varias bases de datos (en sitios remotos es requerido el uso de estilos transaccionales y cooperativos).

CAPITULO 2

2.1 Especificación de Requisitos de Software

2.1.1 Introducción

Este documento es una Especificación de Requisitos de Software para el Sistema de Control de No Conformidades. Todo su contenido ha sido elaborado con la colaboración de de los usuarios, directivos y desarrolladores de la institución.

2.1.2 Propósito

El Objetivo de la especificación es definir de manera clara y precisa las funcionalidades y condicionantes técnicas del sistema computacional que se desea desarrollar. Este documento va dirigido a los directivos y usuarios finales del sistema, será el medio de comunicación entre las partes involucradas en el desarrollo del Sistema de Control de No Conformidades.

Esta especificación está sujeta a revisiones por el grupo de usuarios a las diversas versiones que se generen.

2.1.3 Ámbito del Sistema

El objetivo es desarrollar un sistema que permita el registro y control de sucesos o evidencias que se encuentran en algunos procesos denominados No Conformidades, podríamos citar que la definición de una No Conformidad es el incumplimiento de un requisito.

El Sistema contempla lo siguiente:

- Registro y seguimiento de las acciones propias de las distintas etapas de los procesos así como sus respectivas acciones correctivas y preventivas.
- Determinar el nivel de satisfacción de dicha acción si fue efectiva o no.
- Reducir los tiempos de respuesta y acción al momento de encontrar una No-Conformidad.
- Correspondencia de No-Conformidades enviadas y recibidas.

- Estadísticas de No-Conformidades.

2.1.4 Definiciones.

Usuario Administrador	Persona que tiene acceso a todas las opciones del sistema
Usuarios del Sistema	Empleados de la Institución que interactúan con los procesos
Tramites	Tramites que se dan dentro de la Institución

2.1.5 Acrónimos.

ERS	Especificaciones de Requerimientos del Sistema
PCS	Computadores Personales
NCO	Expediente de No Conformidad
ICI	Industria Cartonera INCARPALM

2.1.6 Referencias.

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std 830, 1998.

2.1.7 Responsables del documento.

Levantamiento de la Información:

Participantes en el levantamiento de la información.

Gerente de Sistemas Integrados de Gestión: Ing. Gina Motta.

Aprobación de Usuario de los ERS: Ing. Jessenia Sanchez.

Aprobación Técnica de los ERS: Marcelo Serrano.

2.2 Descripción General

Este sistema brinda los utilitarios para el registro y clasificación de las No Conformidades encontradas en las diferentes áreas de la institución, así como la clasificación de la información para los reportes y estadísticas.

2.3 Requisitos Específicos

Esta es la sección más larga e importante del documento ya que aquí se especifican detalladamente los requisitos. Esto permite que los diseñadores realicen un sistema que satisfaga completamente los requisitos de los usuarios y que se planifique las pruebas de aceptación para la satisfacción de los directivos y usuarios.

2.3.1 Requisitos Funcionales

Los requisitos funcionales serán agrupados en casos de uso. Los casos de uso dan una perspectiva del sistema desde el punto de vista desde los actores para tener una visión general, identificar los participantes y el alcance del sistema.

2.3.2 Descripción de Funcionalidades

La prioridad de automatización puede ser: existente, manual, opcional, deseable, necesario u obligatorio.

Manual	La funcionalidad se registra manualmente y no se registra en el sistema
Opcional	Se especificara explícitamente por parte de la persona responsable si se automatizara las funcionalidades, bajo la supervisión del responsable de control de aceptación de los ERS
Deseable	El responsable del control de aceptación de los ERS definirá la automatización de estas funcionalidades
Necesario	Estas funcionalidades podrían ser implementadas de diferentes formas
Obligatorio	Estas Funcionalidades serán automatizadas 100%

2.3.3 Casos de Usos

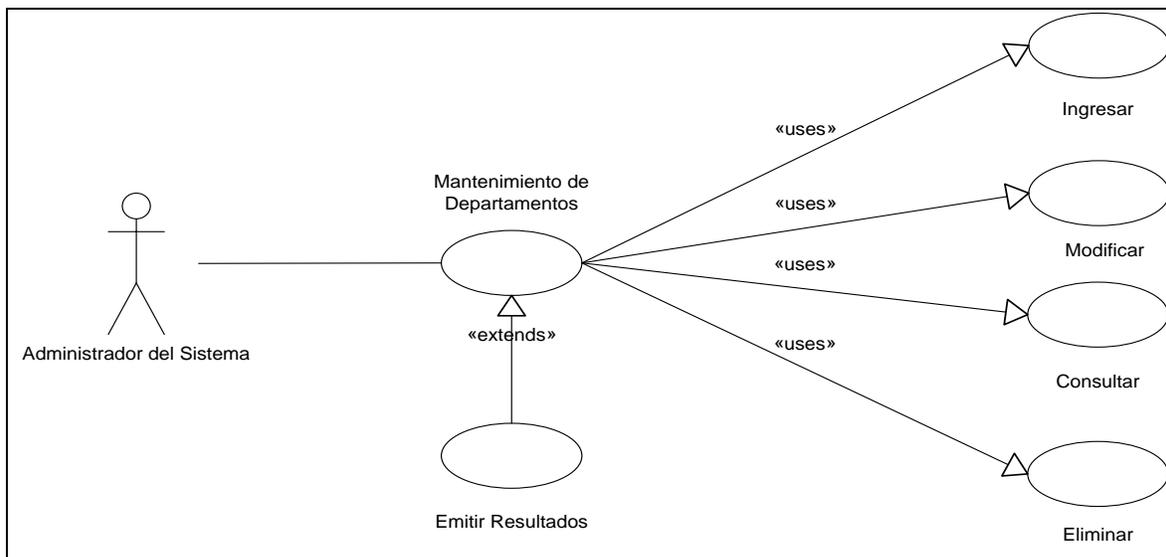


Figura 2.1 Caso de Uso 10 Mantenimiento de Departamentos

Caso de Uso 10	Mantenimiento de Departamentos
Actor	Administrador del Sistema
Descripción	Se definirán los Departamentos que forman parte de la institución
Prioridad	Obligatorio
Requisitos Asociados	
R.10.2	El sistema permitirá el ingreso de la información de los departamentos de la institución, como: Nombre, Jefe de Departamento, Fecha de Creación, Teléfono, Email, Estado
R.10.10	El sistema codificará a cada departamento mediante un número secuencial, que es único
R.10.20	El sistema modificará la información del departamento si el caso lo amerita todo esto a excepción del código asignado, que es único y no se repetirá
R.10.30	El sistema permitirá eliminar al departamento creado en el caso que no existan empleados creados asignados a un departamento.
R.10.40	El sistema presentará una interface grafica en forma de formulario para un fácil uso por parte del usuario.

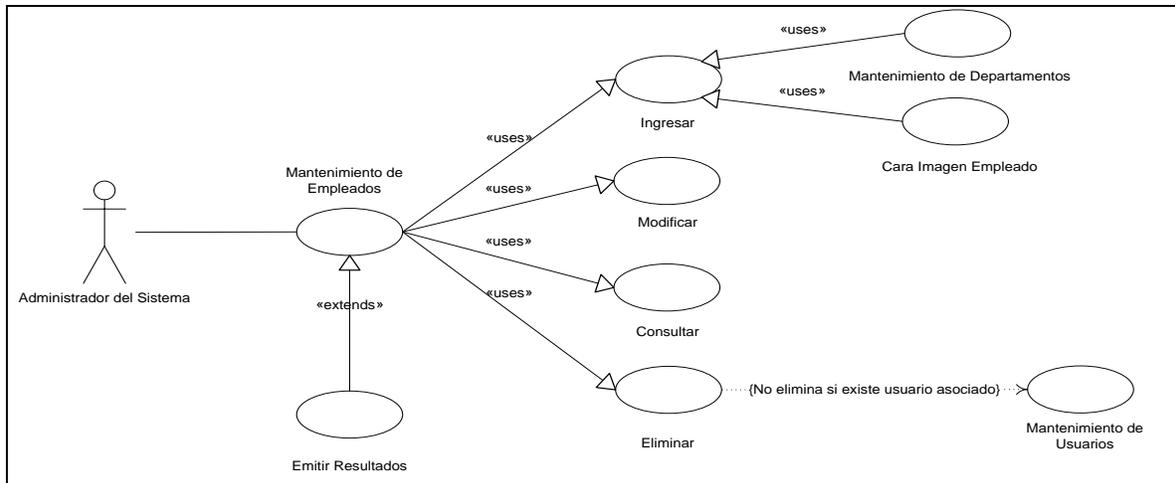


Figura 2.2 Caso de Uso Mantenimiento de Empleados

Caso de Uso 20	Mantenimiento de Empleados
Actor	Administrador del Sistema
Descripción	Se definen los Empleados del Sistema
Prioridad	Obligatorio
Requisitos Asociados	
R.20.2	El sistema permitirá el ingreso de la Información de los Empleados de la Institución, esto es los Nombres, Apellidos, Sexo, Fecha de Nacimiento.
R.20.5	El sistema permitirá el ingreso de la Imagen del Empleado para su historial
R.20.10	El sistema codificará a cada empleado mediante un número secuencial.
R.20.20	El sistema modificará a cada empleado si el caso así lo amerita todo esto a excepción del código asignado que es único y no se repetirá.
R.20.30	El sistema permitirá eliminar el empleado creado en el caso que no existan usuarios creados que hayan tomado un empleado para su creación.
R.20.40	El sistema asociará un empleados a un departamento
R.20.50	El sistema presentará una interface gráfica e forma de formulario para un fácil uso por parte del usuario.

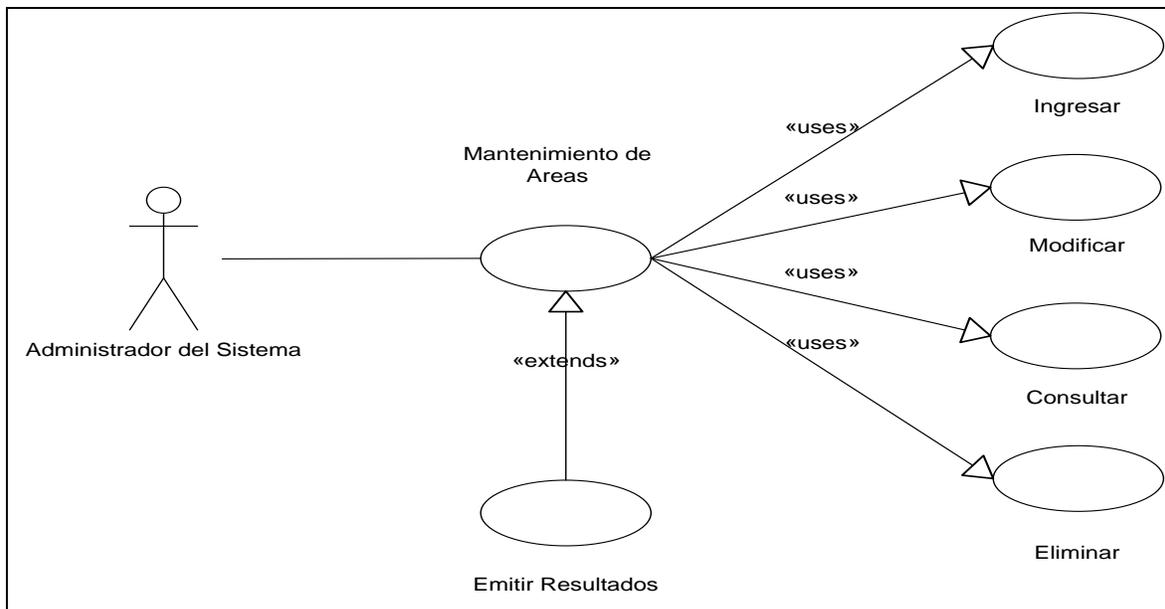


Figura 2.3 Caso de Uso Mantenimiento de Areas

Caso de Uso 30	Mantenimiento de Areas
Actor	Administrador del Sistema
Descripción	Se definen las áreas de la institución
Prioridad	Obligatorio
Requisitos Asociados	
R.30.2	El sistema permitirá el ingreso de la información de las áreas de la institución. Esta se almacena en los campos: Nombre de Area, Descripción de Funciones, Estado, Fecha de Ingreso, Usuario de Ingreso
R.30.10	El sistema codificará cada área con un número secuencial
R.30.20	El sistema modificará el área creada si fuera necesario, todo esto a excepción del código asignado que es único y no se repetirá.
R.30.30	El sistema eliminará el área en caso si no existiera una No Conformidad asociada.
R.30.40	El sistema presentará una interface grafica en forma de formulario para un fácil uso por parte del usuario

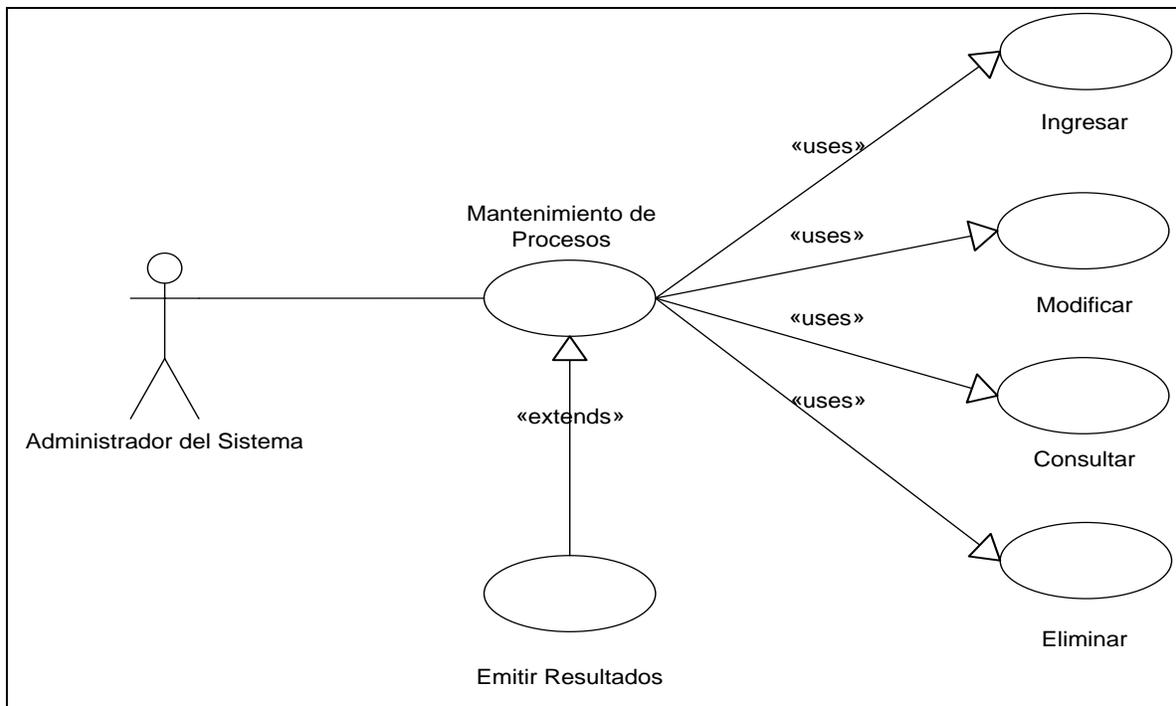


Figura 2.4 Caso de Uso Mantenimiento de Procesos

Caso de Uso 40	Mantenimiento de Procesos
Actor	Administrador del Sistema
Descripción	Se definen las procesos de la institución
Prioridad	Obligatorio
Requisitos Asociados	
R.40.2	El sistema permitirá el ingreso de la información de las procesos de la institución, como: Nombre de Proceso, Descripción, Estado
R.40.10	El sistema codificará cada proceso con un numero secuencial
R.40.20	El sistema modificará el proceso creado si fuera necesario, todo esto a excepción del código asignado que es único y no se repetirá.
R.40.30	El sistema eliminará el proceso en caso si no existiera una No Conformidad asociada.
R.40.40	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario.

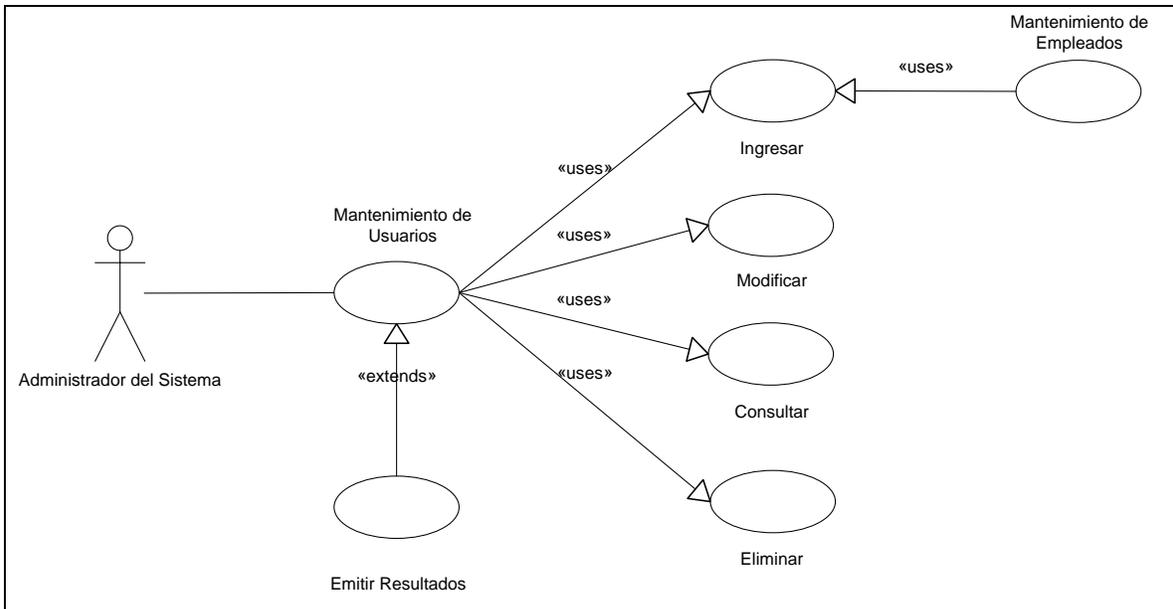


Figura 2.5 Caso de Uso Mantenimiento de Usuarios

Caso de Uso 50	Mantenimiento de Usuarios
Actor	Administrador del Sistema
Descripción	Se definen las usuarios de la institución
Prioridad	Obligatorio
Requisitos Asociados	
R.50.2	El sistema permitirá el ingreso de la información de las usuarios del sistema como Nombre de Usuario, Clave de Acceso
R.50.10	El Administrador del Sistema codificará cada usuario con una letra y el primer apellido del empleado
R.50.20	El sistema asociará un empleado a un usuario, es decir los usuarios deben ser empleados de la Institución
R.50.30	El sistema modificará el usuario creado si fuera necesario, todo esto a excepción del código asignado que es único y no se repetirá
R.50.40	El sistema eliminará el usuario creado en caso si no existiera un empleado asociado
R.50.50	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario

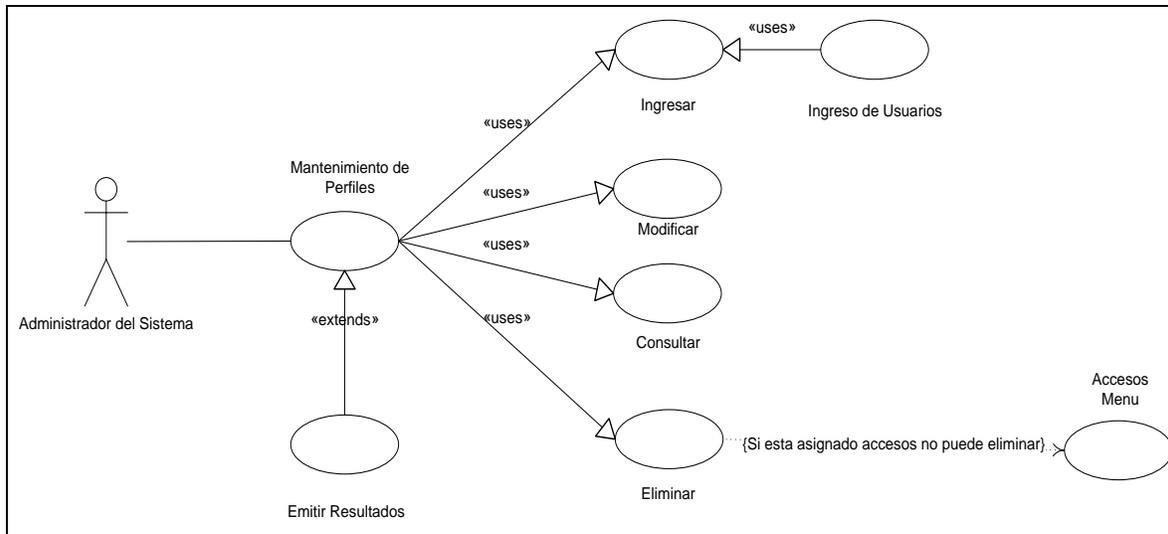


Figura 2.6 Caso de Uso Mantenimiento de Perfiles

Caso de Uso 60	Mantenimiento de Perfiles
Actor	Administrador del Sistema
Descripción	Se definen las perfiles para cada usuario del sistema
Prioridad	Obligatoria
Requisitos Asociados	
R.60.2	El sistema permitirá el ingreso de la información de las perfiles del sistema, esta se almacenará en los campos: Código de Perfil, Nombre de Perfil, Descripción de Perfil
R.60.10	El sistema codificará cada perfil de manera manual para otorgarle libertad de creación
R.60.20	El sistema asociará un perfil a un usuario, es decir los usuarios deben ser usuarios del sistema
R.60.30	El sistema modificará la información del perfil creado si fuera necesario, todo esto a excepción del código asignado que es único y no se repetirá
R.60.40	El sistema eliminará el perfil en caso si no estuviese asociado a un usuario
R.60.50	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario

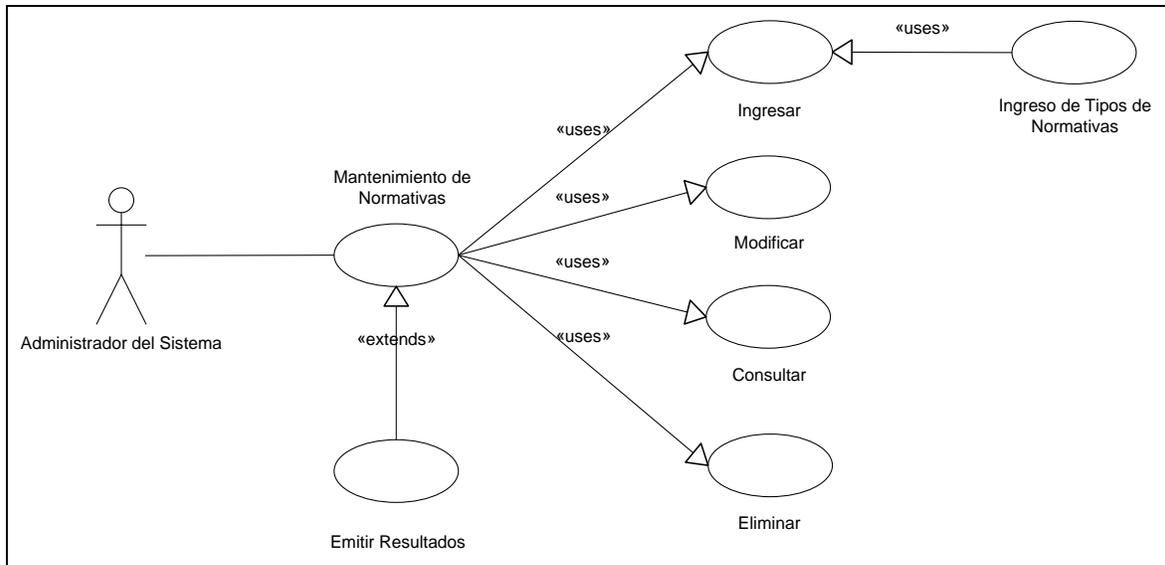


Figura 2.7 Caso de Uso Mantenimiento de Normativas

Caso de Uso 70	Mantenimiento de Normativas
Actor	Administrador del Sistema
Descripción	Se definen las normativas que formaran parte del sistema
Prioridad	Obligatorio
Requisitos Asociados	
R.70.2	El sistema permitirá el ingreso en forma jerárquica de la información de las normativas, como: Código de Normativa Nombre de Normativa Código Padre
R.70.10	El sistema asociará una normativa a un tipo de Normativa
R.70.20	El sistema modificará una normativa en caso de ser necesario, todo esto a excepción del código ya que es único y no se repetirá
R.70.30	El sistema eliminará las normativas solo en caso de ser requerido
R.70.40	El sistema presentará una interface grafica en forma de formulario para un fácil uso por parte del usuario

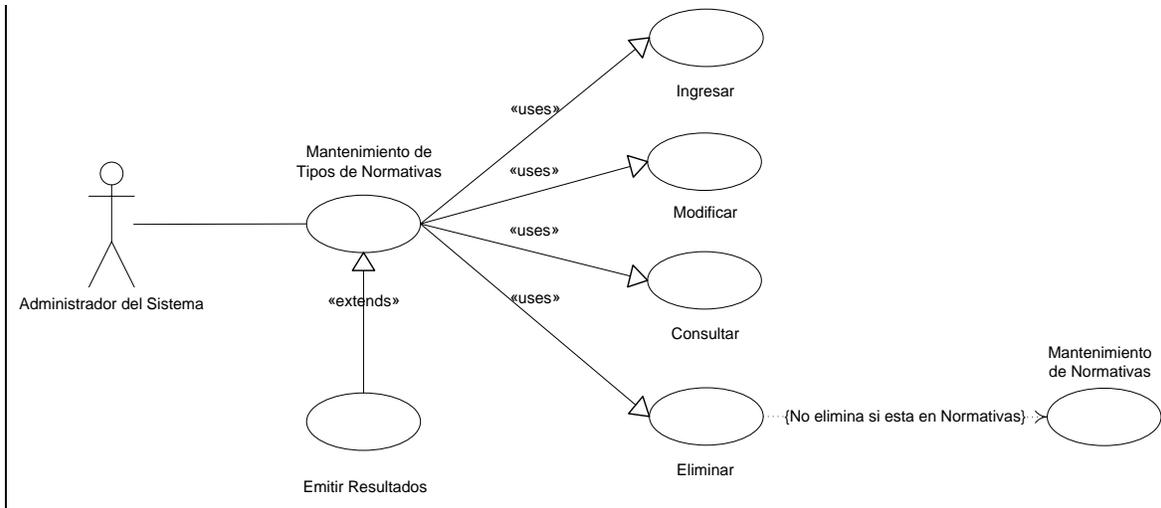


Figura 2.8 Caso de Uso Mantenimiento de Tipos de Normativas

Caso de Uso 80	Mantenimiento de Tipos de Normativas
Actor	Administrador del Sistema
Descripción	Se definen los Tipos de Normativa que formaran parte del Sistema
Prioridad	Obligatorio
Requisitos Asociados	
R.80.2	El sistema permitirá el ingreso de la información de los tipos de normativas. Esta se registrará en los campos Código Tipo Normativa, Nombre de Tipo de Normativa
R.80.30	El sistema eliminará un tipo de normativa si esta no ha sido asociada a una normativa vigente en el sistema.
R.80.40	El sistema presentará una interface grafica en forma de formulario para un fácil uso por parte del usuario.

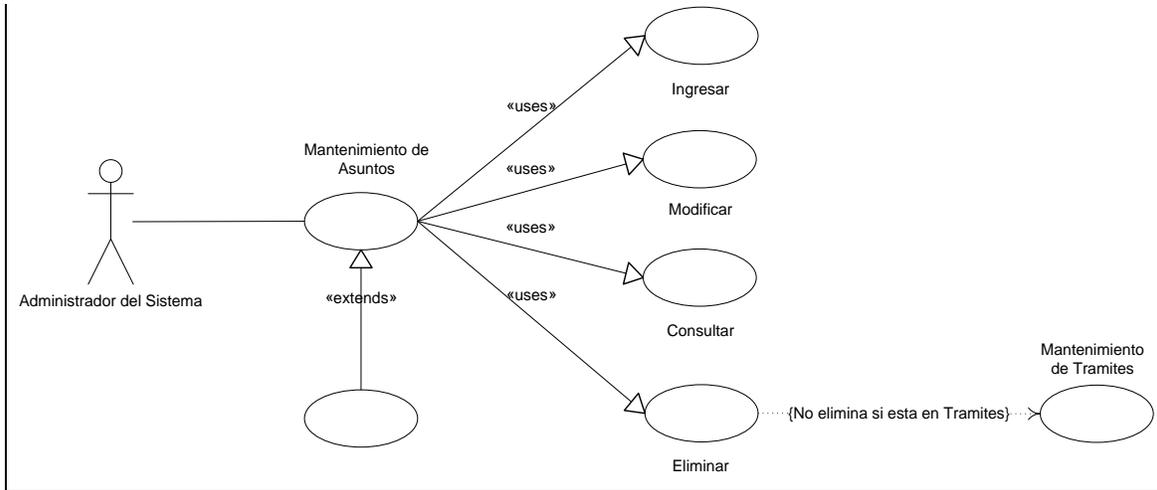


Figura 2.9 Caso de Uso Mantenimiento de Asuntos

Caso de Uso 90	Mantenimiento de Asuntos
Actor	Administrador del Sistema
Descripción	Se definen los asuntos que forman parte del sistema
Prioridad	Obligatorio
Requisitos Asociados	
R.90.2	El sistema permitirá el ingreso de la información de las de los asuntos. Esta se ingresará en los campos Código de Asunto, Nombre de Asunto, Fecha de Ingreso
R.90.30	El sistema eliminará el asunto si este no ha sido asociado a un expediente de no conformidad
R.90.40	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario.

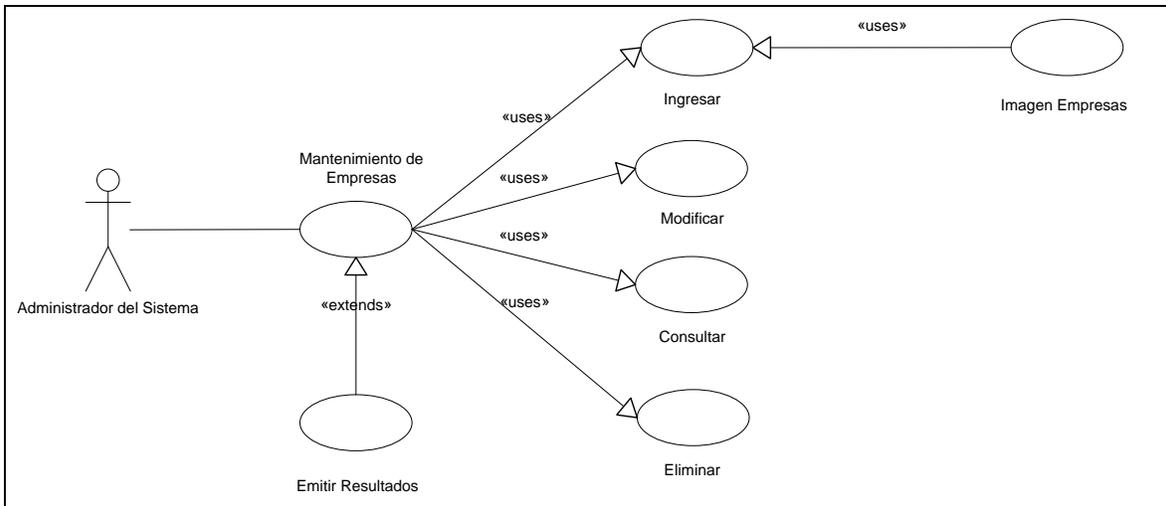


Figura 2.10 Caso de Uso Mantenimiento de Empresas

Caso de Uso 100	Mantenimiento de Empresas
Actor	Administrador del Sistema
Descripción	Se definen que empresa va a utilizar el producto
Prioridad	Obligatorio
Requisitos Asociados	
R.100.2	El sistema permitirá el ingreso de la información de la empresa. Esta se ingresará en los campos: Nombre de Empresa, Dirección Teléfono, Fax, Website, Representante Legal. Logo Empresa
R.100.30	El sistema modificará la información de la empresa si este fuere necesario, a excepción del código de empresa ya que este es único
R.100.40	El sistema no eliminará la información de la empresa
R.100.50	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario.

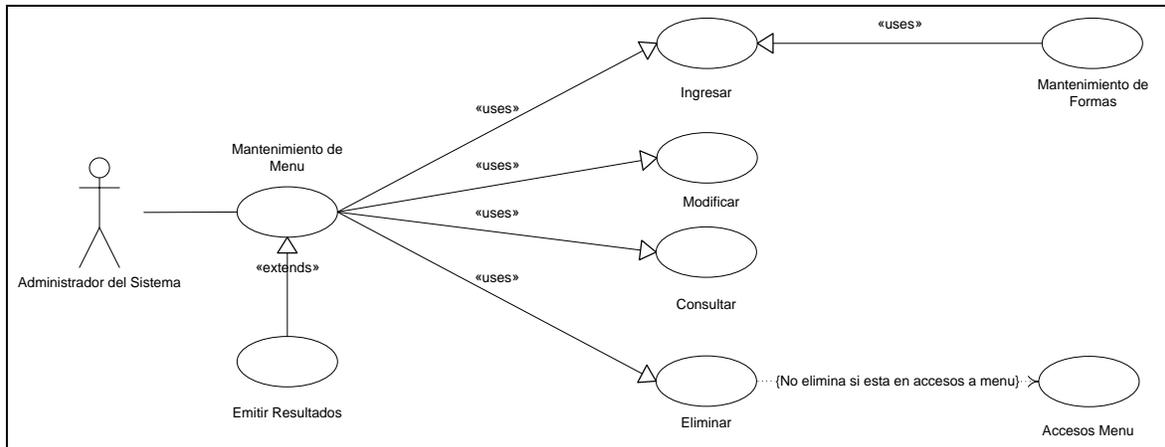


Figura 2.11 Caso de Uso Mantenimiento de Menú

Caso de Uso 110	Mantenimiento de Menú
Actor	Administrador del Sistema
Descripción	Se definen las opciones de menú del sistema
Prioridad	Obligatorio
Requisitos Asociados	
R.110.2	El sistema permitirá el ingreso de la información de las opciones el sistema. Esta se ingresará en los campos Código, Nombre, Path, Código Padre, Orden
R.110.10	El sistema permitirá ingresar de manera jerárquica las opciones del sistema, asignando un orden específico
R.110.20	El sistema permitirá asociar por cada opción una forma en el menú
R.110.30	El sistema permitirá ingresar información relacionada con las formas de cada opción, la descripción de cada forma asociada y la ruta de acceso
R.110.50	El sistema modificará la información de la opción si fuera necesario, solo podrá modificar la el nombre y la descripción de la forma
R.110.50	El sistema eliminará la opción si fuera necesario, si esta no esta asociada al acceso de un usuario
R.110.60	El sistema asociará una forma a por cada opción
R.110.70	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario.

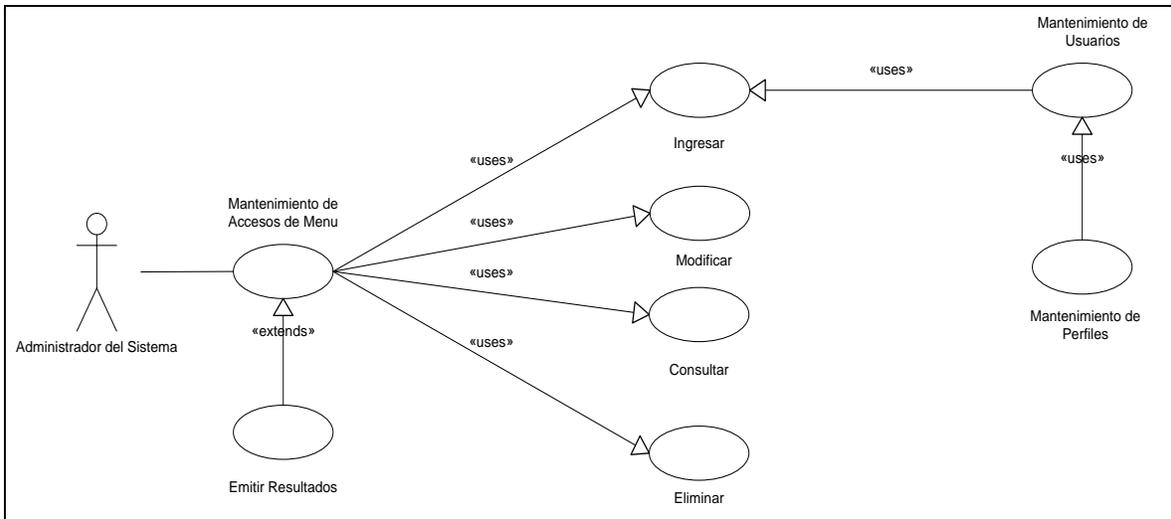


Figura 2.12 Caso de Uso Mantenimiento de Accesos al Menú

Caso de Uso 120	Mantenimiento de Accesos Menú
Actor	Administrador del Sistema
Descripción	Se definen los accesos al menú para cada usuario del sistema
Prioridad	Obligatorio
Requisitos Asociados	
R.120.2	El sistema permitirá el ingreso de la información de los accesos a las opciones de cada usuario al sistema
R.120.10	El sistema asociará una opción a cada usuario , no todos los usuario tiene acceso a todas las opciones
R.120.20	El sistema permitirá definir que el acceso al sistema dependiendo del perfil asignado a cada usuario
R.120.30	El sistema eliminara el acceso a la opción en caso de ser requerido
R.120.40	El sistema presentara una interface grafica en forma de formulario para un fácil uso por parte del usuario

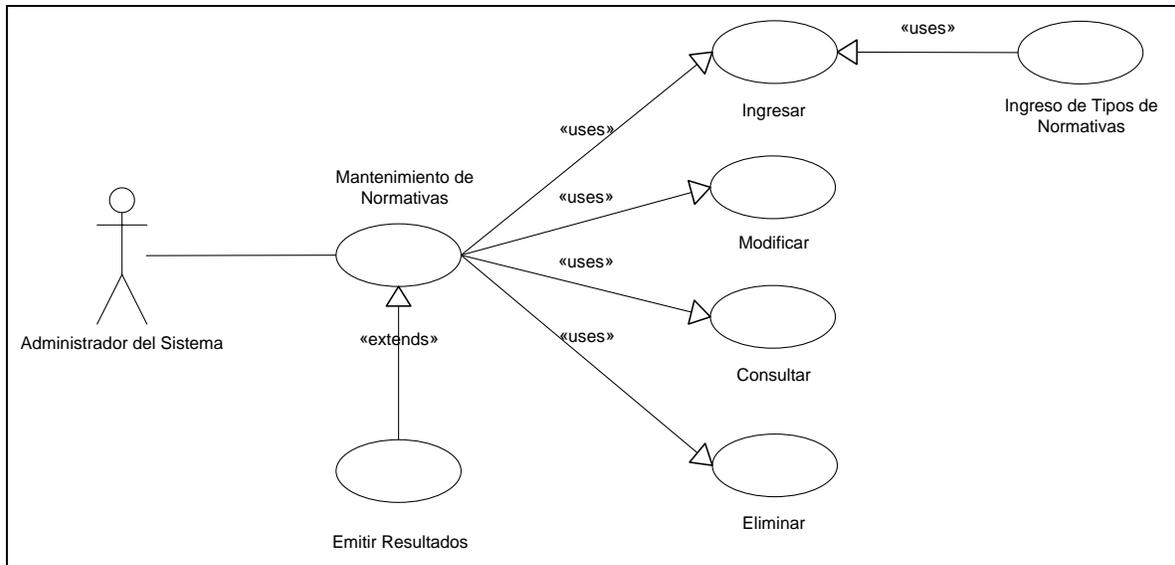


Figura 2.13 Caso de Uso Mantenimiento de Normativas

Caso de Uso 130	Mantenimiento de Normativas
Actor	Administrador del Sistema
Descripción	Se definen las Normativas del que se llevan como modelo dentro de la institución
Prioridad	Obligatorio
Requisitos Asociados	
R.130.2	El sistema permitirá el ingreso en forma jerárquica de la información de las normativas del sistema
R.130.10	El sistema permitirá modificar la información de las normativas en caso de ser requerido, todo esto a excepción del código ya que es único y no se repetirá
R.130.20	El sistema presentara una interface grafica en forma de formulario para un fácil uso por parte del usuario

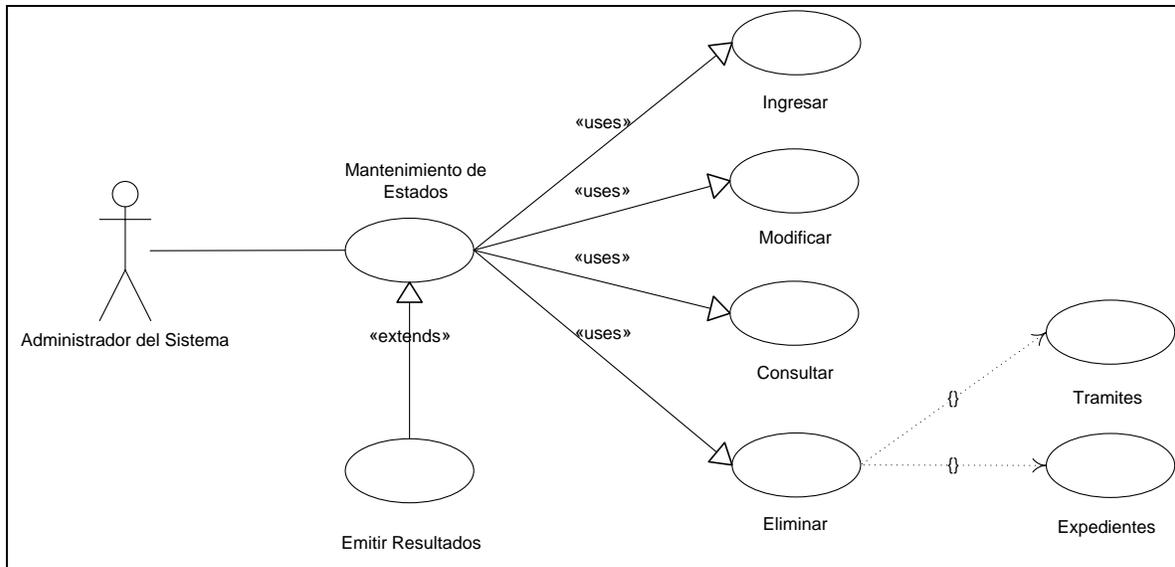


Figura 2.14 Caso de Uso Mantenimiento de Estados

Caso de Uso 140	Mantenimiento de Estados
Actor	Administrador del Sistema
Descripción	Se definen los estados de las formas y los estados de los Expedientes de No Conformidad. Esta información se almacenará en los campos: Código, Nombre, Descripción, Código Padre
Prioridad	Obligatorio
Requisitos Asociados	
R.140.2	El sistema permitirá el ingreso de la información de las estados
R.140.10	El sistema eliminará el estado si este no ha sido asociado a un expediente de no conformidad
R.140.20	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario

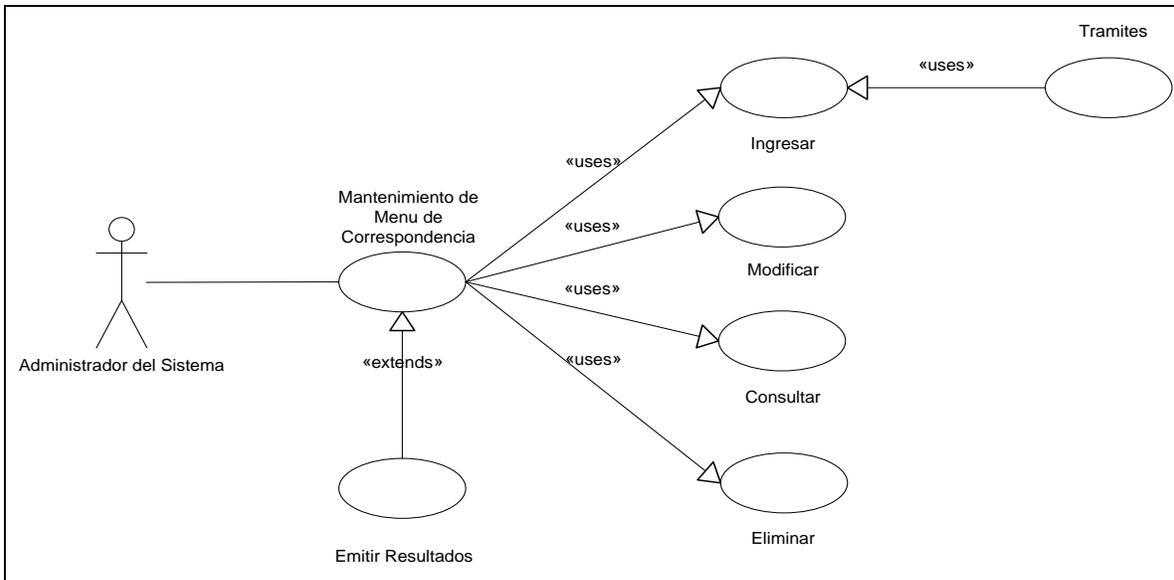


Figura 2.16 Caso de Uso Mantenimiento de Menú de Correspondencia

Caso de Uso 160	Mantenimiento de Menú de Correspondencia
Actor	Administrador del Sistema
Descripción	Se definen las opciones del menú de correspondencia del sistema
Prioridad	Obligatorio.
Requisitos Asociados	
R.150.2	El sistema permitirá el ingreso en forma jerarquice de cada una de las opciones de correspondencia Bandeja de Recibidos, Bandeja de Enviados, relacionadas con un Trámite del sistema
R.150.10	El sistema asociará una opción a una opción padre en el menú de correspondencia.
R.150.30	El sistema permitirá asociar un opción (nodo hijo del árbol)del menú de correspondencia un objeto de la base de datos.
R.150.40	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario.

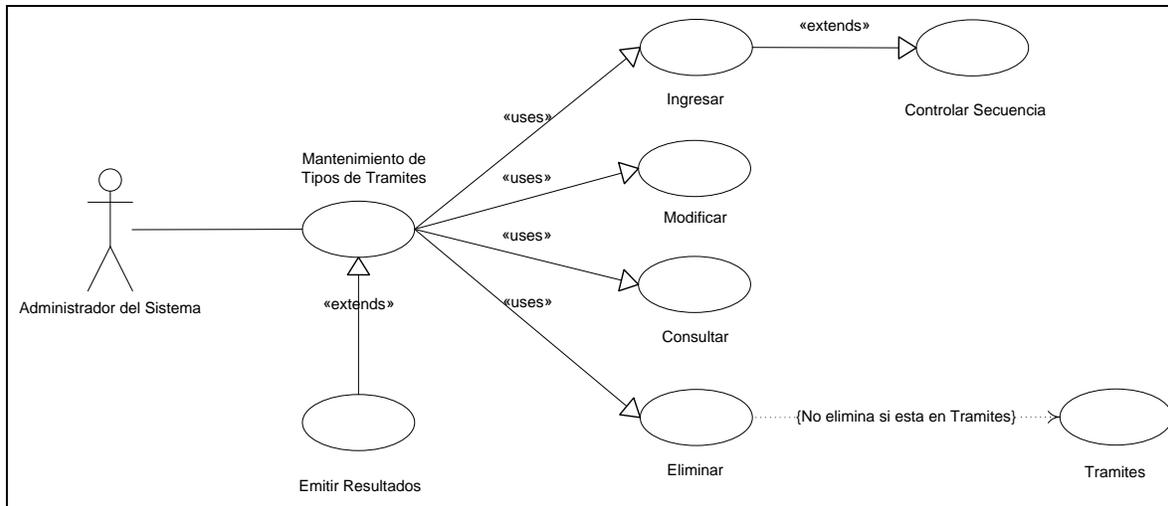


Figura 2.17 Caso de Uso Mantenimiento Tipos de Trámites

Caso de Uso 170	Mantenimiento de Tipos Trámites
Actor	Administrador del Sistema
Descripción	Se definen los tipos de Trámites de la Institución. Esta se almacenara en los campos: Código Tipo Trámite, Nombre Descripción
Prioridad	Obligatorio
Requisitos Asociados	
R.170.2	El sistema permitirá el ingreso de los Tipos de Trámites de la institución. (No Conformidad) descripción del tipo de tramite
R.170.10	El sistema actualizará automáticamente la secuencia por cada tipo de Trámite ingresado.
R.170.30	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario.

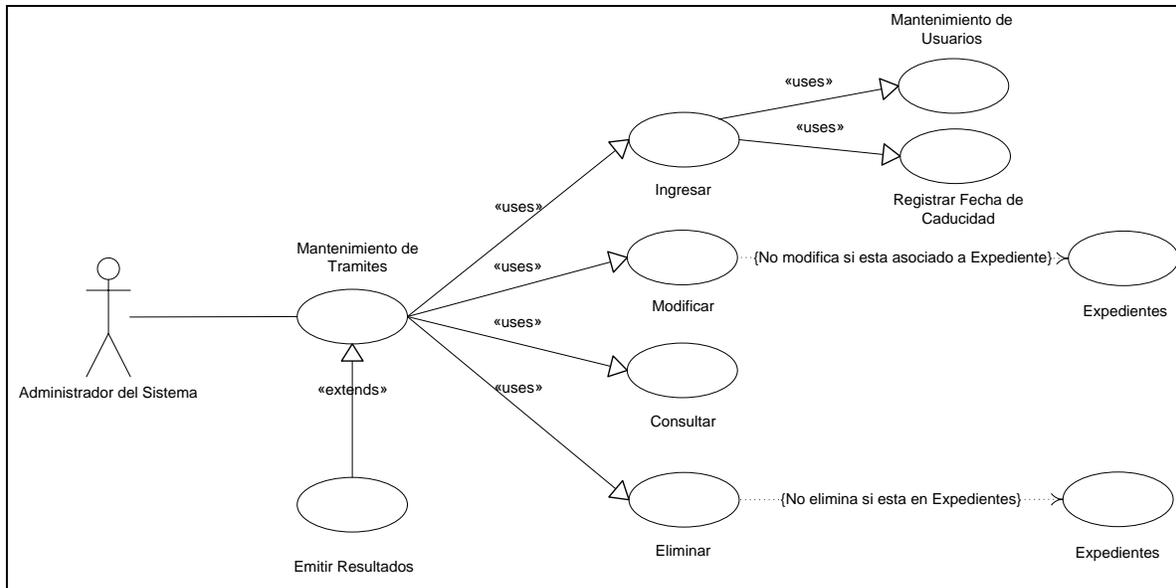


Figura 2.18 Caso de Uso Mantenimiento de Trámites

Caso de Uso 180	Mantenimiento de Trámites
Actor	Administrador del Sistema
Descripción	Se definen los Trámites de la institución. Esta información se ingresará en los campos: Código de Trámite, Nombre, Descripción, Manual de Proceso, Tiempo Máximo, Proceso
Prioridad	Obligatorio
Requisitos Asociados	
R.180.2	El sistema permitirá el ingreso de los Trámites de la institución.
R.180.10	El sistema permitirá ingresar la información base para un trámite numero, descripción, manual de proceso
R.180.20	El sistema ingresará el tiempo máximo en días para calcular la caducidad del tramite a partir del ingreso de los días definidos para su ejecución
R.180.30	El sistema asociará un trámite a un proceso de la institución
R.180.40	El sistema asociará un trámite con una normativa del sistema previamente creada
R.180.50	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario

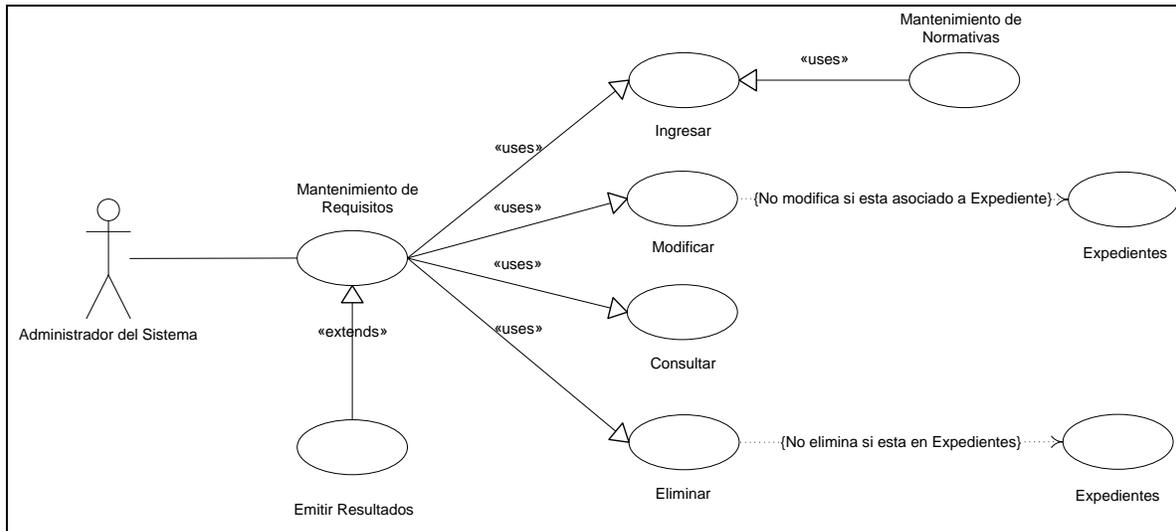


Figura 2.19 Caso de Uso Mantenimiento de Requisitos

Caso de Uso 190	Mantenimiento de Requisitos
Actor	Administrador del Sistema
Descripción	Se definen los Requisitos que formaran de un Trámite o Proceso que se deben controlar cuando se haya evidenciado una no conformidad. Esta información se almacenará en los campos: Req_Codigo, Req_Nombre, Req_Codigo_Alterno
Prioridad	Obligatorio
Requisitos Asociados	
R.190.2	El sistema permitirá el ingreso de los requisitos que serán parte del sistema
R.190.10	El sistema permitirá codificar los requisitos como están establecidos en la norma
R.190.20	El sistema permitirá ingresar información de los requisitos que se deben controlar cuando se evidencia una no conformidad
R.190.30	El sistema permitirá ingresar la información de manera jerárquica
R.190.40	El sistema asociará un requisito con una normativa
R.190.50	El sistema permitirá modificar la información del requisito, todo esto excepto el código ya que es único y no se repetirá
R.190.60	El sistema presentara una interface gráfica en forma de formulario para un fácil uso por parte del usuario

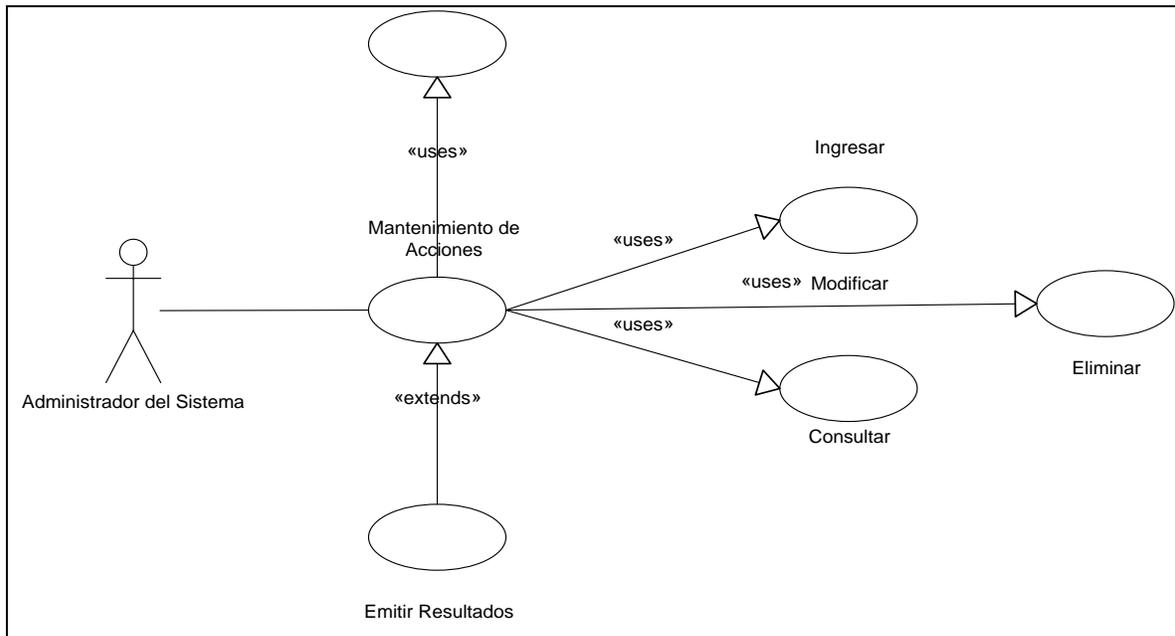


Figura 2.20 Caso de Uso Mantenimiento de Acciones

Caso de Uso 200	Mantenimiento de Acciones
Actor	Administrador del Sistema
Descripción	Se definen los tipos de acciones que formaran parte del sistema, cada vez que se registre una No Conformidad. Esta información de almacenará en los campos: Código, Nombre, Descripción
Prioridad	Obligatorio
Requisitos Asociados	
R.190.2	El sistema permitirá el ingreso de los tipos de acciones que se deben dar a una No Conformidad
R.190.10	El sistema permitirá codificar los acciones como está establecido en la norma
R.190.60	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario

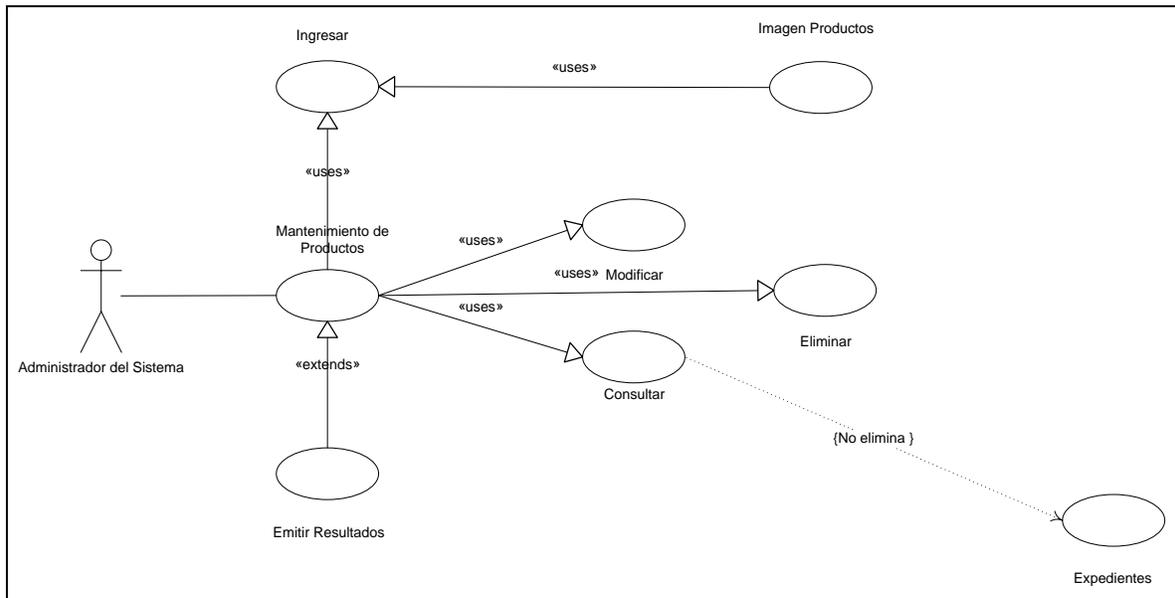


Figura 2.21 Caso de Uso Mantenimiento de Productos

Caso de Uso 210	Mantenimiento de Productos
Actor	Administrador del Sistema
Descripción	Se definen los productos que formaran parte del sistema
Prioridad	Necesario
Requisitos Asociados	
R.210.2	El sistema permitirá el ingreso de los tipos de productos
R.210.10	El sistema permitirá codificar los productos de acuerdo a la definición de los productos de la empresa
R.210.20	El sistema permitirá ingresar la imagen del producto si fuera necesario para ayudar al usuario a su identificación.
R.210.30	El sistema presentará una interface grafica en forma de formulario para un fácil uso por parte del usuario

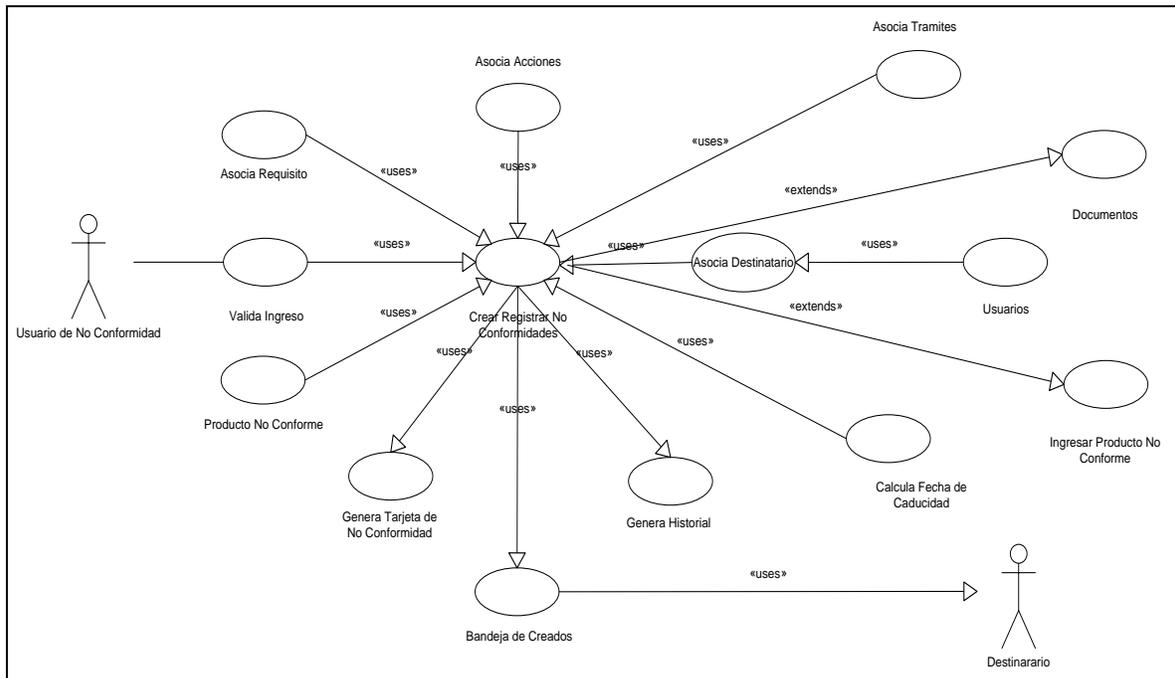


Figura 2.20 Caso de Uso Registrar No Conformidad

Caso de Uso 220	Creación o Registro de Expedientes de No Conformidad
Actor	Usuario (Auditor de Calidad)
Descripción	<p>En este caso el auditor de calidad cuando encuentra una no conformidad, procede a registrarla, para luego proceder a enviarla al destinatario ingresado en el registro.</p> <p>Esta información se almacenara en los campos: Exp_Numero, Nombre, Remitente, Destinatario, Asunto, Area, Proceso, Estado Análisis de Causa, Observación, Requisitos, Acciones.</p> <p>Puede también ingresar la información de o los productos que durante el muestreo se encuentre una anomalía, y a su vez adjuntar un documento para su revisión.</p>
Prioridad	Obligatorio
Requisitos Asociados	

R.220.2	El sistema permitirá el ingreso de los expedientes de no conformidad, encontrados por el auditor
R.220.5	El sistema codificará de manera automática a cada expediente dependiendo de del tipo de trámite.
R.220.10	El sistema asociará un expediente con una área donde se encontró la No Conformidad
R.220.20	El sistema asociará un expediente con un proceso en que se encontró la No Conformidad
R.220.30	El sistema asociará un expediente a un trámite identificándolo por el tipo de trámite
R.220.40	El sistema asociará la fecha de caducidad el expediente en función de los días asignados a cada Trámite.
R.220.50	El sistema permitirá ingresar uno varios Requisitos que se incumplan cada vez que se registre una No Conformidad
R.220.60	El sistema permitirá ingresar uno o varias Acciones Correctivas o Preventivas dependiendo del criterio del Auditor
R.220.70	El sistema permitirá asociar uno o varios productos en caso de encontrarse evidencia una No Conformidad
R.220.80	El sistema permitirá anexar un documento en formato pdf en caso de ser necesario
R.220.90	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario

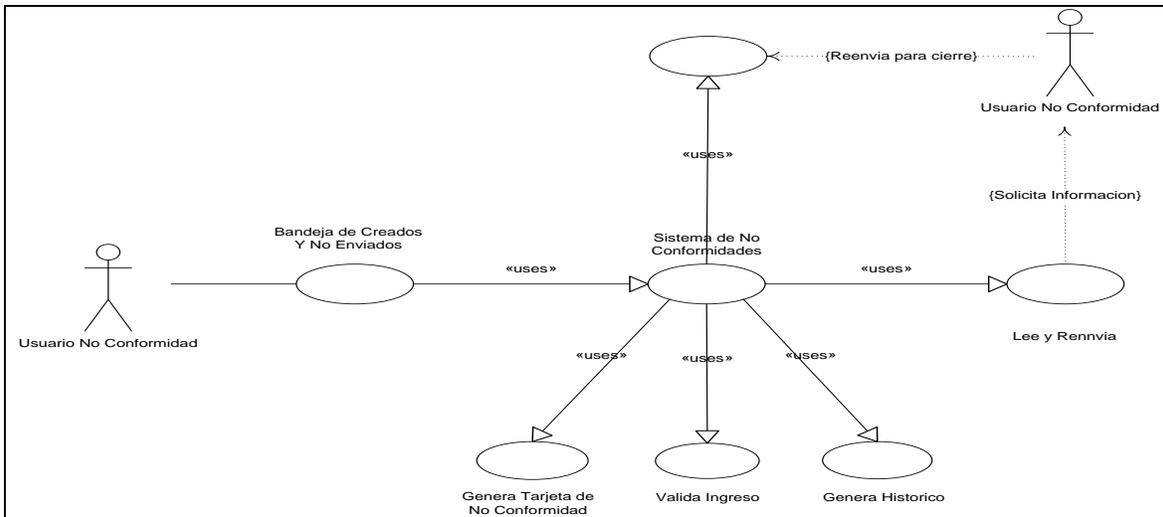


Figura 2.30 Caso de Leer y No Reenviar Conformidad

Caso de Uso 230	Leer Reenviar Expedientes de No Conformidad
Actor	Usuario
Descripción	En este caso el usuario del sistema es capaz de poder revisar el expediente de no conformidad enviado por el auditor, podrá revisar cuales son los requisitos que se incumplen y cuáles son las acciones correctivas o de mejora que fueron asignadas, a su vez este puede reenviarlo al usuario remitente una vez que haya corregido o no, la no conformidad
Prioridad	Obligatorio
Requisitos Asociados	
R.230.2	El sistema permitirá la revisión de los expedientes de no conformidad
R.230.5	El sistema permitirá revisar que requisitos se incumplen y que acciones han sido asignadas
R.230.10	El sistema permitirá reenviar el expediente una vez corregida o no la no conformidad
R.230.20	El sistema asociará un nuevo destinatario para su reenvío
R.230.30	El sistema permitirá revisar los documentos anexados al expediente de no conformidad
R.230.40	El sistema presentará una interface gráfica en forma de formulario para un fácil uso por parte del usuario

CAPITULO 3 DISEÑO DEL SISTEMA

3.1 Modelo de los Datos

3.1.1 Modelo Entidad Relación

Definición del Modelo de Entidad Relación (E-R) es un modelo de datos de alto nivel. Está basado en la percepción de un mundo real que consiste en una colección de objetos básicos denominados entidades y de relaciones entre estos objetos.

Para la creación de las definiciones de la Base de Datos E-R utilizaremos como herramienta para modelar Erwin Data Modeler.

Para Crear un Modelo dar click en **Nuevo**  y aparecerá esta opción seleccionamos **Create a new Model.**

El Diagrama Entidad Relación puede ser visto en el **Anexo 1**

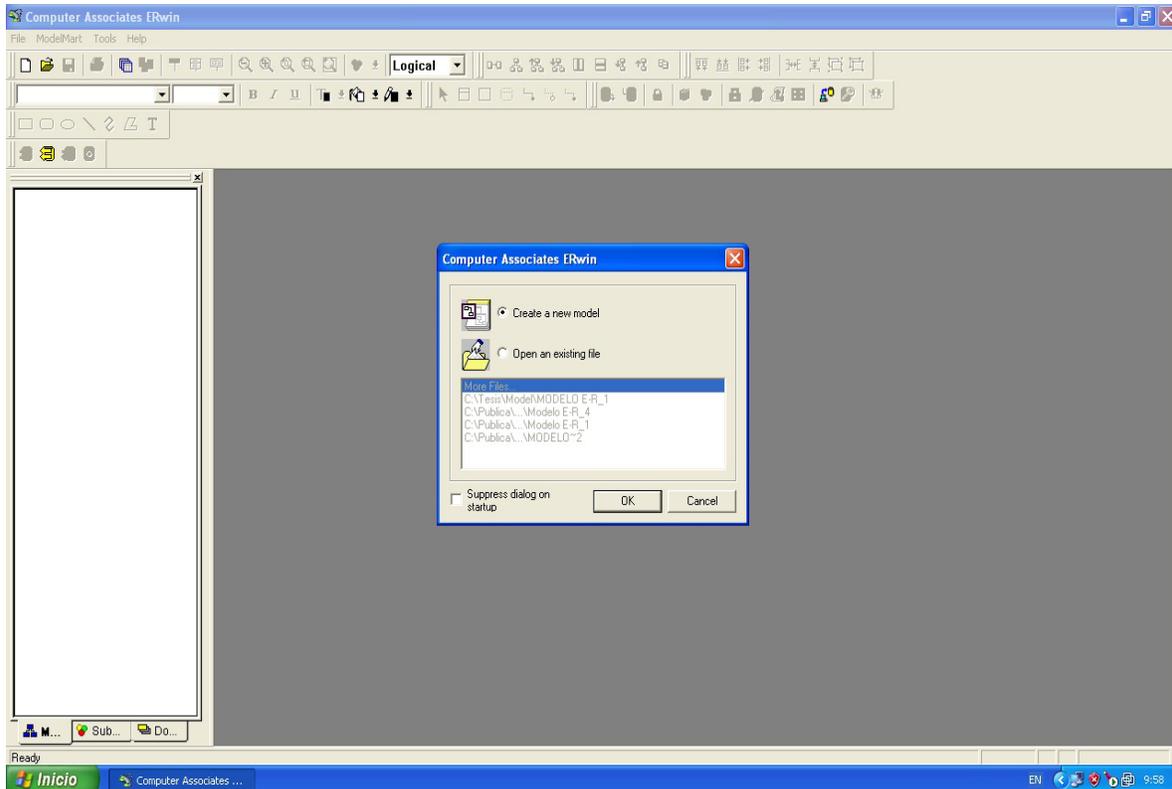
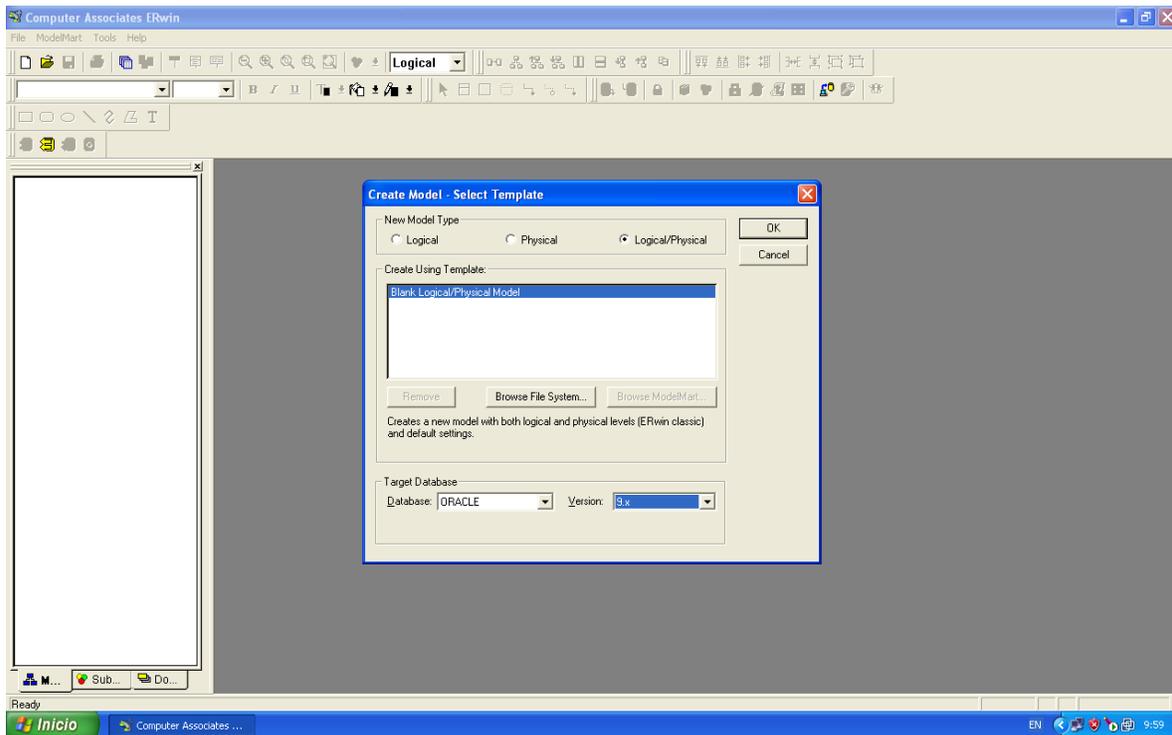
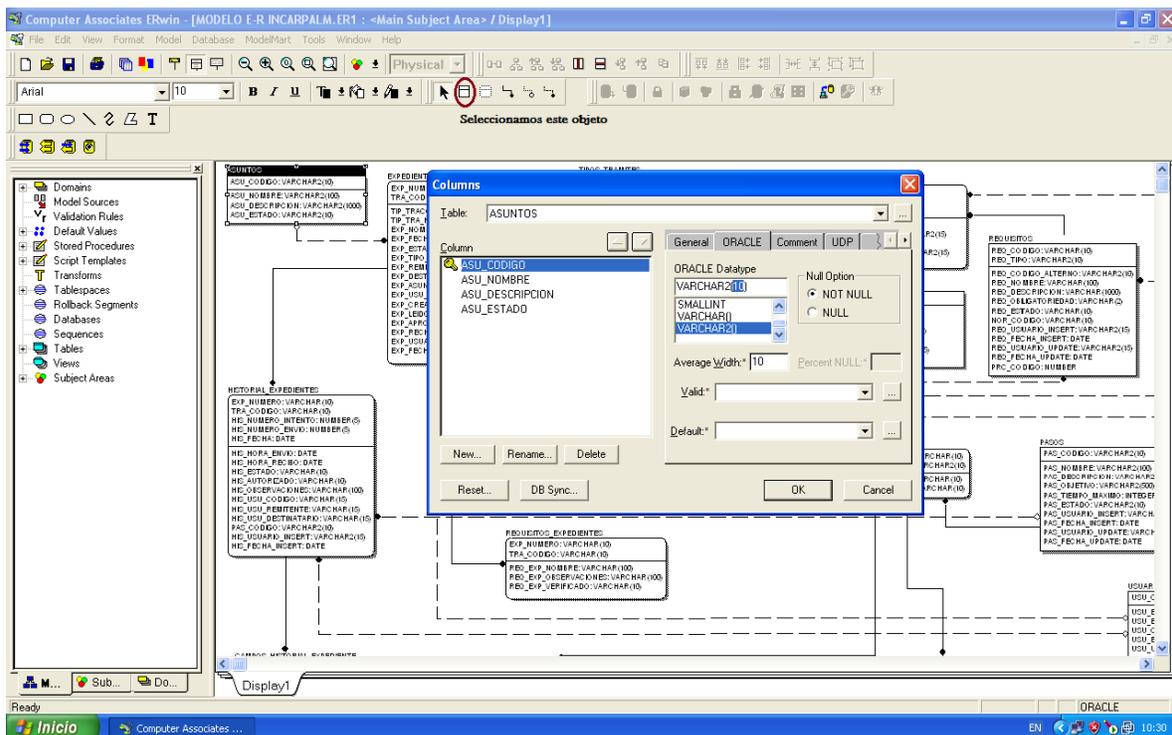


Figura 3.1 Crear Modelo Entidad Relación

Ahora tendrá la siguiente pantalla en donde debe seleccionarse **Logical/Physical**



Seleccionamos el Objeto marcado y lo llevamos al lienzo, para proceder a crear la estructura de la tabla, así como la definición de sus atributos



3.1.2 Diccionario de Datos

Tabla ACCESOS_ARBOL_ADMINISTRADOR			
Propósito: almacena las opciones de menú a las que los usuarios tienen acceso			
Name	Type	Nullable	Comments
Arb_Adm_Id	Number		Código opción de árbol
Usu_Codigo	Varchar2(15)		Código de usuario que va a tener permisos
Acc_Arb_Usuario_Insert	Varchar2(15)	Y	Usuario de insert
Acc_Arb_Fecha_Insert	Date	Y	Fecha de insert
Acc_Arb_Usuario_Update	Varchar2(15)	Y	Usuario de update
Acc_Arb_Fecha_Update	Date	Y	Fecha de update

Tabla ACCIONES			
Propósito: almacena las acciones que se dan a las no conformidades.			
Name	Type	Nullable	Comments
Acc_Codigo	Varchar2(10)		Código de acción
Acc_Nombre	Varchar2(100)	Y	Nombre de acción
Acc_Descripcion	Varchar2(500)	Y	Descripción de acción
Acc_Codigo_alterno	Varchar2(10)	Y	Código alterno o código padre

Tabla ACCIONES_EXPEDIENTES			
Propósito: almacena las acciones de los expedientes ingresados al sistema.			
Name	Type	Nullable	Comments
Acc_Codigo	Varchar2(10)		Código de acción
Exp_Numero	Varchar2(10)		Numero de expediente de no conformidad
Tra_Codigo	Varchar2(10)		Código de Tramite
Acc_Exp_Seleccion	Varchar2(10)	Y	Acción seleccionada
Acc_Descripcion	Varchar2(4000)	Y	Descripción de acción seleccionada
Acc_Numero	Number		Numero de acción

Tabla ARBOL_CORRESPONDENCIA			
Propósito: almacenar las opciones del menú de correspondencia en forma jerárquica(árbol)			
Name	Type	Nullable	Comments
Arb_Cor_Id	Varchar2(10)		Código de opción de árbol de correspondencia
Tra_Codigo	Varchar2(10)	Y	Código de tramite
Arb_Cor_Tipo	Varchar2(10)		Código de tipo de opción
Arb_Cor_Nombre	Varchar2(100)		Nombre de opción o nombre del nodo
Arb_Cor_Nivel	Varchar2(1)		Nivel de nodo
Arb_Cor_Id_Subnivel	Varchar2(10)	Y	Código alternativo o código padre
Arb_Cor_Estado	Varchar2(10)	Y	Estado del nodo
Arb_Id_Orden	Number	Y	Orden de la opción
Arb_Cor_Objeto	Varchar2(100)	Y	Nombre del objeto que va a hacer referencia vw_expedientes_recibidos

Tabla ARBOL_ADMINISTRADOR			
Propósito: almacenar las opciones del menú del sistema en forma jerárquica (árbol)			
Name	Type	Nullable	Comments
Arb_Adm_Id	Number		Código de opción de nodo
Arb_Adm_Nombre	Varchar2(100)		Nombre de opción de nodo
Arb_Adm_Nivel	Varchar2(1)		Nivel de Nodo
Arb_Adm_Id_Subnivel	Number	Y	Código alternativo o código padre
Ardol_Nodo	Varchar2(10)	Y	Nombre del nodo

Tabla AREAS			
Propósito: almacenar las áreas que de la empresa que intervienen en los procesos de control.			
Name	Type	Nullable	Comments
Are_Codigo	Varchar2(10)		Código de área
Are_Nombre	Varchar2(100)	Y	Nombre de área
Are_Usuario_insert	Varchar2(15)	Y	Usuario de insert
Area_Fecha_insert	Date	Y	Fecha de insert
Are_Usuario_update	Varchar2(15)	Y	Usuario de update
Are_Fecha_update	Date	Y	Fecha de update
Are_Codigo_alterno	Varchar2(10)	Y	Código alternativo o código padre para estructura jerárquica

Tabla ASUNTOS			
Propósito: almacenar los asuntos que se puedan tener en el levantamiento de una no conformidad.			
Name	Type	Nullable	Comments
Asu_Codigo	Varchar2(10)		Código de asunto
Asu_Nombre	Varchar2(100)	Y	Nombre de asunto
Asu_Descripcion	Varchar2(1000)	Y	Descripción de asunto
Asu_Estado	Varchar2(10)	Y	Estado de asunto 11 habilitado 12 deshabilitado

Tabla DEPARTAMENTOS			
Propósito: almacenar los departamentos y su información que existan en la empresa.			
Name	Type	Nullable	Comments
Dep_Codigo	Varchar2(10)		Código de departamento
Dep_Nombre	Varchar2(100)		Nombre de departamento
Dep_Jefeci	Varchar2(10)		Identificación de jefe
Dep_Direccion	Varchar2(100)		Dirección o ubicación de departamento
Dep_Telefono	Varchar2(10)		Teléfono
Dep_Email	Varchar2(50)		Email
Dep_Fecha_i	Date		Fecha de ingreso
Dep_Estado	Varchar2(10)	Y	Estado 11 habilitado 12 deshabilitado
Dep_Usuario_Insert	Varchar2(15)	Y	Usuario de insert
Dep_Fecha_Insert	Date	Y	Fecha de insert
Dep_Usuario_Update	Varchar2(15)	Y	Usuario de update
Dep_Fecha_Update	Date	Y	Fecha de update

Tabla DOCUMENTOS_EXPEDIENTES			
Propósito: almacenar los documentos de que se adjuntan a expedientes ingresados			
Name	Type	Nullable	Comments
Doc_Codigo	Varchar2(10)		Número de documento
Exp_Numero	Varchar2(10)		Número de expediente de no conformidad
Tra_Codigo	Varchar2(10)		Código de Trámite
Doc_Path	Varchar2(500)	Y	Path de documento
Doc_Documento	Blob	Y	Imagen de documento

Tabla ESTADOS			
Propósito: almacenar los estados de las formas y de los expedientes en forma Jerárquica			
Name	Type	Nullable	Comments
Est_Codigo	Varchar2(10)		Código de estado, estados de las formas
Est_Nombre	Varchar2(100)	Y	Nombre de estado
Est_Codigo_Alterno	Varchar2(10)	Y	Código padre de estado
Est_Descripcion	Varchar2(1000)	Y	Descripción de cada estado

Tabla EMPRESAS			
Propósito: almacenar información de la empresa como parámetros del sistema.			
Name	Type	Nullable	Comments
Emp_Codigo	Varchar2(10)		Código de empresa
Emp_Nombre	Varchar2(100)	Y	Nombre de la empresa
Cod_Emp_Descripcion	Varchar2(1000)	Y	Descripción funciones de la empresa
Emp_Direccion	Varchar2(1000)	Y	Dirección
Emp_Telefono_1	Varchar2(10)	Y	Teléfono 1
Emp_Website	Varchar2(100)	Y	Sitio web
Emp_Telefono_2	Varchar2(10)	Y	Teléfono 2
Emp_Email	Varchar2(100)	Y	Email de la empresa
Emp_Logo	Blob	Y	Logo de la empresa
Emp_Estado	Varchar2(10)	Y	Estado de la empresa
Emp_Path	Varchar2(500)	Y	Path de logo de la empresa
Emp_Ruc_Representante	Varchar2(15)	Y	R.U.C de representante legal

Tabla FORMAS_PATH			
Propósito: almacenar las formas del sistema con sus respectivas rutas de acceso.			
Name	Type	Nullable	Comments
Frm_Codigo	Varchar2(10)		Código de forma
Frm_Nombre	Varchar2(100)	Y	Nombre de la forma
Frm_Descripcion	Varchar2(1024)	Y	Descripción de la forma
Frm_Path	Varchar2(1000)	Y	Path de la forma
Frm_Usuario_insert	Varchar2(15)	Y	Usuario de insert
Frm_Fecha_insert	Date	Y	Fecha de insert
Frm_Usuario_update	Varchar2(15)	Y	Usuario de update
Frm_Fecha_update	Date	Y	Fecha de update
Arb_Adm_Id	Number		Código de opción en el árbol de administrador de menú

Tabla EXPEDIENTES			
Propósito: almacenar los expedientes de no conformidad cuando encuentren anomalías			
Name	Type	Nullable	Comments
Exp_Numero	Varchar2(10)		Número de expediente de no conformidad
Exp_Nombre	Varchar2(400)		Nombre de expediente de no conformidad
Exp_Fecha	Date		Fecha de expediente
Exp_Fecha_Expiracion	Date		Fecha de expiración
Exp_Estado	Varchar2(10)		Estado de expedientes
Exp_Tipo_Tramite	Varchar2(10)		Tipo Trámite interno
Exp_Remitente	Varchar2(15)		Usuario remitente
Exp_Destinatarario	Varchar2(15)		Usuario destinatario
Tra_Codigo	Varchar2(10)		Código de tramite
Exp_Usu_Codigo	Varchar2(15)	Y	Usuario de ingreso
Exp_Usuario_Insert	Varchar2(15)	Y	Usuario que ingreso no conformidad
Exp_Fecha_Insert	Date	Y	Fecha de ingreso de no conformidad
Exp_Usuario_Update	Varchar2(15)	Y	Usuario de update de no conformidad
Exp_Fecha_Update	Date	Y	Fecha de update de no conformidad
Tip_Tra_Codigo	Varchar2(10)	Y	Tipo tramite no conformidad
Asu_Codigo	Varchar2(10)	Y	Código de asunto
Are_Codigo	Varchar2(10)	Y	Código de área
Exp_Observacion	Varchar2(500)	Y	Observación
Prc_Codigo	Number	Y	Proceso
Exp_Analisis_Causa	Varchar2(4000)	Y	Análisis de causa
Exp_Evidencia_Cumplimiento	Varchar2(4000)	Y	Evidencia de cumplimiento
Exp_Porcentaje_Rendimiento	Number	Y	Porcentaje de rendimiento una vez finalizada no conformidad

Tabla HISTORIAL_EXPEDIENTES			
Propósito. Almacenar registros históricos (Bitácora) de los expedientes por cada evento que en ellos se efectue, si fue creado, enviado, leído y reenviado, terminado, caducado.			
Name	Type	Nullable	Comments
Exp_Numero	Varchar2(10)		Número de expediente de no conformidad
Tra_Codigo	Varchar2(10)		Código de Trámite
His_Fecha	Date		Fecha
His_Hora_Envio	Date		Hora de envío
His_Numero_Intento	Number(5)		Numero de intento
His_Numero_Envio	Number(5)		Numero de envío
His_Estado	Varchar2(10)	Y	Estado de expediente de No Conformidad
His_Hora_Recibo	Date	Y	Hora de recibo
His_Autorizado	Varchar2(15)	Y	Si expediente está autorizado x
His_Anulado	Varchar2(10)	Y	Si expediente está anulado
His_Enviado	Varchar2(10)	Y	Si expediente está enviado
His_Terminado	Varchar2(10)	Y	Si expediente está terminado
His_Caducado	Varchar2(10)	Y	Si expediente está caducado
His_Replanificado	Varchar2(10)	Y	Si expediente está replanificado
His_Fecha_Expiracion	Date	Y	Fecha de expiración
His_Creado	Varchar2(10)	Y	Si expediente está creado
His_Leido_Reenviado	Varchar2(10)	Y	Si expediente está leído y reenviado
His_Rechazado	Varchar2(10)	Y	Si expediente está rechazado
His_Aprobado	Varchar2(10)	Y	Si expediente está aprobado
Pas_Codigo	Varchar2(10)	Y	Paso
His_Usu_Codigo	Varchar2(15)		Usuario de Ingreso
His_Usu_Remitente	Varchar2(15)		Usuario Remitente
His_Usu_Destinario	Varchar2(15)		Usuario Destinatario
His_Usuario_Insert	Varchar2(15)	Y	Usuario de Insert
His_Fecha_Insert	Date	Y	Fecha de Insert
His_Usuario_Update	Varchar2(15)	Y	Usuario de Update
His_Fecha_Update	Date	Y	Fecha de Update
His_Observaciones	Varchar2(1000)	Y	Observaciones

Tabla IMAGEN_PERSONA			
Propósito: almacenar las imágenes de los empleados.			
Name	Type	Nullable	Comments
Img_Per_Codigo	Varchar2(10)		Código de imagen de persona
Per_Codigo	Varchar2(15)		Código de persona
Per_Tipo	Varchar2(10)		Tipo de persona empleado(1) proveedor(2)
Img_Per_Ruta	Varchar2(500)	Y	Path de imagen
Img_Per_imagen	Blob	Y	Imagen de persona
Img_Per_usuario_insert	Varchar2(15)	Y	Usuario de Insert
Img_Per_fecha_insert	Date	Y	Fecha de Insert
Img_Per_usuario_update	Varchar2(15)	Y	Usuario de Update
Img_Per_fecha_update	Date	Y	Fecha de Update

Tabla IMAGEN_PRODUCTOS			
Propósito: almacenar las imágenes de los productos			
Name	Type	Nullable	Comments
Prd_Codigo	Varchar2(10)		Código de producto
Img_Prđ_Codigo	Varchar2(10)		Código de imagen
Img_Prđ_Path	Varchar2(500)	Y	Path de imagen de producto
Img_Prđ_Imagen	Blob	Y	Imagen de producto

Tabla PERFILES			
Propósito: almacenar los perfiles de usuario para el tratamiento de los expedientes.			
Name	Type	Nullable	Comments
Prf_Codigo	Varchar2(10)		Código de perfil
Prf_Nombre	Varchar2(50)		Nombre de perfil
Prf_Descripcion	Varchar2(100)		Descripción de perfil
Prf_Estado	Varchar2(10)		Estado de perfil habilitado(11), inhabilitado(12)
Prf_Usuario_Insert	Varchar2(15)	Y	Usuario de insert
Prf_Fecha_Insert	Date	Y	Fecha de insert
Prf_Usuario_Update	Varchar2(15)	Y	Usuario de update
Prf_Fecha_Update	Date	Y	Fecha de update

Tabla NORMATIVAS			
Propósito: almacenar las normativas			
Name	Type	Nullable	Comments
Nor_Codigo	Varchar2(10)		Código de normativa
Nor_Nombre	Varchar2(100)		Nombre de normativa
Nor_Descripcion	Varchar2(1000)		Descripción de normativa
Nor_Codigo_Alterno	Varchar2(10)	Y	Código padre de normativa
Nor_Fecha_Promulgacion	Date		Fecha de ingreso o fecha de promulgación
Nor_Estado	Varchar2(10)		Estado de normativa habilitado(11), inhabilitado(12)
Nor_Usuario_Insert	Varchar2(15)	Y	Usuario de Insert
Nor_Fecha_Insert	Date	Y	Fecha de Insert
Nor_Usuario_Update	Varchar2(15)	Y	Usuario de Update
Tip_Nor_Codigo	Varchar2(10)	Y	Tipo de Normativa
Nor_Fecha_Update	Date	Y	Fecha de Update

Tabla PERFILES			
Propósito: almacenar los perfiles de usuario para el tratamiento de los expedientes.			
Name	Type	Nullable	Comments
Prf_Codigo	Varchar2(10)		Código de perfil
Prf_Nombre	Varchar2(50)		Nombre de perfil
Prf_Descripcion	Varchar2(100)		Descripción de perfil
Prf_Estado	Varchar2(10)		Estado de perfil habilitado(11), inhabilitado(12)
Prf_Usuario_Insert	Varchar2(15)	Y	Usuario de Insert
Prf_Fecha_Insert	Date	Y	Fecha de Insert
Prf_Usuario_Update	Varchar2(15)	Y	Usuario de Update
Prf_Fecha_Update	Date	Y	Fecha de Update

Tabla PERFILES_USUARIOS			
Propósito: almacenar perfiles asignados a los usuarios del sistema			
Name	Type	Nullable	Comments
Prf_Codigo	Varchar2(10)		Código de perfil foreign de perfiles
Usu_Codigo	Varchar2(15)		Código de usuario foreign de usuarios

Tabla PERSONA			
Propósito: almacenar la información relacionada como persona como clase.			
Name	Type	Nullable	Comments
Per_Codigo	Varchar2(15)		Código de persona
Per_Tipo	Varchar2(10)		Tipo de persona empleado(1), proveedor(2)
Per_Ci	Varchar2(15)	Y	Cedula de identificación
Per_Ruc	Varchar2(15)	Y	Ruc
Per_Pasaporte	Varchar2(15)	Y	Pasaporte
Per_Nombre	Varchar2(200)	Y	Nombre de persona
Per_Tipo_Id	Varchar2(10)	Y	Tipo de identificación cedula(c), pasaporte(p), ruc(r)
Per_Fecha_n	Date		Fecha de nacimiento
Per_Direccion	Varchar2(100)		Dirección
Per_Sexo	Varchar2(1)		Masculino(m), Femenino (f)
Per_Edad	Number		Edad
Per_Telefono	Varchar2(10)		Teléfono
Per_Salario	Number	Y	Salario
Per_Estado	Varchar2(10)	Y	Estado
Per_Usuario_insert	Varchar2(15)	Y	Usuario de Insert
Per_Fecha_insert	Date	Y	Fecha de Insert
Per_Usuario_update	Varchar2(15)	Y	Usuario de Update
Per_Fecha_update	Date	Y	Fecha de Update
Dep_Codigo	Varchar2(10)	Y	Código de departamento foreing de departamentos

Tabla PROCESOS			
Propósito: almacenar los procesos que forman parte del sistema			
Name	Type	Nullable	Comments
Prc_Codigo	Number		Código de proceso
Prc_Codigo_Alterno	Varchar2(10)	Y	Código padre de proceso
Prc_Nombre	Varchar2(100)	Y	Nombre de proceso
Prc_Descripcion	Varchar2(500)	Y	Descripción de procesos
Prc_Usuario_insert	Varchar2(15)	Y	Usuario de Insert
Prc_Fecha_insert	Date	Y	Fecha de Insert
Prc_Usuario_update	Varchar2(15)	Y	Usuario de Update
Prc_Fecha_update	Date	Y	Fecha de Update

Tabla PRODUCTOS			
Propósito: almacenar información de los productos que fabrica la empresa			
Name	Type	Nullable	Comments
Prd_Codigo	Varchar2(10)		Código de producto
Prd_Nombre	Varchar2(100)		Nombre de producto
Prd_Descripcion	Varchar2(500)		Descripción de producto
Prd_Codigo_Alterno	Varchar2(10)	Y	Código padre de producto
Prd_Clise	Varchar2(10)	Y	Número de clise
Prd_Troquel	Varchar2(10)	Y	Troquel
Prd_Precio_1	Number	Y	Precio 1
Prd_Precio_2	Number	Y	Precio 2
Prd_Precio_3	Number	Y	Precio 3
Prd_Peso	Number	Y	Peso
Prd_Largo	Number	Y	Largo
Prd_Ancho	Number	Y	Ancho
Prd_Alto	Number	Y	Alto
Prd_Test	Number	Y	Test
Prd_Faluta	Number	Y	Flauta
Prd_Largo_Lamina	Number	Y	Largo lamina
Prd_Cantidad_Cajas	Number	Y	Cantidad de cajas
Prd_Stock_Minimo	Number	Y	Stock mínimo
Prd_Stock_Maximo	Number	Y	Stock máximo
Prd_Observacion	Varchar2(500)	Y	Observación de producto

Tabla PRODUCTOS_EXPEDIENTES			
Propósito: almacenar información de los productos con evidencia de no conformidad			
Name	Type	Nullable	Comments
Exp_Numero	Varchar2(10)		Número de expediente de no conformidad
Per_Codigo	Varchar2(15)	Y	Código de persona
Per_Tipo	Varchar2(10)	Y	Tipo de persona
Prd_Codigo	Varchar2(10)	Y	Código de producto
Tra_Codigo	Varchar2(10)		Código de tramite foreing tramites
Prd_Exp_precio_1	Number	Y	Precio 1 si se encuentra evidencia de no conformidad
Prd_Exp_precio_2	Number	Y	Precio 2 si se encuentra evidencia de no conformidad
Prd_Exp_precio_3	Number	Y	Precio 3 si se encuentra evidencia de no conformidad
Prd_Exp_peso	Number	Y	Peso si se encuentra evidencia de no conformidad
Prd_Exp_largo	Number	Y	Largo si se encuentra evidencia de no conformidad
Prd_Exp_ancho	Number	Y	Ancho si se encuentra evidencia de no conformidad
Prd_Exp_alto	Number	Y	Alto si se encuentra evidencia de no conformidad
Prd_Exp_test	Number	Y	Test si se encuentra evidencia de no conformidad
Prd_Exp_flauta	Number	Y	Flauta si se encuentra evidencia de no conformidad
Prd_Largo_lamina	Number	Y	Largo lamina si se encuentra evidencia de no conformidad
Prd_Exp_cantidad_cajas	Number	Y	Cantidad si se encuentra evidencia de no conformidad
Prd_Exp_stock_maximo	Number	Y	Stock mínimo si se encuentra evidencia de no conformidad
Prd_Exp_stock_minimo	Number	Y	Stock máximo si se encuentra evidencia de no conformidad
Prd_Exp_observacion	Varchar2(500)	Y	Observación si se encuentra evidencia de no conformidad
Ord_Pr_codigo	Number	Y	
Prd_Exp_Existe_Orden	Varchar2(1)	Y	Si existe orden

Tabla REQUISITOS			
Propósito: almacenar la información de los requisitos que forman parte de una normativa			
Name	Type	Nullable	Comments
Req_Codigo	Varchar2(10)		Código de requisito
Req_Tipo	Varchar2(10)		Tipo de requisito
Req_Nombre	Varchar2(500)		Nombre de requisito
Req_Codigo_Alterno	Varchar2(10)	Y	Código padre de requisito
Req_Descripcion	Varchar2(4000)		Descripción de requisito
Req_Obligatoriedad	Varchar2(2)		Obligatoriedad si o no
Req_Estado	Varchar2(10)		Estado habilitado(11) inhabilitado(12)
Req_Usuario_insert	Varchar2(15)	Y	Usuario de Insert
Req_Fecha_insert	Date	Y	Fecha de Insert
Req_Usuario_update	Varchar2(15)	Y	Usuario de Update
Req_Fecha_update	Date	Y	Fecha de Update
Nor_Codigo	Varchar2(10)	Y	Código de normativa foreing de normativa
Prc_Codigo	Number	Y	Código de proceso foreing de procesos

Tabla REQUISITOS_EXPEDIENTES			
Propósito: almacenar los requisitos que se incumplen cuando se ingresan en un expediente			
Name	Type	Nullable	Comments
Req_Codigo	Varchar2(10)		Código requisito
Req_Tipo	Varchar2(10)		Tipo de requisito
Tra_Codigo	Varchar2(10)		Código de tramite
Exp_Numero	Varchar2(10)		Numero expediente no conformidad
Req_Exp_nombre	Varchar2(100)		Nombre requisito
Req_Exp_observaciones	Varchar2(500)	Y	Observaciones
Req_Exp_verificado	Varchar2(2)	Y	Verificado

Tabla TIPOS NORMATIVAS			
Propósito: almacenar los tipo de normativas			
Name	Type	Nullable	Comments
Tip_Nor_Codigo	Varchar2(10)		Tipo de Normativa
Tip_Nor_Nombre	Varchar2(100)	Y	Nombre
Tip_Nor_Descripcion	Varchar2(200)	Y	Descripción

Tabla TIPOS TRAMITES			
Propósito: almacenar los tipos de tramites de una empresa referenciando a un proceso			
Name	Type	Nullable	Comments
Tip_Tra_Codigo	Varchar2(10)		Código de tipo de tramite
Tip_Tra_Secuencia	Number		Secuencia de tramites
Tip_tra_nombre	Varchar2(100)	Y	Nombre de tramite hace referencia a un proceso
Tip_tra_descripcion	Varchar2(1000)	Y	Descripción de un tramite
Tip_tra_usuario_insert	Varchar2(15)	Y	Usuario de Insert
Tip_tra_fecha_insert	Date	Y	Fecha de Update
Tip_tra_usuario_update	Varchar2(15)	Y	Usuario de Update
Tip_tra_fecha_update	Date	Y	Fecha de Update
Nor_codigo	Varchar2(10)	Y	Código de normativa foreing de normativas

Tabla USUARIOS			
Propósito: almacenar los usuarios que son parte del sistema			
Name	Type	Nullable	Comments
Usu_codigo	Varchar2(15)		Código de usuario
Usu_clave_acceso	Varchar2(15)		Calve de acceso
Usu_estado	Varchar2(10)		Estado habilitado(11), inhabilitado(12)
Usu_per_nombre	Varchar2(100)		Nombre de persona
Usu_usuario_insert	Varchar2(15)	Y	Usuario de Insert
Usu_fecha_insert	Date	Y	Fecha de Insert
Usu_usuario_update	Varchar2(15)	Y	Usuario de Update
Usu_fecha_update	Date	Y	Fecha de Update
Per_tipo	Varchar2(10)	Y	Tipo de persona empleado(1)
Per_codigo	Varchar2(15)	Y	Código de persona

Tabla INFORMES			
Propósito: almacenar los informes(reportes) con su respectiva ruta de acceso y formato de impresión que hace referencia			
Name	Type	Nullable	Comments
Inf_codigo	Varchar2(10)		Código de informe
Inf_nombre	Varchar2(100)	Y	Nombre de informe
Inf_estado	Varchar2(10)	Y	Estado de informe
Frm_codigo	Varchar2(10)	Y	Código de forma
Frm_imp_codigo	Varchar2(10)		Código de formato de impresión

3.2 Modelo de Procesos

3.2.1 Diagrama de Contexto

También llamado Flujo de Datos de Contexto, es el modelo de procesos más general que puede utilizarse en un sistema o aplicación. El sistema o aplicación se representa como un único proceso. Sus interacciones con la institución otros sistemas y el mundo exterior se dibujan como flujos de datos de entradas y salidas.

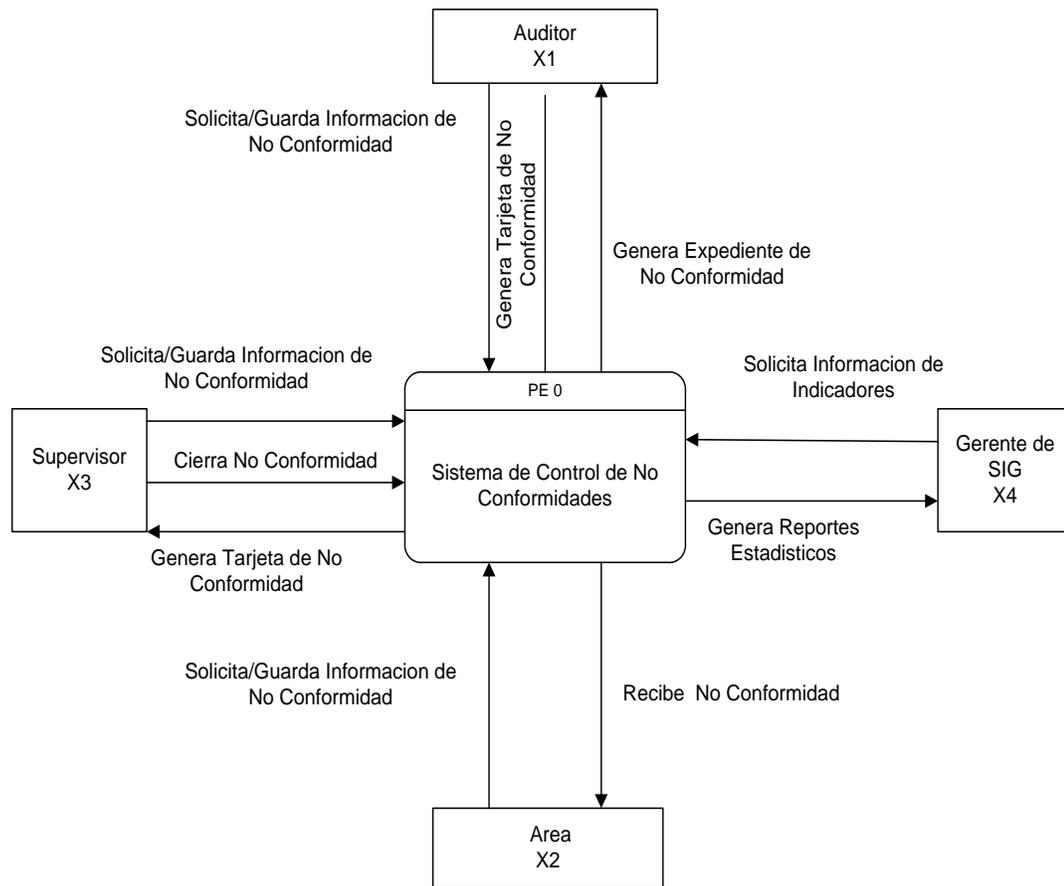


Figura 3.3 Diagrama de Contexto

3.2.2 Diagrama de Descomposición

Un diagrama de descomposición, también denominado gráfico de jerarquías, muestra la estructura o descomposición funcional en sentido descendente, de un sistema. También proporciona un esquema para elaborar nuestros Diagramas de Flujos de Datos (DFD).

El único símbolo utilizado en el diagrama de descomposición es el símbolo de proceso, por lo demás el mismo símbolo se utiliza en los DFD. Los procesos se conectan entre si en una estructura arborescente. Los nombres de los procesos deberán ajustarse a las directrices sobre nombres explicados para los DFD, el proceso superior también llamado **hizo**, representa a todo el sistema cuyas necesidades se están definiendo, el proceso raíz se desglosa o divide en subsistemas, funciones y tareas, donde el número de niveles depende enteramente de la dimensión del proyecto.

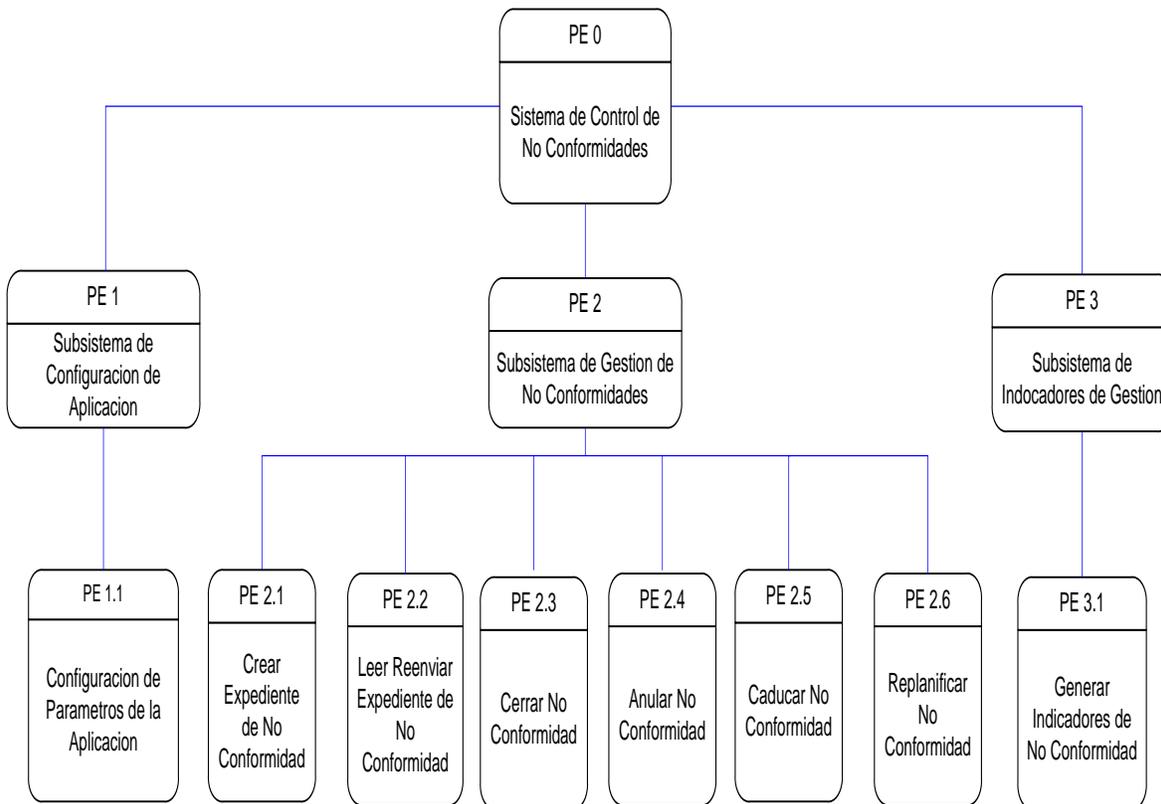


Figura 3.4 Diagrama de Descomposición

3.2.3 Diagrama de Flujo de Datos del nivel 1

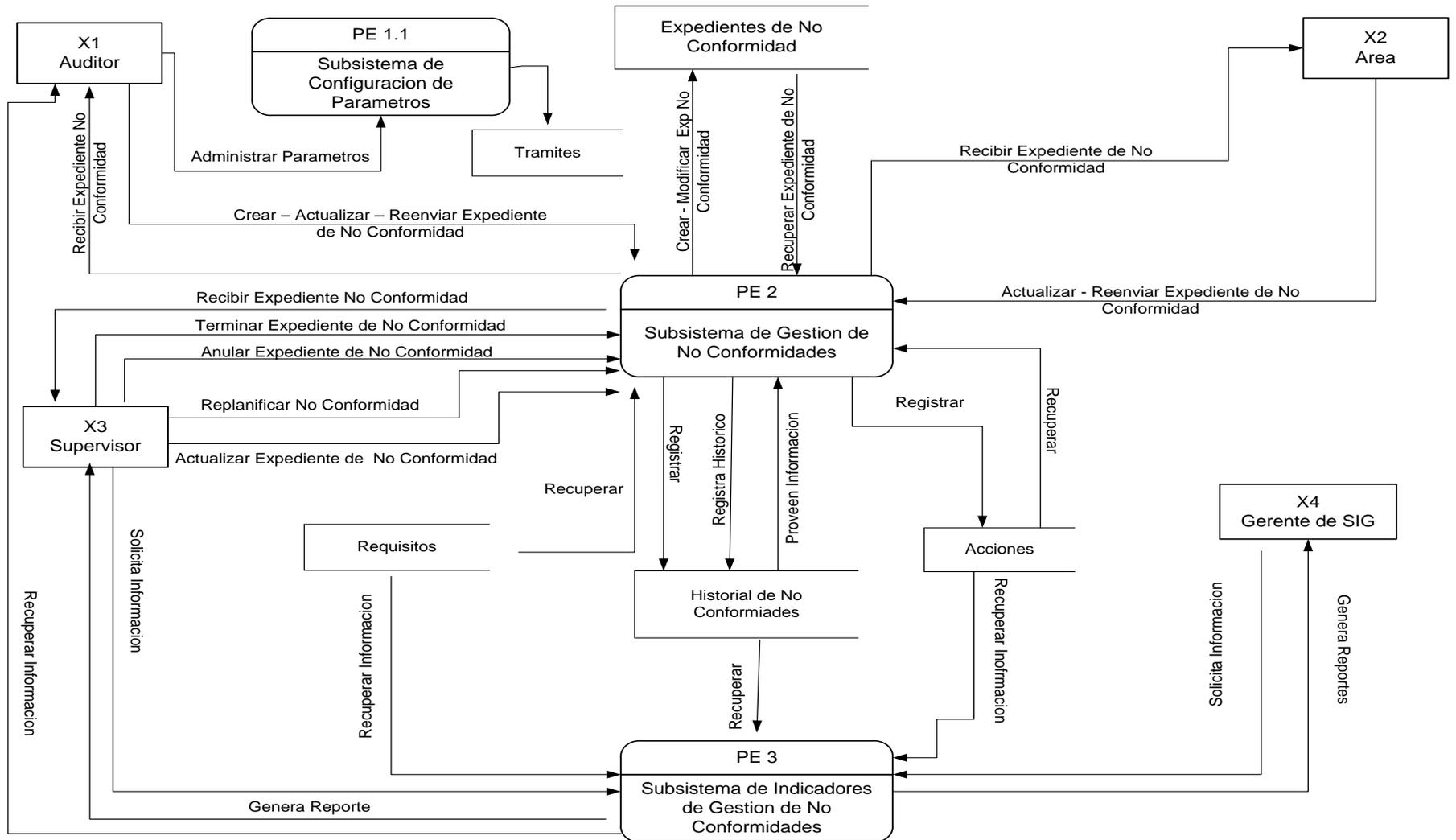


Figura 3.5 Diagrama de Flujo de Datos Nivel 1

3.2.4 Diagrama de Flujo de Datos del nivel 2

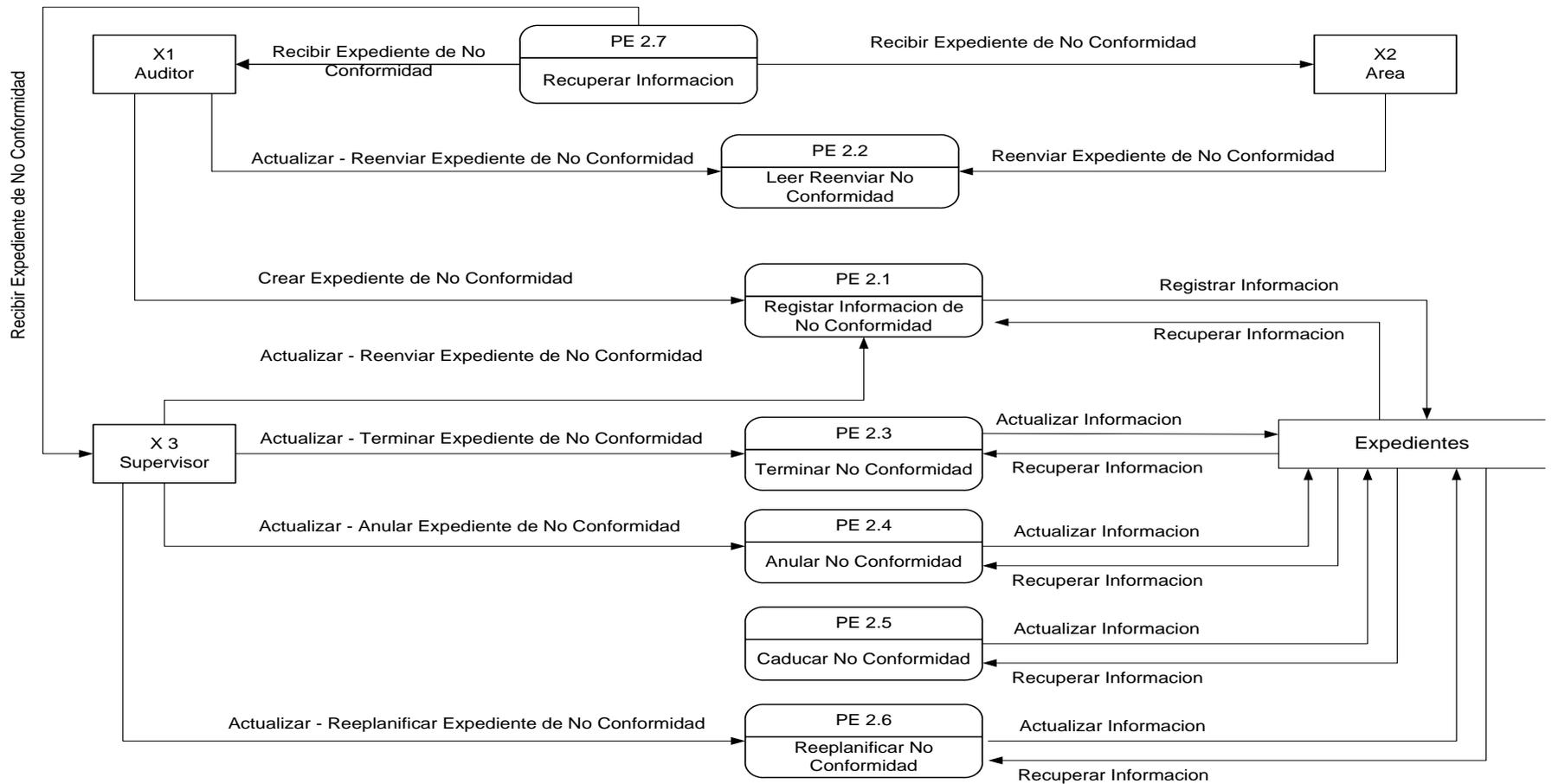


Figura 3.6 Diagrama de Flujo de Datos Nivel 2

CAPITULO 4

DESARROLLO DE LA APLICACION

4.1 Creación de las definiciones de la Base de Datos

Para esta aplicación se ha utilizado el Gestor de Base de datos Oracle, tomando en consideración la robustez y seguridad que nos ofrece Oracle.

La Base de Datos residirá en un equipo instalado el Sistema Operativo Server 2003.

Pasos para la Instalación:

- Insertar el DVD de instalación abrimos y seleccionamos el instalador para nuestro caso: D:\10201_database_win32\database, donde seleccionamos **setup.exe**.

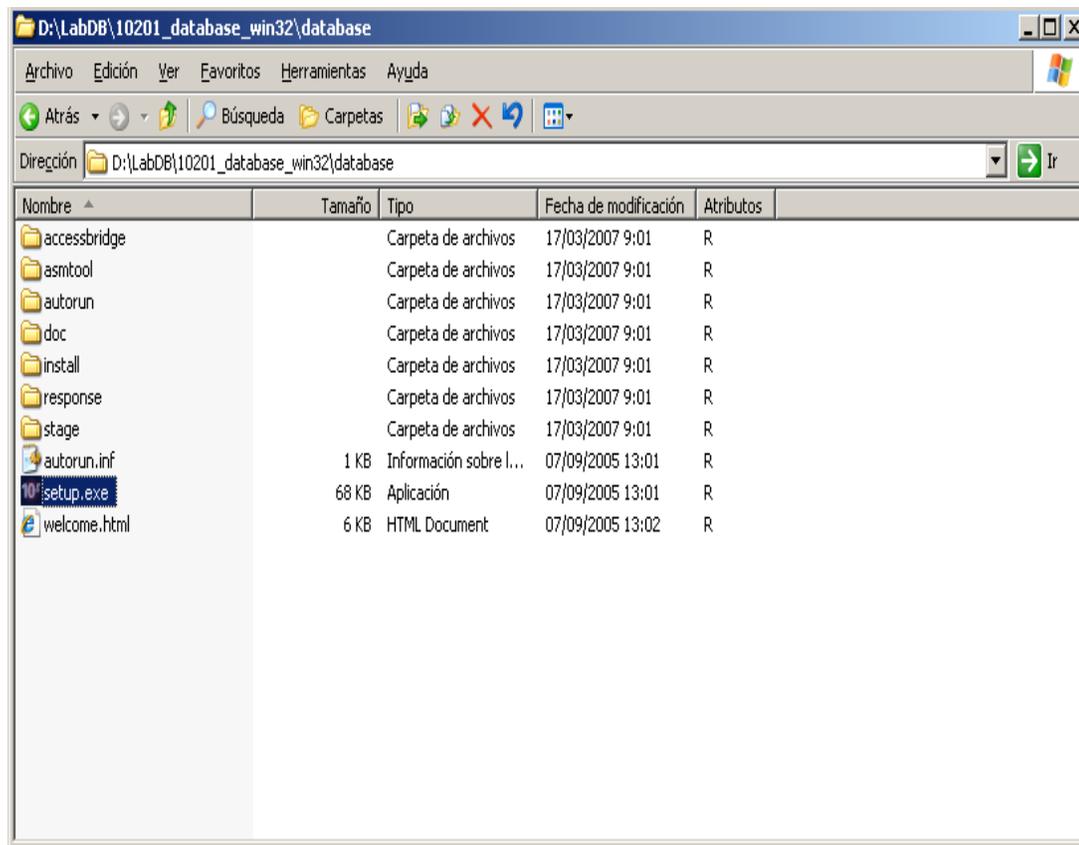


Figura 4.1 Instalación de Base de Datos Oracle (Inicio).

Aparecerá esta ventana donde ingresaremos el nombre de la base de datos global así como las contraseñas.

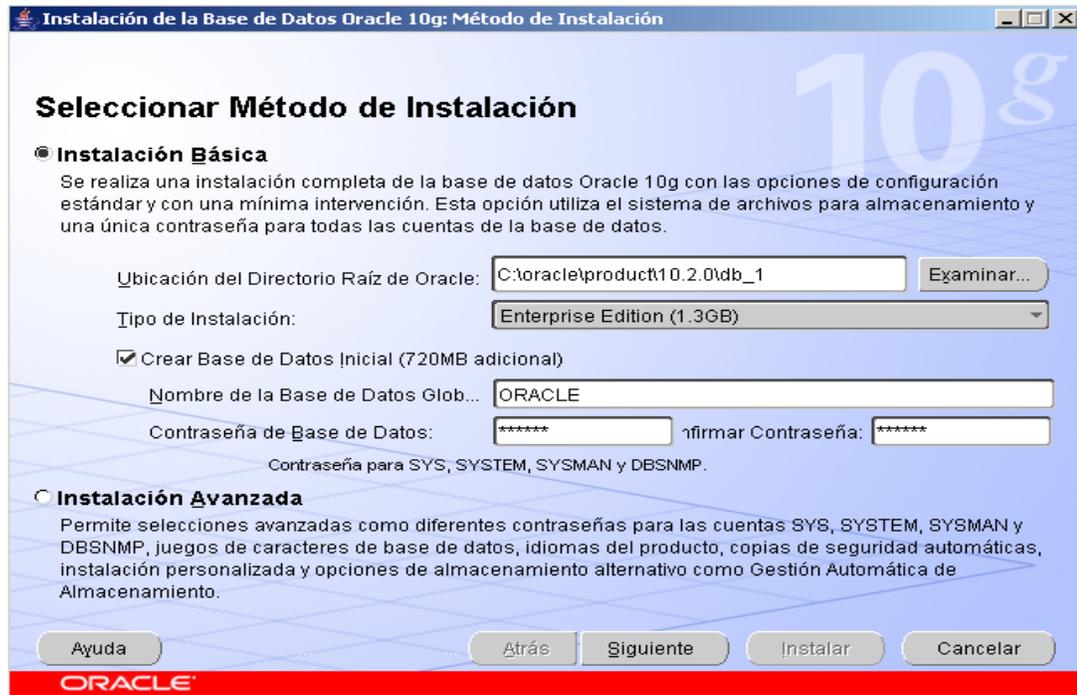


Figura 4.2 Selección de Método de Instalación

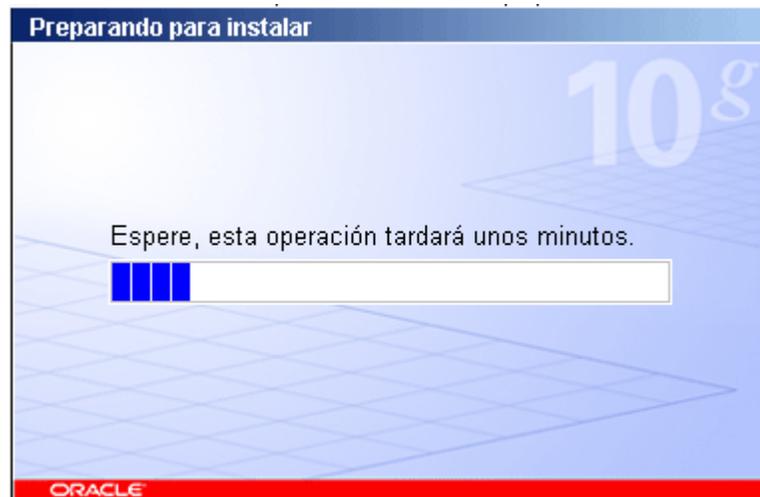


Figura 4.3 Preparación del Intaller

Luego aparecerá esta ventana donde se realiza las Comprobaciones de Requisitos del Producto

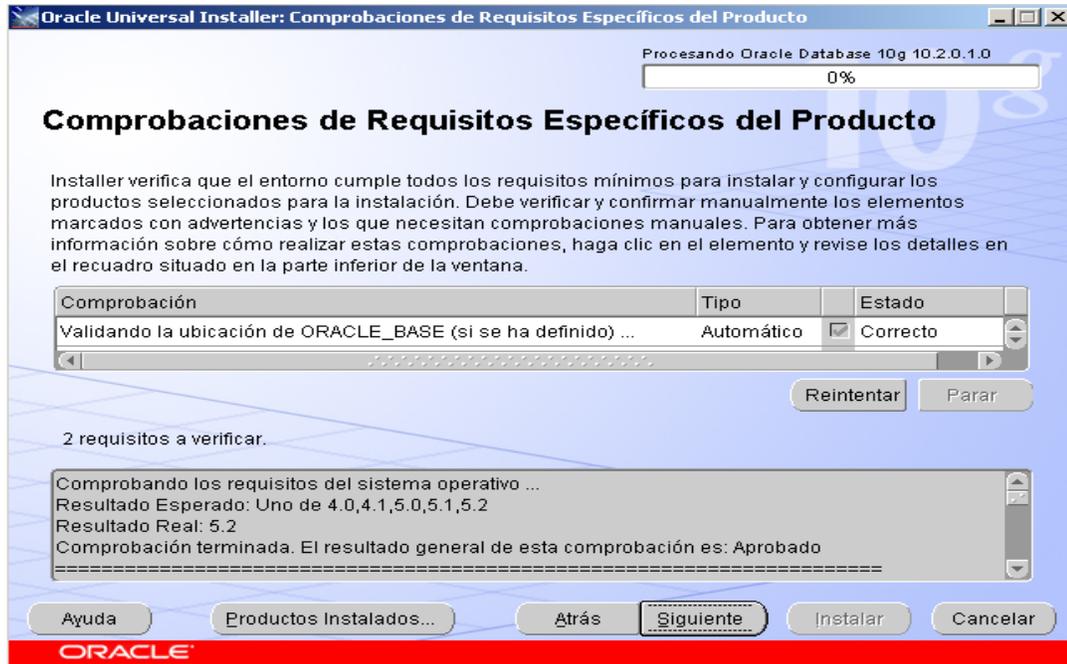


Figura 4.4 Comprobaciones de Requisitos Específicos del Producto

Dar clic en siguiente y aparecerá la ventana de resumen de los productos que se van a instalar, dar click en Instalar

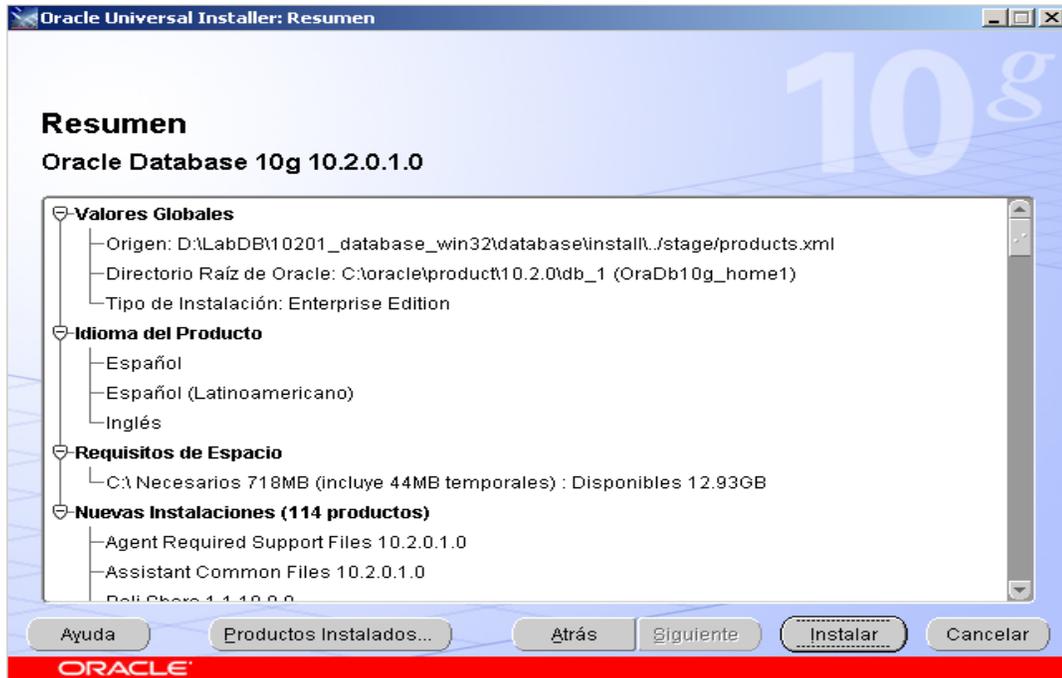


Figura 4.5 Resumen de los Productos que se van a Instalar

Una vez seleccionada la opción Instalar aparecerá esta ventana, donde se detalla el progreso de instalación

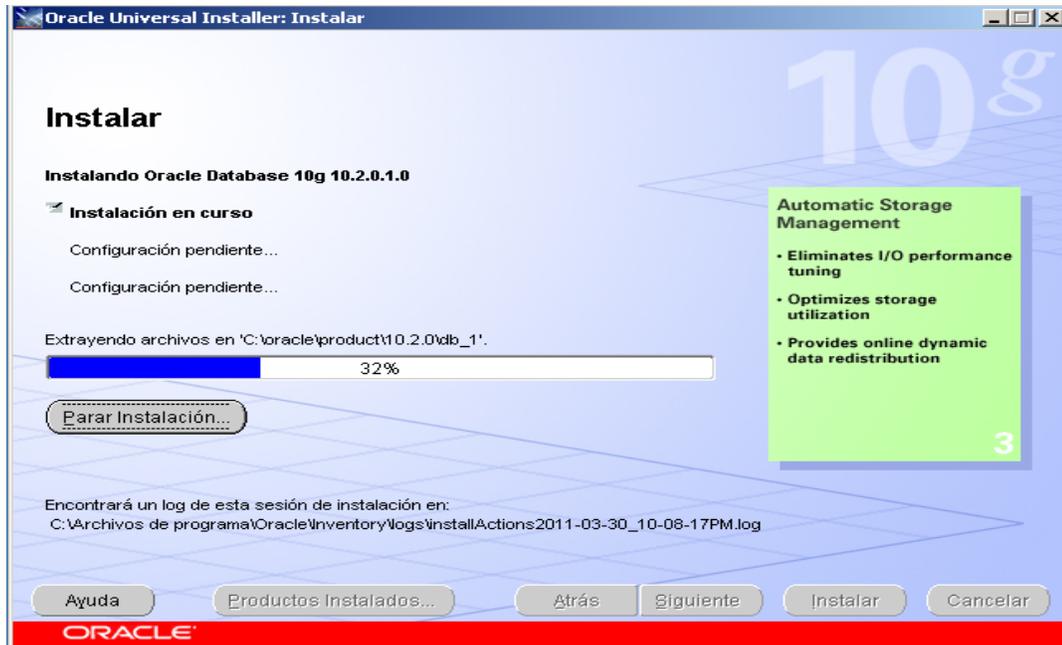


Figura 4.6 Progreso de la Instalación

Luego aparecerá una ventana de Asistentes de Configuración que es en donde se configuran e inician los componentes seleccionados.



Figura 4.7 Asistente de Configuración

Esta ventana informa el progreso de la instalación.

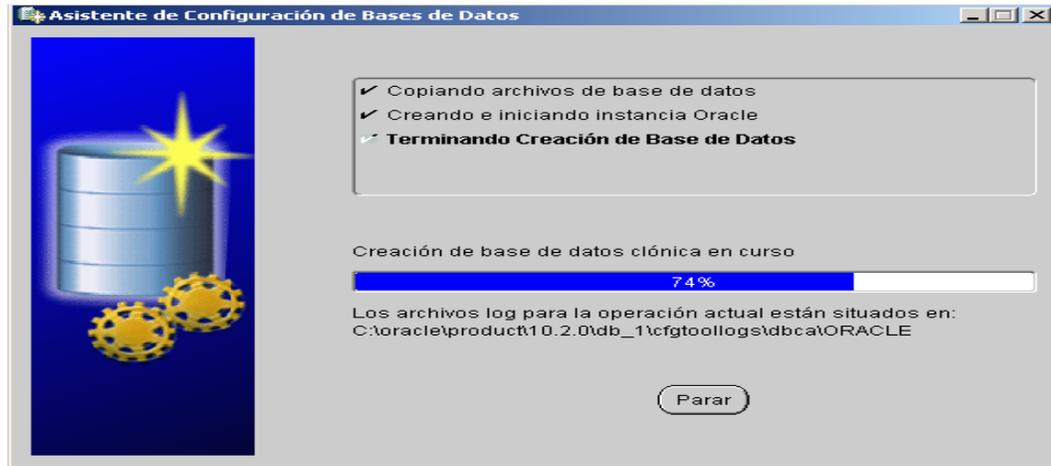
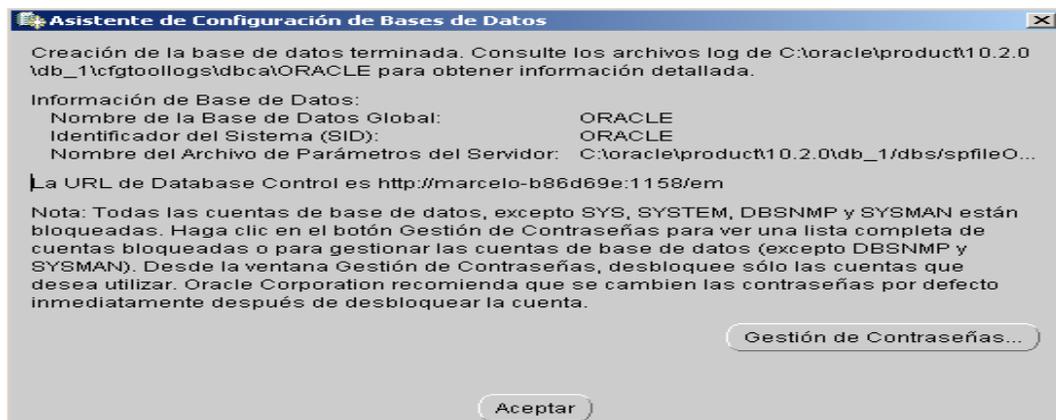


Figura 4.8 Asistente de Configuración

En este paso dar clic en Gestión de Contraseñas



Ingresar las contraseñas para los usuarios SYS, SYSTEM, HR, SCOTT luego dar click en aceptar



Figura 4.9 Gestión de Contraseñas

Se pueden modificar las contraseñas mediante código Plsql.

Cuando aceptamos se presentará esta ventana en donde se indicará el fin de la instalación, y el estado de la instalación, para nuestro caso ha sido correcta luego damos click en salir.

Esta ventana también da una información en la que se recalca la dirección Url de Enterprise Manager Database Control, que se accesa por el puerto 1158.

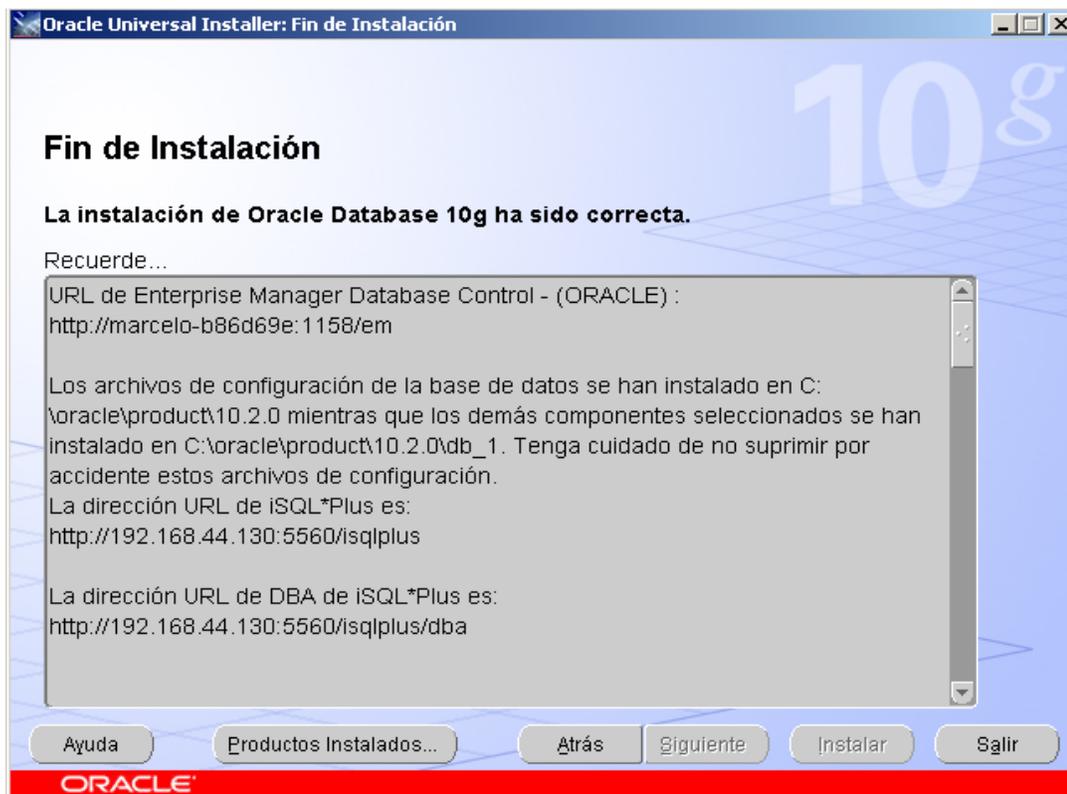


Figura 4.10 Fin de Instalación

Para poder acceder y comprobar lo instalado procedemos a ingresar al Enterprise Manager o por medio de Oracle SQL *Plus o Plsql Developer

Pasos para la creación de Tablespace y DataFiles

Ingresamos a Oracle SQL *Plus, y nos conectamos con nuestras credenciales de administrador.

```
CREATE TABLESPACE GESTION_TRAMITES  
DATAFILE  
'C:\ORACLE\PRODUCT\10.2.0\ORADATA\ORACLE\GESTION_TRAMITES1.DBF'  
SIZE 64M
```

Es en este Tablespace donde residirán las tablas de este proyecto.

Pasos para la creación de Usuarios

```
CREATE USER SG_TRAMITES IDENTIFIED BY SG_TRAMITES  
DEFAULT TABLESPACE GESTION_TRAMITES  
TEMPORARY TABLESPACE TEMP;
```

Pasos para crear Roles, asignar permisos y asignar a Usuario

```
GRANT DBA TO SG_TRAMITES;  
CREATE ROLE ACCESO_TOTAL IDENTIFIED BY ACCESO_TOTAL;  
GRANT DBA TO ACCESO_TOTAL;  
GRANT ACCESO_TOTAL TO SG_TRAMITES;
```

Creación y Generación de las Tablas

Para la creación de las Tablas utilizaremos la Herramienta ERwin Data Modeler en donde hemos desarrollado nuestro modelo relacional, es aquí que con el generador de la herramienta procederemos a crear la Base de Datos Física de este proyecto desde la herramienta mencionada.

Para hacer esto posible debemos conectarnos a la Base de Datos creada de la siguiente manera:

Seleccionar en la barra de herramientas de ERwin abrimos el menú de la Opcion Database y seleccionamos Database Connection luego ingresamos nuestras credenciales

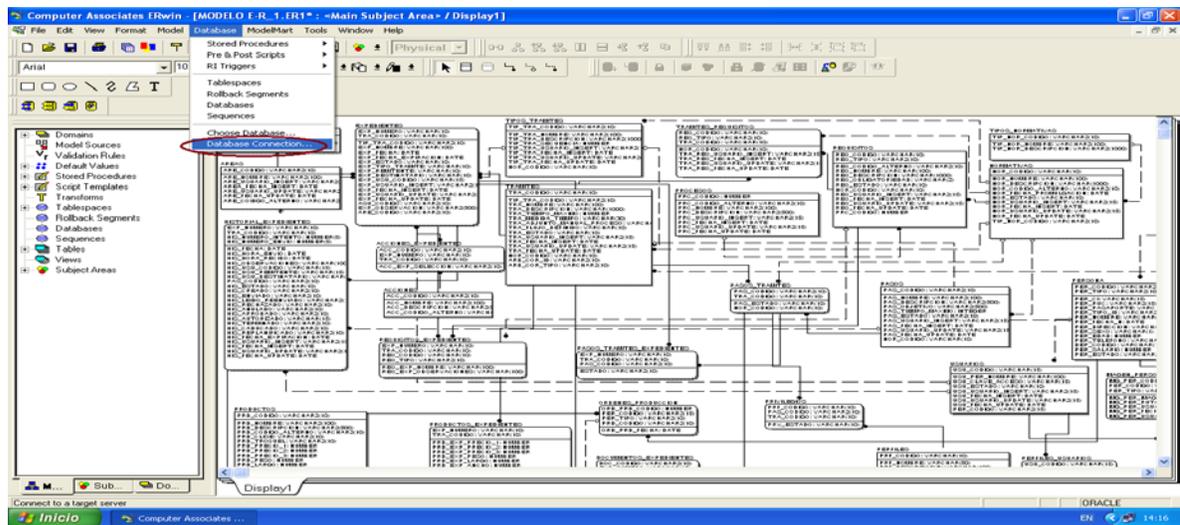


Figura 4.11 Conexión a Base de Datos

Una vez conectados a nuestra base de datos procedemos a generar las tablas

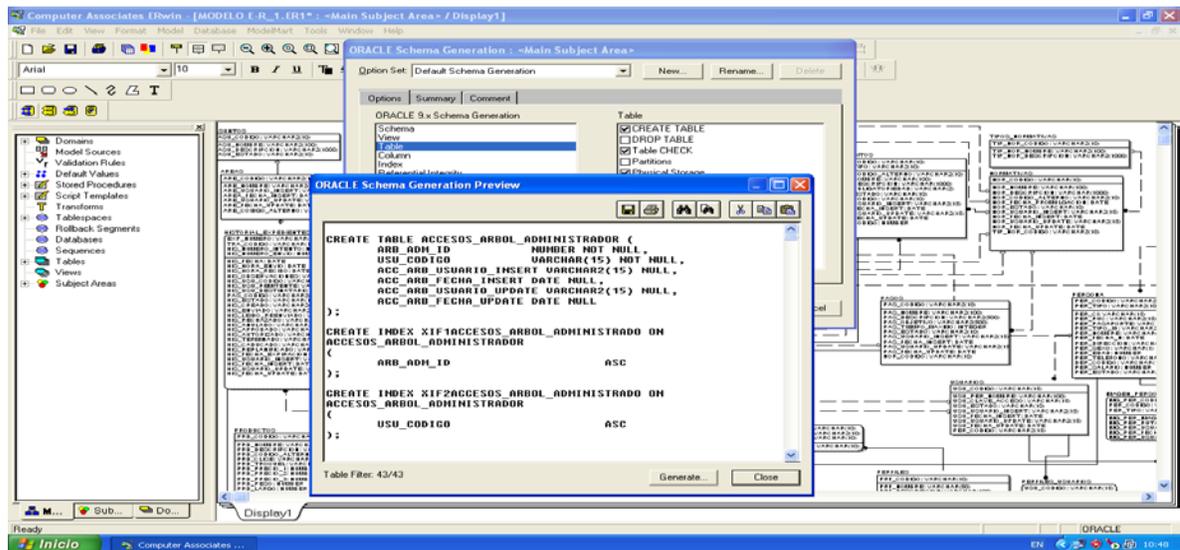


Figura 4.11 Generación de Tablas en Base de Datos

4.2 Programación con Herramientas Developer Suite de Oracle.

Oracle Developer Suite es el entorno de desarrollo más completo e integrado, que combina el poder de desarrollo de aplicaciones y herramientas de inteligencia empresarial en una única suite que se basa en los últimos estándares de la industria. Oracle Developer Suite permite a los desarrolladores crear rápidamente aplicaciones transaccionales de alta calidad, que pueden ser desplegados en múltiples canales - incluyendo portales, servicios Web y dispositivos inalámbricos, y que se puede ampliar con capacidades de inteligencia de negocios. Para nuestro caso de estudio utilizaremos dos herramientas de este grupo, Oracle Forms y Oracle Reports.

Oracle Forms Es considerada de 4ta generación y permite construir aplicaciones transaccionales muy robustas. Ofrece ventajas como un estándar de programación plenamente adaptado a las normas de Sql Oracle; un entorno de programación mejorado para facilitar el desarrollo. Oracle Forms se compone de tres programas principales:

- **Forms Builder** crea las pantallas y da funcionalidad a las mismas.
- **Forms Generate** convierte el fichero en un fichero seudoejecutable.
- **Forms Runtime** ejecuta el seudoejecutable.

Es importante saber que Forms no genera ficheros *.exe.

Forms genera tres tipos de ficheros principales sin los que una aplicación nunca podrá existir en Forms:

- *.Fmb incluye todas las partes que intervienen en el desarrollo, pantallas, procedimientos, fruncionalidad.
- *.Fmx es el seudoejecutable.

Otros tipos de archivos que genera Forms son menús y librerías.

Menús:

- *.mmb fichero fuente manual del menú.
- *.mmt fichero fuente en modo texto del menú.

- *.mmx fichero seudoejecutable del menú.

Librerías:

- *.pll código fuente
- *.pld código fuente en modo texto.
- *.plx ejecutable.

4.2.1 Objetos y Propiedades.

Objetos son todos aquellos que se puedan insertar en una pantalla y que permitan interactuar con el exterior, cada uno de estos objetos tienen propiedades y se agrupan de acuerdo a sus características, estos pueden ser cuadros de texto, botones, lienzos, etc

Cuando se genera un objeto automáticamente se crean sus propiedades que tomarán un valor por defecto inicial dado por Forms, dichas propiedades se podrán modificar en tiempo de diseño o en tiempo de ejecución dependiendo del objeto y de la propiedad, y lo que es más importante dichas propiedades se comportarán como variables de tipo global en cuanto a su contenido, es decir si son modificadas mantendrán el valor asignado a lo largo del tiempo de ejecución o hasta que su valor sea nuevamente alterado.

Los objetos se dividen en:

- **Items:** Interfaz de comunicación entre el usuario y la aplicación, textos, botones, listas, etc.
- **Bloques:** Agrupaciones de ítems. Estos bloques pueden estar asociados a tablas o no, es más todo ítem debe estar asociado a un bloque de tal modo que será necesario crear al menos un bloque por si solo existe un ítem en la aplicación.

4.2.2 Trigger

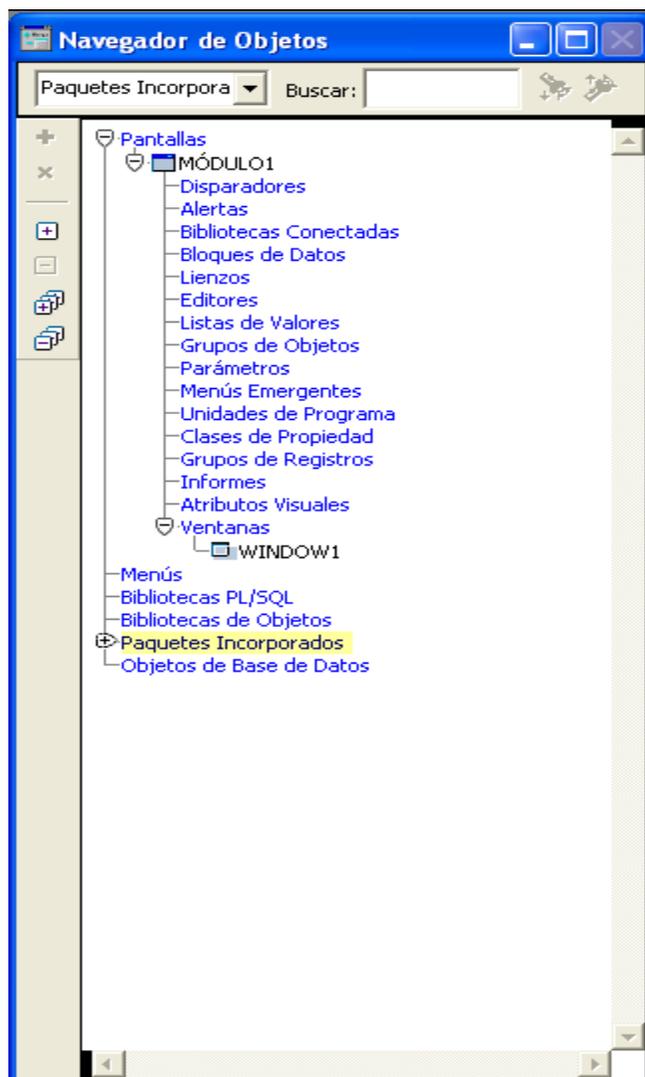
Permiten dar funcionalidad a los eventos (instrucciones asociadas a un ítem cuando se produce un evento). No deben confundirse con los eventos de forma aislada, un trigger es un conjunto formado por un evento y unas instrucciones, es un concepto algo distinto del simple concepto de evento que existe en otros entornos de desarrollo visuales, debe ser tratado de una forma más conceptual que práctica, su dominio requiere un poco de experiencia. No obstante podemos encontrar referencias a un trigger solamente como disparador o evento.

Los triggers principales en Forms son:

- **-Key:** apenas utilizado en Forms, son ejecutados al pulsar una tecla o combinación de teclas, principalmente se usan en entornos de modo texto.
- **-When:** el más utilizado, son desencadenados por múltiples eventos
- **-On:** interviene en acciones por defecto del sistema, On Insert, On Delete.
- **-Pre:** se disparan antes de ejecutar el evento correspondiente. PRE INSERT, por ejemplo para comprobar condiciones específicas de inserción en función de unos valores.
- **-Post:** misma funcionalidad que el anterior pero a la inversa temporalmente Post Insert, Post Commit

4.2.3 Elementos de Diseño.

Navegador de Objetos (Object Navigator): En el visualizaremos tanto los objetos efectivamente instanciados o incluidos en nuestra aplicación como los que tenemos a nuestra disposición para ser incluidos (lo que podríamos denominar para entendernos; clases) .



Como se puede apreciar en la Imagen 1 el navegador despliega los componentes que intervienen en toda aplicación desarrollada con Forms, estos son los Módulos, Menús, Bibliotecas, Paquetes Incorporados y Objetos de la Base de Datos.

Módulos: Cada módulo puede contener una aplicación suficiente pero podremos trabajar con que incluyan uno o más módulos.

Menús: Menús creados por el programador.

Bibliotecas: Funciones y procedimientos creados por el programador

Paquetes incorporados: funciones y procedimientos almacenados en la BBDD accesibles en todo momento para el programador, por ejemplo el paquete DBMS_OUTPUT tiene una serie de funciones asociadas, entre ellas PUTLINE, que permite imprimir datos por pantalla.

Este componente incluye todas las funciones y controles integrados por Oracle en Forms.

Objetos de la base de datos: aquí encontramos una imagen de las tablas y vistas a las que tenemos acceso, además de otros objetos de la base de datos a la que estemos accediendo.

Pasamos a describir el contenido de el componente MODULO1 (nombre asignado al modulo por defecto al entrar en un Forms “vacío”, lo habitual es que los programadores asignen al modulo el nombre del fichero fmb que va a contener la aplicación.)

Como se comenta en la descripción del modulo, cada modulo puede contener una aplicación autosuficiente pero podremos trabajar con aplicaciones que incluyan uno o más módulos, cuantos más módulos incluyamos más difícil será trabajar con ellos dado que el numero de objetos en el navegador aumenta rápidamente.

Disparadores: aquí se encuentran todos los eventos posibles en una aplicación Forms, incluidos eventos propios de Forms y eventos Windows (como la carga de un form en memoria, la pulsación de teclas o el movimiento del ratón)

Alertas: avisos predefinidos por la aplicación como mensajes de error o advertencias.

Bibliotecas incorporadas: funciones y procedimientos locales al modulo y por tanto solo accesibles por este.

Bloques de Datos: Bloques lógicos de ítems.

Vistas Lienzo(Canvas): Todas las páginas de la aplicación. Toda aplicación Forms como hemos podido comprobar parte de un canvas o lienzo, es imposible hacer nada sin este elemento y al menos un control, pasemos ahora a analizar más en detalle este elemento.

Editores: sirven para poder escribir comentarios en un ítem de la aplicación.

Listas de valores: listas que permitirán la elección de un campo dentro de las mismas.

Grupos de objetos: nos permitirán heredar propiedades, agregar nuevas, tener objetos hijos, etc.

Parámetros: valores pasados a la aplicación con algún fin.

Unidades de Programa, funciones y procedimientos creados por el usuario

Clases de propiedad: conjunto de propiedades para crear objetos idénticos (varios botones iguales por ejemplo).

Atributos visuales: es idéntico a clases de propiedad pero con atributos gráficos.

Ventanas: ventanas del modulo.

Grupo de Registros: cursores internos creados por Forms

Es importante saber que los objetos visuales se distinguirán de las variables del código PL/SQL por un prefijo que será el símbolo gramatical ':' de tal modo que para referenciar a un ítem en código PL/SQL se antepondrá dicho símbolo. El lector habrá advertido que los objetos visuales son tratados como una variable más en el código ya que en el fondo dichos objetos no son más que eso, variables desde el punto de vista del programador de Forms.

4.2.4 Funciones Regalas y Comandos para el manejo de Fechas, Cadenas y Elementos de Control

Funciones de valores simples:	
ABS(n)	Devuelve el valor absoluto de (n).
CEIL(n)	Obtiene el valor entero inmediatamente superior o igual a "n".
FLOOR(n)	Devuelve el valor entero inmediatamente inferior o igual a "n".
MOD (m, n)	Devuelve el resto resultante de dividir "m" entre "n".
NVL (valor, expresión)	Sustituye un valor nulo por otro valor.
POWER (m, exponente)	Calcula la potencia de un número.
ROUND (numero [, m])	Redondea números con el número de dígitos de precisión indicados.
SIGN (valor)	Indica el signo del "valor".
SQRT(n)	Devuelve la raíz cuadrada de "n".
TRUNC (numero, [m])	Trunca números para que tengan una cierta cantidad de dígitos de precisión.
VARANCE (valor)	Devuelve la varianza de un conjunto de valores.

Funciones de grupos de valores	
AVG(n)	Calcula el valor medio de "n" ignorando los valores nulos.
COUNT (* Expresión)	Cuenta el número de veces que la expresión evalúa algún dato con valor no nulo. La opción "*" cuenta todas las filas seleccionadas.
MAX (expresión)	Calcula el máximo.
MIN (expresión)	Calcula el mínimo.
SUM (expresión)	Obtiene la suma de los valores de la expresión.
GREATEST(valor1, valor2)	Obtiene el mayor valor de la lista.
LEAST (valor1, valor2...)	Obtiene el menor valor de la lista.
Funciones que devuelven valores de caracteres	

CHR(n)	Devuelve el carácter cuyo valor en binario es equivalente a "n".
CONCAT (cad1, cad2)	Devuelve "cad1" concatenada con "cad2".
LOWER (cad)	Devuelve la cadena "cad" en minúsculas.
UPPER (cad)	Devuelve la cadena "cad" en mayúsculas.
INITCAP (cad)	Convierte la cadena "cad" a tipo título.
LPAD (cad1, n[,cad2])	Añade caracteres a la izquierda de la cadena hasta que tiene una cierta longitud.
RPAD (cad1, n[,cad2]).	Añade caracteres a la derecha de la cadena hasta que tiene una cierta longitud
LTRIM (cad [,set])	Suprime un conjunto de caracteres a la izquierda de la cadena.
RTRIM (cad [,set])	Suprime un conjunto de caracteres a la derecha de la cadena.
REPLACE (cad, cadena_búsqueda [, cadena_sustitucion])	Sustituye un carácter o caracteres de una cadena con 0 o más caracteres.
SUBSTR (cad, m [,n])	Obtiene parte de una cadena.
TRANSLATE (cad1, cad2, cad3)	Convierte caracteres de una cadena en caracteres diferentes, según un plan de sustitución marcado por el usuario.

Funciones que devuelven valores numéricos	
ASCII(cad)	Devuelve el valor ASCII de la primera letra de la cadena "cad".
INSTR (cad1, cad2[, comienzo [,m]])	Permite una búsqueda de un conjunto de caracteres en una cadena pero no suprime ningún carácter después.
LENGTH (cad)	Devuelve el número de caracteres de cad.

Funciones para el manejo de fechas	
SYSDATE	Devuelve la fecha del sistema.
ADD_MONTHS (fecha, n)	Devuelve la fecha "fecha" incrementada en "n" meses.
LASTDAY (fecha)	Devuelve la fecha del último día del mes que contiene "fecha".
MONTHS_BETWEEN (fecha1, fecha2)	Devuelve la diferencia en meses entre las fechas "fecha1" y "fecha2".
NEXT_DAY (fecha, cad)	Devuelve la fecha del primer día de la semana indicado por "cad" después de la fecha indicada por "fecha".

Funciones de conversión	
TO_CHAR	Transforma un tipo DATE o NUMBER en una cadena de caracteres.
TO_DATE	Transforma un tipo NUMBER o CHAR en DATE.
TO_NUMBER	Transforma una cadena de caracteres en NUMBER.

CAPITULO 5

IMPLEMENTACION Y PRUEBAS

5.1 CONFIGURACION DE SERVIDOR DE APLICACIONES.

Para el funcionamiento de esta aplicación Instalaremos el Sistema Operativo Server 2003, aquí residirán los fuentes y ejecutables de la aplicación que se comunicaran con la Base de Datos en el intercambio de información.

Pasos para la Instalación de Developer Suite

- Insertamos el DVD de instalación abrimos en la ruta y procedemos a seleccionar el archivo de instalación, para nuestro caso se encuentra en:

D:\LabDB\DevSuite\developer_Suite\Disk1, como lo muestra la gráfica.

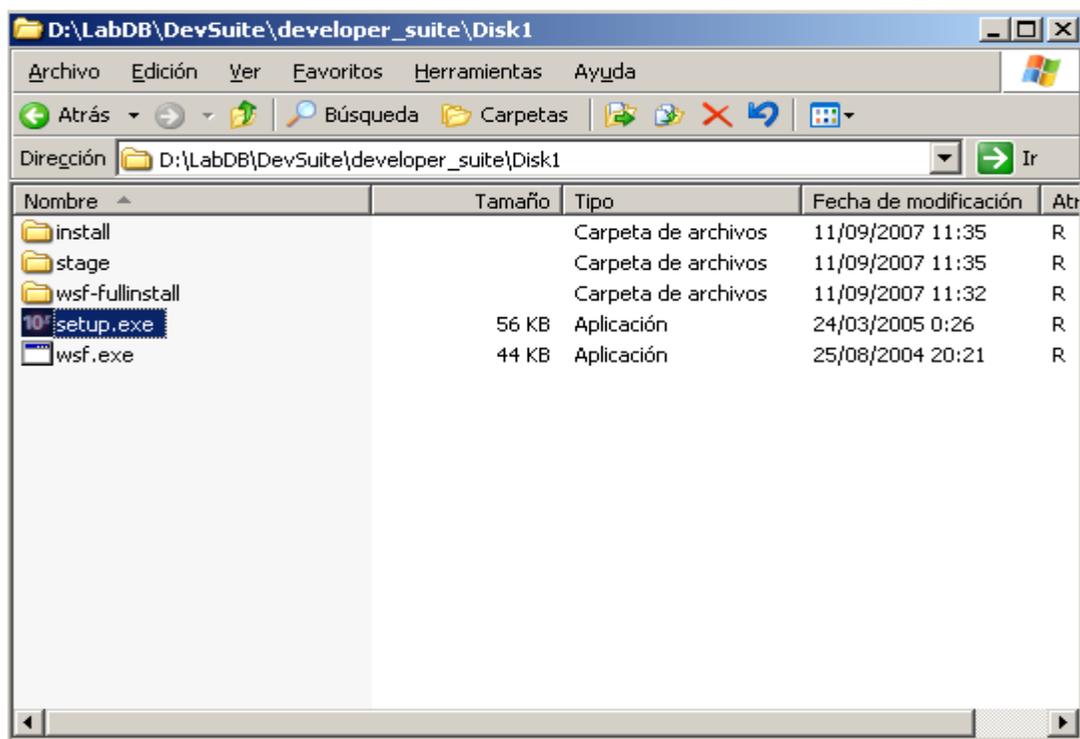


Figura 5.1 Instalación de Developer Suite (Inicio)

El asistente de instalación verifica los requisitos necesarios para la instalación como lo muestra la figura 5.2

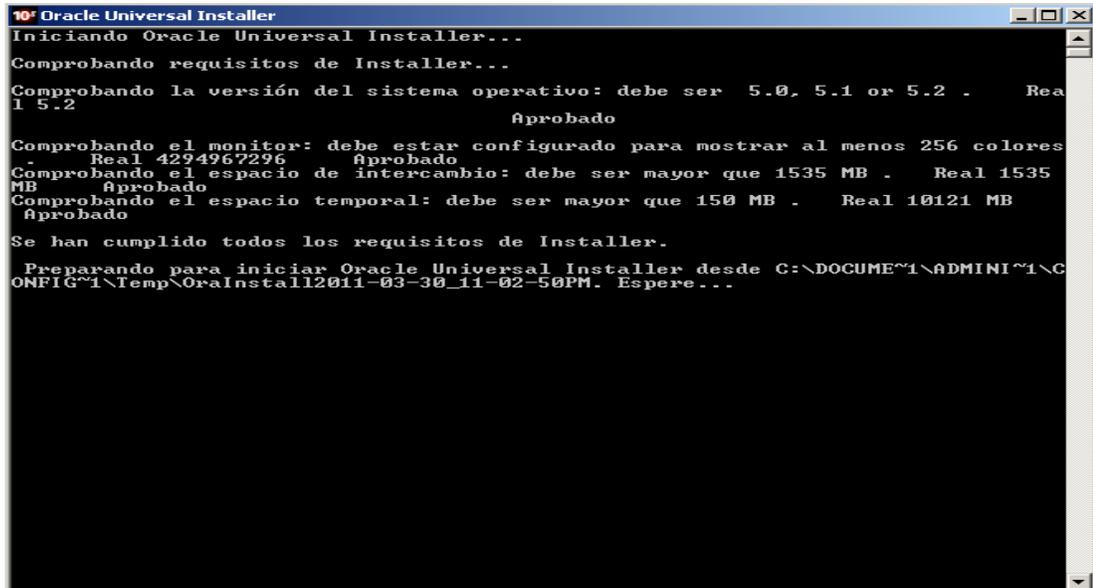


Figura 5.2 Comprobando Requisitos de Installer

Aparecerá esta ventana dar clic en siguiente

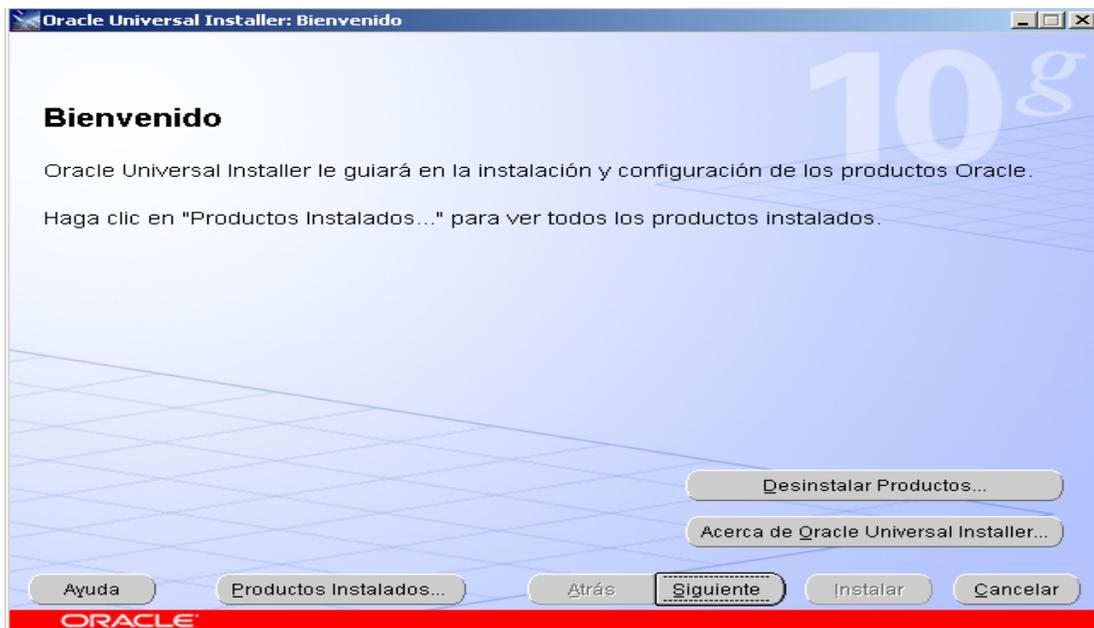


Figura 5.3 Asistente de Instalación.

Especificaremos la ubicación de los archivos

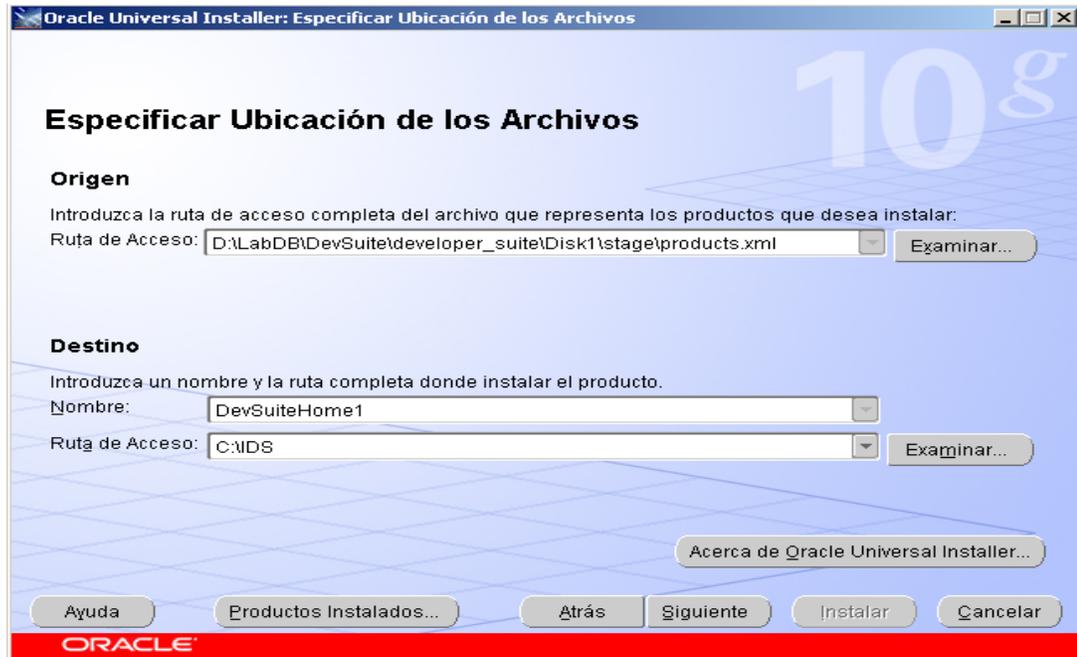


Figura 5.4 Especificar Ubicación de Archivos

En esta ventana seleccionamos la opción Terminar que contiene el Forms Developer, Report Builder y Oracle Designer, Podemos dar click en Idioma del Producto y seleccionar el de nuestra preferencia y damos click en siguiente.

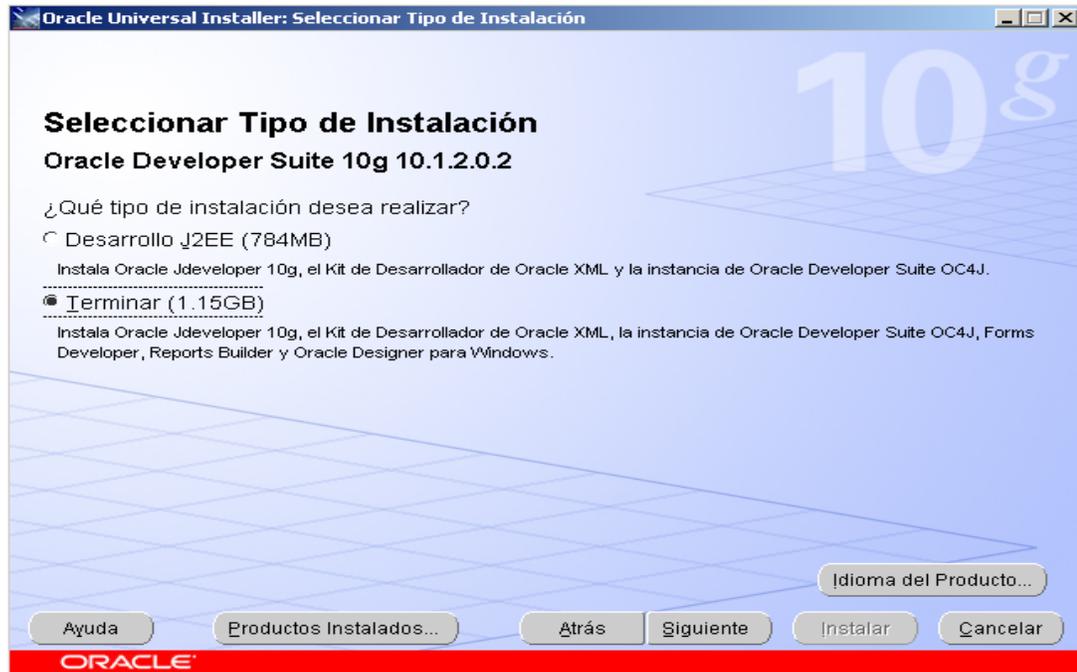


Figura 5.5 Seleccionar Tipo de Instalación

En esta ventana podremos ingresar una dirección de servidor de correo saliente para nuestro caso la dejamos en blanco y damos click en siguiente en donde aparecerá una barra indicando progreso de este paso

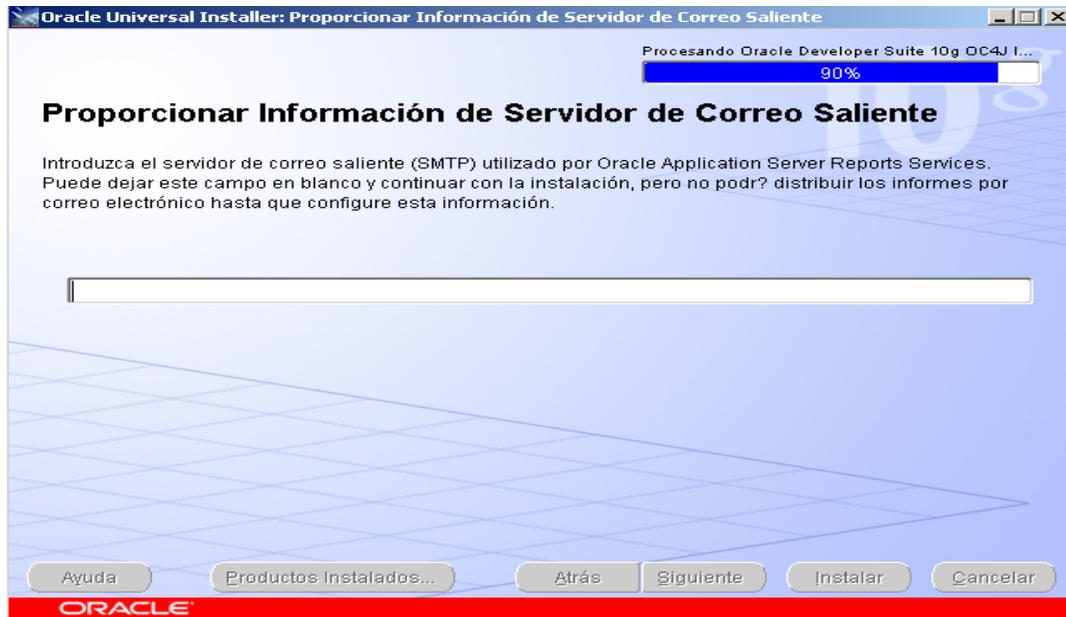


Figura 5.6 Información de Servidor de Correo Saliente

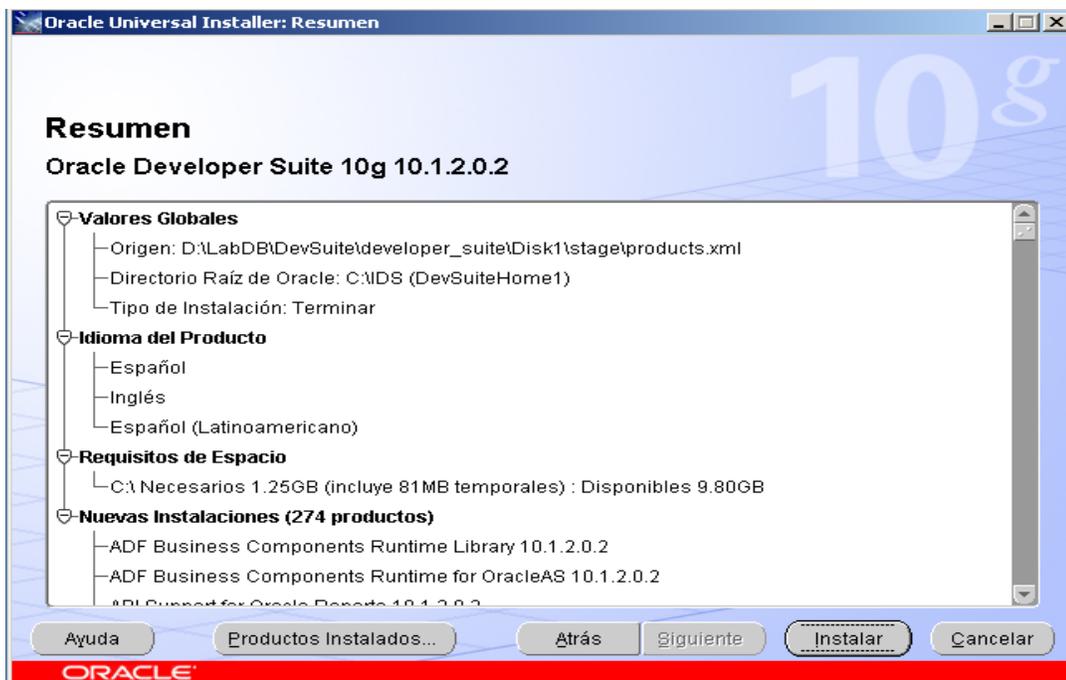


Figura 5.7 Resumen de Instalación

En esta ventana nos mostrará los productos que van a instalarse, dar click en Instalar

En esta ventana nos mostrará el progreso de la instalación de los productos en mención

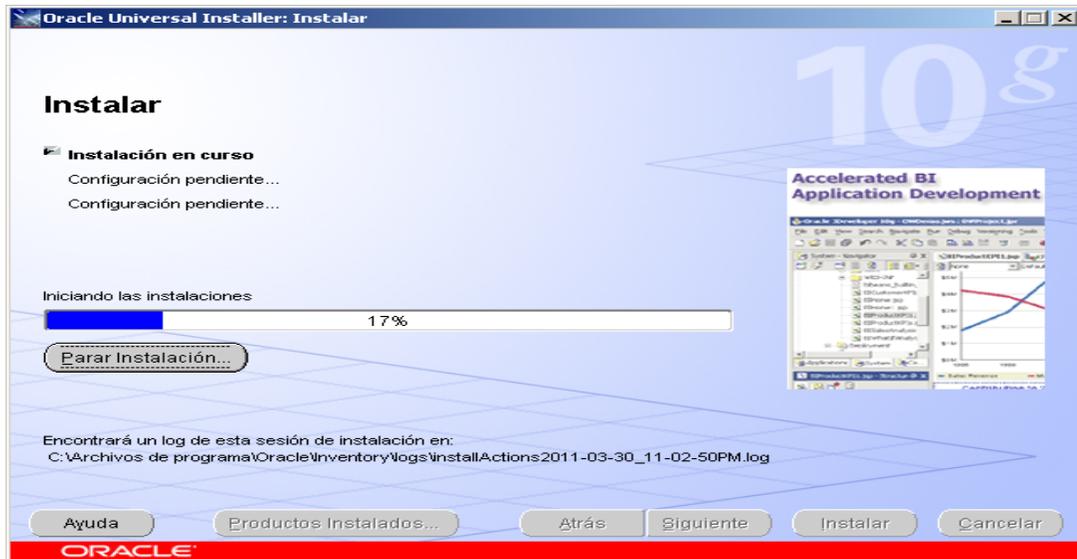


Figura 5.8 Progreso de la Instalación

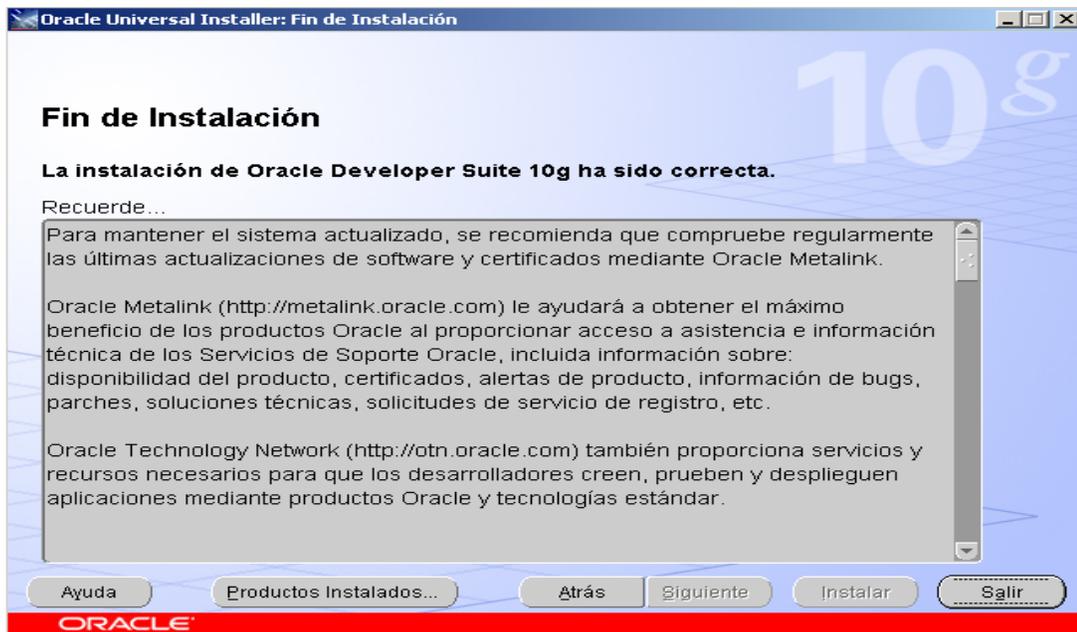


Figura 5.9 Fin de la Instalación

Esta ventana muestra el final de la instalación así como su estado para nuestro caso ha sido correcta. Procedemos a dar click en salir.

Como componente adicional necesario para el funcionamiento de nuestra aplicación instalamos el Plugin **JInitiator1.3.1.22**

Una vez instalado el Developer Suite en el Servidor, procedemos con la configuración de los archivos necesarios para el correcto funcionamiento de la aplicación.

Configuración de Archivos para la Aplicación.

- Copiar los siguientes archivos TNSNAMES.ora y SQLNET.ora de la siguiente ruta
C:\ora\product\10.2.0\db_1\NETWORK\ADMIN y pegarlos en
C:\IDS\NETWORK\ADMIN.
- La siguiente cadena de conexión que se encuentra en el TNSNAMES sirve para poder conectarse al servidor de Base de Datos.

```
ORACLE =  
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = DESARROLLO)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = ORACLE)  
    (SID = ORACLE)  
  )  
)
```

En donde **1521** es el Puerto de la Base de Datos.

Para ejecutar la aplicación es necesario crear los directorios donde residirán las aplicaciones esto es de la siguiente manera:

- C:\Tesis\FORMS
- C:\Tesis\Reports
- C:\Tesis\Librerias
- C:\Tesis\Documentos
- C:\Tesis\Iconos

Para poder ejecutar la aplicación de Oracle Forms 10g se deben dar los paths descritos a los siguientes directorios:

- Regedit.
- Default.env.

En el Regedit:

```
HKEY_LOCAL_MACHINE
  SOFTWARE
    ORACLE
      KEY_DevSuiteHome
        FORMS_PATH
```

C:\IDS\cgenf61\admin;C:\IDS\forms;C\Tesis\Forms;C\Tesis\Fmx;C:\Tesis\Librerias;
C:\Tesis\Documentos

```
HKEY_LOCAL_MACHINE
  SOFTWARE
    ORACLE
      KEY_DevSuiteHome
        REPORTS_PATH
```

C:\IDS\repadm61\srw;C:\IDS\cgenr61\admin\crretc;C:\IDS\cgenr61\admin;C:\IDS\reports\templates;C:\IDS\reports\samples\demo;C:\IDS\reports\integ;C:\IDS\reports\printers;
C:\Tesis\Reports

En el Default.env:

Abrir el archive Default.env que se encuentra en el directorio C:\IDS\forms\server\

C:\IDS\cgenf61\admin; **C:\IDS\forms;C:\Tesis\Forms; C:\Tesis\Fmx; C:\Tesis\Reports;
C:\Tesis\Librerias; C:\Tesis\Documentos;**

Nota: Se deben ingresar solo los valores en **Negrita**, pues los otros paths son propios de la instalación del Developer Suite.

Para que aparezcan los iconos de las formas generadas en ambiente web.

Verificar que el archivo en C:\IDS\forms\java\oracle\forms\registry\Registry.dat tenga el siguiente parámetro.

```
default.icons.iconpath=http://localhost/Iconos/
```

```
default.icons.iconextension=gif
```

Donde **http://localhost/Iconos/** es un servidor web con las imágenes de los botones de la aplicación.

Para nuestro servidor web con las imágenes que tenga nuestra aplicación instalaremos **Apache 2.0**

```
C:\Archivos de programa\Apache Group\Apache2\htdocs\Iconos
```

Concepto de WebUtil Oracle

WebUtil ofrece una manera sencilla de lograr la integración del lado del cliente durante la ejecución de las formas de Oracle en la Web.

Se simplifica la actualización de las aplicaciones existentes a la Web, y mejora las capacidades de la Web las aplicaciones basadas en Oracle Forms.

WebUtil tiene un objetivo primordial para producir una utilidad que cualquier desarrollador puede utilizar las formas para llevar a cabo las tareas complejas en las máquinas cliente del explorador simplemente codificación PL / SQL.

Para nuestro proyecto instalaremos y configuraremos WebUtil para aprovechar de las funciones que nos ofrece esta librería.

Instalación y Configuración de WebUtil

Tener la carpeta WEBUTIL en la unidad C:\

- Copiar las 4 Librerías de la Carpeta

```
\WEBUTIL\DLL\ d2kwut60.dll  
\WEBUTIL\DLL\ ffisamp.dll  
\WEBUTIL\DLL\ jacob.dll  
\WEBUTIL\DLL\ JNIsharedstubs.dll
```

A

```
C:\IDS\DevSuiteHome\BIN\
```

- Copiar la Librería

```
\WEBUTIL\ jacob.jar
```

A

```
C:\IDS\DevSuiteHome\Forms\Java
```

- Copiar la Librería

```
\WEBUTIL\DLL\ JACOB.DLL
```

A

```
C:\IDS\DevSuiteHome\forms\webutil\
```

- Abrir el Archivo
C:\IDS\DevSuiteHome\Forms\Server**formsweb.cfg**
y copiar lo que está al final del archivo

```
WebUtilArchive=frmwebutil.jar,jacob.jar  
WebUtilLogging=off  
WebUtilLoggingDetail=normal  
WebUtilErrorMode=Alert  
WebUtilDispatchMonitorInterval=5  
WebUtilTrustInternal=true  
WebUtilMaxTransferSize=16384  
baseHTMLjiniator=webutiljini.htm  
baseHTMLjpi=webutiljpi.htm
```

Ir al inicio del archivo y a partir del DEFAULT pego lo copiado

- Luego las siguientes líneas que están a partir de # System parameter: default base HTML file las comento y pongo los nuevos valores:

```
# System parameter: default base HTML file
#baseHTML=base.htm
baseHTML=webutilbase.htm
# System parameter: base HTML file for use with JInitiator client
#baseHTMLjinitiator=basejini.htm
baseHTMLjinitiator=webutiljini.htm
# System parameter: base HTML file for use with Sun's Java Plug-In
#baseHTMLjpi=basejpi.htm
baseHTMLjpi=webutiljpi.htm
```

- Para probar las configuraciones probamos la forma
C:\Webutil\Webutil_demo\WU_TEST_106.fmb para ver si esta correcto

Si al momento de probar sale un mensaje que prohíbe el ingreso de las imágenes

Editar el archivo.

C:\IDS\DevSuiteHome\forms\server**webutil.cfg** y al final del archivo encontramos

```
#NOTE: By default the file transfer is disabled as a security measure
transfer.database.enabled=FALSE
transfer.appsrv.enabled=FALSE
transfer.appsrv.workAreaRoot=
transfer.appsrv.accessControl=TRUE
#List transfer.appsrv.read.<n> directories
transfer.appsrv.read.1=c:\temp
#List transfer.appsrv.write.<n> directories
transfer.appsrv.write.1=c:\temp
```

Se debe cambiar por True los valores False

```
transfer.database.enabled=TRUE
transfer.appsrv.enabled=TRUE
transfer.appsrv.workAreaRoot=
transfer.appsrv.accessControl=TRUE
#List transfer.appsrv.read.<n> directories
transfer.appsrv.read.1=c:\temp
#List transfer.appsrv.write.<n> directories
transfer.appsrv.write.1=c:\temp
```

Datos Adicionales

Para heredar el **Object_Group** de la forma C:\Webutil\Webutil_demo\WU_TEST_106.fmb a una forma mía.

Arrastro hasta el **Object_Group** de la forma mencionada a la mía.

Luego agrego ATACHED LIBRARIES Y SELECCIONO EL ARCHIVO
C:\WEBUTIL\ webutil.pll

Si nos pregunta Remove Path Ponemos **No**

Copiamos el código de la forma ejemplo a la nuestra.

Para que no se pierdan las librerías y las clases

- En el Regedit en FORMS_PATH agrego mis paths

FORMS_PATH = C:\Webutil\Webutil_demo

- En el FORMS_PATH DEL ARCHIVO

C:\IDS\DevSuiteHome\forms\server\Default.env

Además agrego mis paths

FORMS_PATH= C:\Webutil\Webutil_demo

5.2 Instalación de la Aplicación

Para la instalación de la aplicación en los equipos que van a ser parte de la solución debemos considerar y hacer lo siguiente:

- Instalar JInitiator 1.3.1.22 en cada equipo.
- Cargar los Componentes del Servidor de Aplicaciones al Equipo, ejecutando esta dirección: **<http://desarrollo:8889/forms/frmservlet/config=aplicacion>**

5.3 Pruebas de la Aplicación.

Las pruebas de software, en inglés **testing** son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador.

El **testing** puede probar la presencia de errores pero no la ausencia de ellos

La importancia de una detección oportuna.

En la cadena de valor del desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Las aplicaciones de software han crecido en complejidad y tamaño, y por consiguiente también en costos. Hoy en día es crucial verificar y evaluar la calidad de lo construido de modo de minimizar el costo de su reparación. Mientras más pronto se detecte su falla más barato será el costo de su corrección.

5.3.1 Tipos de Pruebas

Dentro de los tipos de pruebas existentes para el Software tenemos:

- Pruebas Unitarias.
- Pruebas Funcionales
- Pruebas de Integración
- Pruebas de Carga
- Pruebas de Recorrido
- Caja Blanca
- Caja Negra

Para la comprobación de nuestro software se ha utilizado Pruebas Unitarias y Pruebas de Integración.

Pruebas Unitarias

Es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego con las Pruebas de Integración se podrá asegurar el correcto funcionamiento del sistema en cuestión.

Características

Para que una Prueba Unitaria sea buena debe cumplir con las siguientes características:

- **Automatizable:** no debería requerirse una intervención manual. Esto es especialmente útil para integración continua.
- **Completas:** deben cubrir la mayor cantidad de código.
- **Repetibles o Reutilizables:** no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. También es útil para integración continua.
- **Independientes:** la ejecución de una prueba no debe afectar a la ejecución de otra.
- **Profesionales:** las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.

Ventajas

- **Fomentan el cambio:** Las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura (lo que se ha dado en llamar refactorización), puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.
- **Simplifica la integración:** Puesto que permiten llegar a la fase de integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.
- **Documenta el código:** Las propias pruebas son documentación del código puesto que ahí se puede ver cómo utilizarlo.

Pruebas de Integración

Son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez.

Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos.

En base a lo acotado consideramos que:

Las pruebas de la aplicación se realizaron según el continuo desarrollo del sistema probando cada uno de los módulos con información real.

Los datos ingresados para cada uno de los programas tienen consistencia en la Base de datos.

La integración de los módulos se las realiza mediante un menú codificable el mismo que está adaptado con lo necesario para cada usuario.

5.4 Corrección de los errores encontrados

La mayoría de los errores detectados han sido problemas con la definición de los atributos en cuanto a los tipos y anchos de cada uno de ellos.

En la instalación de la aplicación se ha visto que el Gestor de Base de Datos, así como los programas accedidos responden de manera muy adecuada, constatando la funcionalidad del Sistema de Control de No Conformidades.

5.5 Sistema en Producción.

Luego de las correcciones y pruebas realizadas se determinó que la herramienta ha funcionado correctamente y con ello el desarrollo de aplicación en sus capas de Diseño, Prototipo y Producción.

CAPITULO 6 DOCUMENTACION

6.1 Manual Técnico

Luego de haber comprobado que los equipos que forman parte de la aplicación, tengan los requerimientos necesarios especificados y la red de datos este en pleno funcionamiento procedemos a ingresar a la aplicación

6.2 Manual de Usuario

6.2.1 Gestión de No Conformidades.

El manual de usuario se rige a cada uno de los módulos, en este capítulo y de manera general explicare cómo funcionan los mantenimientos, cabe recalcar que estas pantallas están diseñadas de tal manera que son comprendidas por el usuario que maneja formularios manualmente.

Plantearemos como ejemplo el modulo de Gestión de No Conformidades desde el registro hasta el tratamiento para el cierre de una No Conformidad.

Al ingresar la Opción **Registrar Expediente No Conformidad**, es aquí donde el usuario que tiene un perfil de auditor es capaz de ingresar toda la información correspondiente a una No Conformidad encontrada en un área de la institución de la siguiente manera:

1. Ingresar el Nombre o Título del Expediente.
2. Asociar a un Trámite de la Institución.
3. El sistema rellenara los campos Tipo, Tipo Expediente, Fecha, Fecha de Expiración, Remitente esta información esta codificada en el Mantenimiento de Trámites.
4. Se debe ingresar el campo Asunto tomando los valores de un ComboList.
5. Ingresar Area, Proceso y Estado tomando los valores de un ComboList
6. Ingresar el Análisis de Causa.
7. Ingresar los Requisitos y la observación de porque se incumple.
8. Ingresar las Acciones Correctivas o Preventivas, que se impartan a la No Conformidad encontrada.

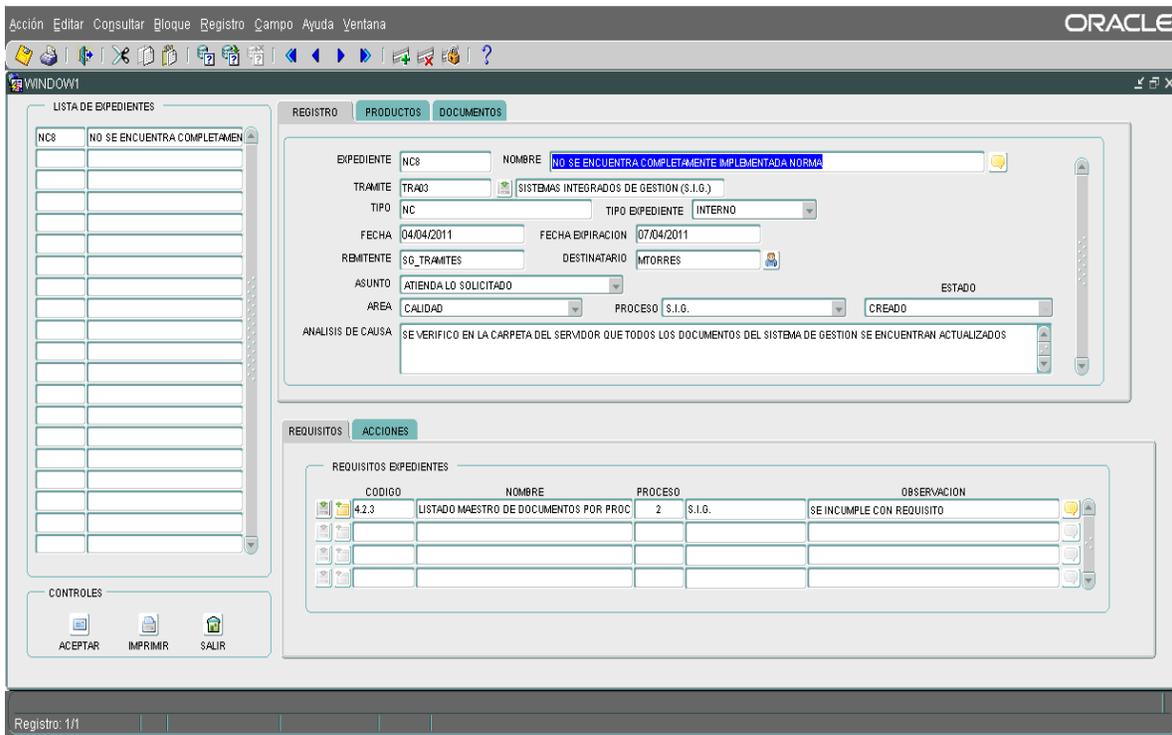


Figura 6.1 Registrando una No Conformidad

Una vez registrada la No Conformidad, esta es almacena en la dentro de la opción Correspondencia en la Bandeja de **Creados y No Enviados**

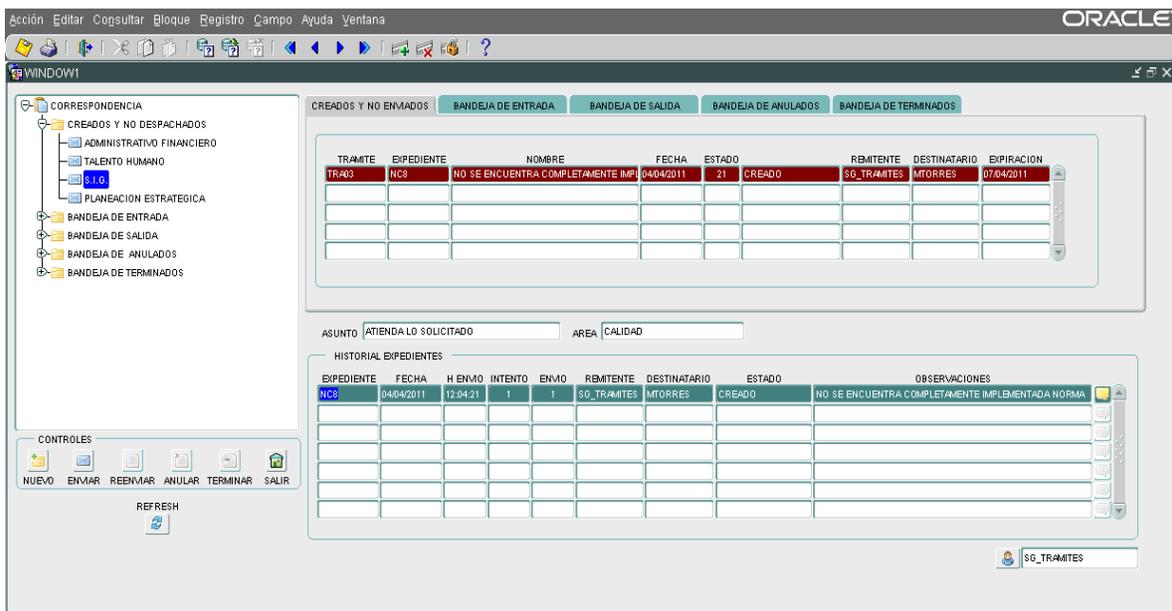


Figura 6.2 Repositorio de Expedientes Creados y No Enviados

Si estamos seguros de la creación de este Expediente procedemos a enviarlo a su destinatario, asociado.

El expediente recientemente enviado se almacena en la bandeja de entrada del destinatario en mención y este una vez resuelta la no conformidad reenviarla para su cierre

REGISTRO PRODUCTOS DOCUMENTOS

EXPEDIENTE: NC8 NOMBRE: NO SE ENCUENTRA COMPLETAMENTE IMPLEMENTADA NORMA

TRAMITE: TRA03 SISTEMAS INTEGRADOS DE GESTION (S.I.G.)

TIPO: NC

TIPO EXPEDIENTE: INTERNO FECHA: 04/04/2011 FECHA EXPIRACION: 07/04/2011

REMITENTE: SG_TRAMITES DESTINATARIO: MITORRES

ASUNTO: ATIENDA LO SOLICITADO AREA: CALIDAD

ESTADO: ENVIADO

ANALISIS DE CAUSA: SE VERIFICO EN LA CARPETA DEL SERVIDOR QUE TODOS LOS DOCUMENTOS DEL SISTEMA DE GESTION SE ENCUENTRAN ACTUALIZADOS

ANEXO

REMITENTE: MITORRES

DESTINATARIO: SG_TRAMITES

OBSERVACION: SE VERIFICO EN LA CARPETA DEL SERVIDOR QUE TODOS LOS DOCUMENTOS DEL SISTEMA DE GESTION SE ENCUENTRAN ACTUALIZADOS

REQUISITOS ACCIONES

NRO	TRAMITE	CODIGO	NOMBRE	DESCRIPCION
1	TRA03	10	ACCIONES CORRECTIVAS	IMPRESION Y FIRMA DE LOS DOCUMENTOS APROBADOS SUBIDOS EN E
2	TRA03	10	ACCIONES CORRECTIVAS	COMPRA DE CARPETAS PARA LA DISTRIBUCION DE LAS COPIAS CONT

REENVIAR SALIR

Figura 6.3 Leer Reenviar para su revisión

Cuando está en esta opción el usuario es capaz de revisar las pestañas como productos encontrados en la No Conformidad o un documento asociado al momento de registrarla.

REGISTRO PRODUCTOS DOCUMENTOS

EXPEDIENTE: NC8 NOMBRE: NO SE ENCUENTRA COMPLETAMENTE IMPLEMENTADA NORMA

TRAMITE: TRA03 SISTEMAS INTEGRADOS DE GESTION (S.I.G.)

TIPO: NC

TIPO EXPEDIENTE: INTERNO FECHA: 04/04/2011 FECHA EXPIRACION: 07/04/2011

REMITENTE: MITORRES DESTINATARIO: SG_TRAMITES

ASUNTO: ATIENDA LO SOLICITADO AREA: CALIDAD

ESTADO: LEIDO Y REENVIADO

ANALISIS DE CAUSA: SE VERIFICO EN LA CARPETA DEL SERVIDOR QUE TODOS LOS DOCUMENTOS DEL SISTEMA DE GESTION SE ENCUENTRAN ACTUALIZADOS

OBSERVACION: SE VERIFICO Y TODO ESTA EN CORRECTO ORDEN

EFICACIA_CUMPLIMIENTO

TODOS LOS DOCUMENTOS DEL SISTEMA DE GESTION SE ENCUENTRAN ACTUALIZADOS

PORCENTAJE %: 100.00

REQUISITOS ACCIONES

NRO	TRAMITE	CODIGO	NOMBRE	DESCRIPCION
1	TRA03	10	ACCIONES CORRECTIVAS	IMPRESION Y FIRMA DE LOS DOCUMENTOS APROBADOS SUBIDOS EN E
2	TRA03	10	ACCIONES CORRECTIVAS	COMPRA DE CARPETAS PARA LA DISTRIBUCION DE LAS COPIAS COI

CONTROLES

TERMINAR SALIR

Figura 6.4 Terminando (Cerrando No Conformidad)

Re planificar una No Conformidad

Una No Conformidad solo se puede Replanificar, cuando esta haya caducado el sistema controla automáticamente este proceso de caducidad.

CODIGO	NOMBRE	PROCESO	S.I.G.	OBSERVACION
8.2.3	SEGUIMIENTO A LOS PROCESOS	2	S.I.G.	SE INCUMPLE CON REQUISITO
8.2.2	DEFINICION DE INDICADORES DE	2	S.I.G.	SE INCUMPLE CON REQUISITO

Figura 6.5 Replanificar No Conformidad

Anular una No Conformidad

En este caso solo el administrador del sistema puede anular una No Conformidad

TRAMITE	EXPEDIENTE	NOMBRE	FECHA	ESTADO	NOMBRE	TIPO	REMITENTE	DESTINATARIO	EXPIRACION	OBSERVACION
TRA03	NC2	PROGRAMA DE AUDITORIA SE ENCUENTR	04/04/2011	27	REPLANIFICADO	INTERNO	MTORRES	SG_TRAMITES	07/04/2011	SE REPLANIFICA POR PETICION DE GINNA

Figura 6.6 Anular No Conformidad

6.2.2 Indicadores de Gestión de No Conformidad

6.2.2.1 Historial de No Conformidades

En esta pantalla se pueden consultar de manera histórica todas las No Conformidades y sus diferentes estados durante la gestión.

TIPOS TRAMITES: NO CONFORMIDAD TRAMITES: SISTEMAS INTEGRADOS DE GESTION (S.I.G.) EXPEDIENTES: NC8 CONSULTAR

HISTORIAL_EXPEDIENTES

HISTORIAL DE EXPEDIENTES

EXPEDIENTE	N ENVIADO	N INTENTO	REMITENTE	DESTINATARIO	FECHA	H ENVIADO	CREADO	ENVIADO	REENVIADO	CADUCADO	REPLANIFICADO	TERMINADO	ANULADO	E. ACTUAL
NC8	2	1	SG_TRAMITES	MTORRES	04/04/2011	12:04:04		X	X					ENVIADO
NC8	1	1	SG_TRAMITES	MTORRES	04/04/2011	12:04:21	X							CREADO

OBSERVACIONES: SE VERIFICO EN LA CARPETA DEL SERVIDOR QUE TODOS LOS DOCUMENTOS DEL SISTEMA DE GESTION SE ENCUENTRAN ACTUALIZADOS.

ANALISIS_CAUSA: SE VERIFICO EN LA CARPETA DEL SERVIDOR QUE TODOS LOS DOCUMENTOS DEL SISTEMA DE GESTION SE ENCUENTRAN ACTUALIZADOS.

IMPRIMIR SALIR

Figura 6.7 Historial de No Conformidades

6.2.2.2 Consulta de No Conformidades

En esta pantalla el usuario puede consultar cada una de las No Conformidades ingresadas en el sistema, así como sus requisitos no cumplidos y sus acciones correctivas asignadas.

LISTA

- NC1 NO SE ENCUENTRA COMPLETAMEN
- NC2 PROGRAMA DE AUDITORIA SE ENC
- NC3 NO SE ENCONTRARON LOS RESUL
- NC4 NO SE ENCONTRARON REGISTROS
- NC5 EXISTE INCONGRUENCIA ENTRE LC
- NC6 EN EL LISTADO MAESTRO DE EQUIP
- NC7 PROBEA
- NC8 NO SE ENCUENTRA COMPLETAMEN

REGISTRO PRODUCTOS DOCUMENTOS

EXPEDIENTE: NC3 NOMBRE: NO SE ENCONTRARON LOS RESULTADOS ACTUALIZADOS DEL CUADRO DEMANDO

TRAMITE: TRAD3 SISTEMAS INTEGRADOS DE GESTION (S.I.G.)

TIPO: NC TIPO EXPEDIENTE: INTERNO

FECHA: 12/02/2011 FECHA EXPIRACION: 15/02/2011

REMITENTE: SG_TRAMITES DESTINATARIO: MTORRES

ASUNTO: CONOZCA Y RESUELVA ESTADO: CADUCADO

ANALISIS DE CAUSA: FALTA DE DEFINICIÓN DE LOS INDICADORES DE GESTIÓN POR PROCESOS
NO SE HAN COMUNICADO LOS INDICADORES EXISTENTES DE MANERA EFECTIVA EN LAS ÁREAS RESPONSABLES
FALTA DE CAPACITACIÓN DEL PERSONAL EN LOS TEMAS DE ANÁLISIS DE DATOS Y SEGUIMIENTO Y MEDICIÓN DE PROCESOS

REQUISITOS ACCIONES

REQUISITOS EXPEDIENTES

CODIGO	NOMBRE	PROCESO	OBSERVACION
8.2.3	SEGUIMIENTO A LOS PROCESOS	2 S.I.G.	SE INCUMPLE CON REQUISITO
8.2.2	DEFINICION DE INDICADORES DE CAPACIDAD D	2 S.I.G.	SE INCUMPLE CON REQUISITO

ACEPTAR IMPRIMIR SALIR

Figura 6.8 Consulta de No Conformidades

CONCLUSIONES Y RECOMENDACIONES

La implementación de esta solución ha permitido llevar un verdadero control, de las incidencias detectadas, mejorando los tiempos de respuesta al momento de registrar una No Conformidad, así como del tratamiento de esta, desde su inicio hasta el cierre, cambiando el paradigma de antiguos sistemas de registro de No Conformidades mediante papel o archivos informáticos (Word, Excel).

La notificación de una no conformidad al responsable de darle solución es inmediata, reduciendo considerablemente el tiempo y los recursos para su gestión y asignación de acciones correctivas o preventivas.

La posibilidad de adjuntar evidencias objetivas de apertura o cierre de las no conformidades y de disponer de la información en tiempo real ha hecho que esta herramienta sea parte de una solución en la institución

Las herramientas Oracle han revolucionado la forma de automatizar los aspectos claves en el desarrollo de los sistemas de información. Oracle brinda toda una gama completa e integrada de componentes y herramientas para el desarrollo de aplicaciones, desarrollo de base de datos e inteligencia de negocios, compatibles con cualquier enfoque de desarrollo, plataforma tecnológica o sistema operativo.

Oracle Developer Suite es un ambiente de desarrollo completo e integrado que combina el poder, la rapidez y exactitud para el desarrollo de aplicaciones con las herramientas de inteligencia de negocio en una sola suite.

Desde que se crearon estas herramientas hasta la actualidad cuentan con una credibilidad y exactitud generando un reconocimiento universal, siendo usadas por cualquier tipo de desarrollador y/o programador que busca un resultado óptimo y eficiente, pero sobre todo que busca esa minuciosidad necesaria de los procesos entre los procesos.

Los objetivos planteados se han cumplido, el tiempo de respuesta para registrar e acceder a la información han mejorado notablemente por tener la información actualizada en tiempo real.

Otro de los aspectos rescatables es la generación de los informes necesarios y presentados por el departamento de S.I.G.(Sistemas Integrados de Gestión), anteriormente al realizarlos en forma manual, su tiempo de entrega tenía muy grandes retrasos.

Los ERS representan eficiencia y un alto grado de efectividad para recopilación de los requisitos del usuario, con lo cual disminuye los cambios en fases posteriores e involucra la coordinación entre técnicos y usuarios.

Una de las recomendaciones principales es poner mucho más énfasis en el mantenimiento de la información, ya que es la parte fundamental del sistema, sus respaldos deben hacerse en periodos no muy extensos.

Adicionalmente, como parte de la evolución de esta herramienta, se ha considerando el diseño y creación de módulos que ayuden a controlar y registrar el análisis del papel por producto, y puedan estos emitir las certificados calidad para cada producto en la orden de un cliente, integrándolos con el Sistema de Control de No Conformidades.

BIBLIOGRAFIA

KENDALL Y KENDALL; “Análisis y Diseño de Sistemas”; Pearson Educación (Agosto 2005); 6ta Edición.

STALLINGS, William; “ Sistemas Operativos ”; HALL Prentice; (Octubre 2009) 4ta Edition.

SILBERSCHATZ, KORT, Sudarshan; “Fundamentos de Base De Datos”; Editorial Mc Graw Hill; 4ta Edición.

DORSEY, Paul; KOLETZKE, Peter, “Manual Avanzado de Oracle Forms y Reports”; McGraw-Hill / Interamericana de España, S.A; Edición Publicada Oficial Oracle Press; (Septiembre 2009).

COSTA, Estany; Josep, Maria; “Norma ISO 9001” ; Editorial Ciss, S.A. (Marzo 2008).

VELASCO SÁNCHEZ, Juan, “Gestión de la calidad, mejora continua y sistemas de gestión”; Ediciones Pirámide, S.A.

ORACLE, “Aplication Server”, <https://support.oracle.com/CSP/ui/flash.html>
[Consulta Noviembre 2010]

OSBORNE ORACLE PRESS, “WebUtil”, <http://www.osborne.com>,
[Consulta, Noviembre 2010]

WIKIPEDIA ENCICLOPEDIA LIBRE, “No Conformidades”,
<http://es.wikipedia.org/wiki/Conformidad>, [Consulta, Diciembre 2010]

COMMITTEE ISO 9001 AUDITING PRACTICES GROUP, “Registrando una No
Conformidad”, <http://www.iso.org/tc176/ISO9001AuditingPracticesGroup> [Consulta
Diciembre, 2010]

	REPORTE Y ANALISIS DE NO CONFORMIDADES Y PLANTEAMIENTO DE ACCIONES CORRECTIVAS, PREVENTIVAS Y DE MEJORA	Código: FR.SIG.04
		Revisión. No. 02 Página 1 de 1

ÁREA/DPTO/PROCESO: PROGRAMACION ESTRATEGICA.

No.

FUENTE DE DETECCIÓN DE LA NO CONFORMIDAD		ORIGEN DE LA NO CONFORMIDAD	
Auditoria:	X	Materia prima o insumo:	
		Producto en proceso:	
Reclamo o queja:		Producto terminado:	
		Impacto ambiental	
Inspección de rutina:		Riesgo o peligro de seguridad y salud ocupacional:	
Comunicación de partes interesadas externas (entes gubernamentales, clientes, proveedores, comunidad, entre otros):		Incidentes:	
Otro:		Emergencias:	
		Otro: Incumplimiento de norma	X

Consecutivo: 05

ISO 9001:2008

ISO 14001: 2004

OHSAS 18001:2007

DESCRIPCION DEL PROBLEMA / PROBLEMA POTENCIAL:		
No existen registros de la remisión periódica a la Dirección, incumpliendo de esta forma la Norma en su numeral 8.5.2, literal e) que dice la Organización debe registrar los resultados de las acciones tomadas.		
Fecha Reporte: Martes, 06 de Julio del 2010	Reportado Por: Franklin Egas.	Responsable de toma de Acciones: Responsable del SIG.

ANALISIS DE CAUSAS:	
Se incumple con este procedimiento debido, a que quizás todas las acciones y decisiones sobre los análisis de las No Conformidades lo manejaba directamente la representante del SIG, y no se informaba a la Dirección sobre las acciones y decisiones tomadas sobre el análisis de las No Conformidades.	
Responsable: Ing. Gina Motta.	Fecha: 06 de Julio del 2010.

PLAN DE ACCIÓN

Corrección (soluciones inmediata) <input checked="" type="checkbox"/>	Responsable	Fecha de Cumplimiento
Llevar registro de las remisiones a la Dirección, sobre todas las acciones correctivas aplicadas.	Ing. Gina Motta	Próxima Auditoría

Acción Correctiva <input type="checkbox"/>	Acción Preventiva <input type="checkbox"/>	Acción de Mejora <input type="checkbox"/>	Responsable	Fecha de Cumplimiento
Practicar el hábito de remitir a la Dirección toda información sobre las acciones correctivas y preventivas tomadas en una Auditoría.			Ing. Gina Motta.	Inmediatamente.
				0

Fecha programada de seguimiento	Próxima Auditoría.
Responsable del seguimiento	Anita Romero.

	REPORTE Y ANALISIS DE NO CONFORMIDADES Y PLANTEAMIENTO DE ACCIONES CORRECTIVAS, PREVENTIVAS Y DE MEJORA	Código: FR.SIG.04
		Versión. No. 02 Página 1 de 1

ÁREA/DPTO/PROCESO: **COMERCIALIZACION**

No. Consecutivo: 1

FUENTE DE DETECCIÓN DE LA NO CONFORMIDAD		ORIGEN DE LA NO CONFORMIDAD	
Auditoria externa:		Materia prima o insumo:	
		Producto en proceso:	
Reclamo o queja:		Producto terminado:	
		Impacto ambiental	
Inspección de rutina:		Riesgo o peligro de seguridad y salud ocupacional:	
Comunicación de partes interesadas externas (entes gubernamentales, clientes, proveedores, comunidad, entre otros):		Incidentes:	
Otro: AUDITORIA INTERNA.		Emergencias:	
		Otro: DOCUMENTACIÓN	

ISO 9001:2008

ISO 14001: 2004

OHSAS 18001:2007

DESCRIPCION DEL PROBLEMA / PROBLEMA POTENCIAL:		
Según el requisito 4.2 de la NORMA ISO 9001 Requisitos de la documentación, en el PROCESO DE COMERCIALIZACION-DISTRIBUCION, ITEM 4.2.3 Control de Documentos existe un incumplimiento total en cuanto a la revisión y actualización de los mismos. PR.CD.02		
Fecha Reporte: 05 de julio del 2010	Reportado Por: Alex Orozco	Responsable de toma de Acciones: Paul Ochoa.

ANALISIS DE CAUSAS:	
Responsable:	Fecha:

PLAN DE ACCIÓN

Corrección (soluciones inmediata)	Responsable	Fecha de Cumplimiento

Acción Correctiva <input checked="" type="checkbox"/>	Acción Preventiva <input type="checkbox"/>	Acción de Mejora <input type="checkbox"/>	Responsable	Fecha de Cumplimiento

Fecha programada de seguimiento	
Responsable del seguimiento	

**DISEÑO DE TESIS
PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SISTEMAS**

PERTENECE A

JACKSON MARCELO SERRANO TINITANA

1. TEMA DE TESIS.

Sistema de Control de No-Conformidades aplicado a la **Industria Cartonera INCARPALM SA**

2. PRESENTACION DEL PROYECTO.

El proyecto utilizara la Arquitectura Tres Capas con sus niveles lógicos

- Capa Cliente (Aplicaciones).
- Capa Servidor de Aplicaciones.
- Capa Servidor de Datos (Base de Datos).

Con el uso de Forms y Reports de la Herramienta Oracle obtendremos la automatización y control de los procesos.

Este sistema realizara las siguientes funciones:

- Permitir la automatización, control y monitoreo de los procesos de producción y procesos administrativos.
- Seguimiento y registro de las acciones propias de las distintas etapas de los procesos así como sus respectivas acciones correctivas y preventivas.
- Diseño de planes de acción, implementación y verificación de efectividad, permite definir el responsable y plazos de ejecución.
- Determinar el nivel de satisfacción de dicha acción si fue efectiva o no.
- Reducir los tiempos de respuesta y acción al momento de encontrar una No-Conformidad.
- Correspondencia de No-Conformidades enviadas y recibidas.
- Estadísticas de No-Conformidades.
- Registrar quien(es) tiene la autoridad para levantar las No-Conformidades se recomendaría que fuera el Representante de la Gerencia de SIG, que es dentro de la organización quien más conoce la norma, sus requisitos e implicancias.

Si existe una evidencia o suceso esta debe ser documentada como no conformidad en vez de ser suavizada con otras definiciones como por ejemplo: Observaciones u Oportunidad de Mejora.

Con el desarrollo del sistema se permitirá al usuario poder registrar e identificar el problema y el área de una No-Conformidad, se podrá asignar las respectivas acciones correctivas y preventivas a la evidencia levantada, una vez dado esto se podrá , hacer un seguimiento a la evidencia hasta cerrar la No-Conformidad.

El proyecto debe considerar algunos aspectos:

- Crear una Base de Datos en la SGA (System Global Area) de Oracle.
- Los Formularios o Modelos a Crear serán de Formato Web.
- Los Reportes emitidos serán en formato Pdf.
- El acceso a los datos será mediante la utilización de Base de Datos relacional.
- Establecer Log's de auditoría para controlar el acceso al sistema, para establecer seguridades.

3. JUSTIFICACION E IMPACTOS

La investigación que se va a realizar permitirá al desarrollador la implementación de conocimientos sobre la creación de aplicaciones Web con un servidor de Base de Datos Oracle.

La necesidad de la Industria Cartonera INCARPALM, es poder controlar las no-conformidades o evidencias que se presentan en las diferentes etapas de la producción, y darles las consecuentes acciones correctivas, para poder así mejorar la calidad de sus productos, reducir y optimizar costos, agilizar el proceso de producción y brindar un producto de calidad al cliente, es por esta razón que se amerita poner en marcha el presente proyecto justificando las mejoras y proyectos a futuro que se puedan realizar al buscar mejores modelos de adaptación.

Usando este tipo de tecnología se garantizará la seguridad de los datos, el manejo y entrega de la información en tiempo real, ya que esta es de mucha importancia para la institución.

IMPACTOS

IMPACTO TECNOLÓGICO

La tecnología hoy en día permite actualmente el control de productos de una forma tan rápida, que es no es considerable cualquier descuido que permita la aparición de defectos que puedan afectar a una producción y más aun no es considerable que no puedan registrarse y darles el respectivo tratamiento a estas anomalías de una manera dinámica.

El control de No-Conformidades y su seguimiento es tecnológicamente indispensable en instituciones que tienen como objetivo establecer los lineamientos para la identificación y tratamiento de los servicios no conformes aplicables a los procesos definidos en el alcance del Sistema de Gestión de Calidad. Actualmente la información manipulada por la Industria Cartonera INCARPALM S.A. está basada en Hojas de Cálculo y Papel llenados a mano con niveles de seguridad muy bajos, y en la mayoría de veces extraviándose o deteriorándose, dando como consecuencia falencias en la producción planificada. Por eso es necesario tener una herramienta flexible y tecnológicamente capaz de controlar estos procesos de información.

La aplicación planteada pretende usar tecnología de vanguardia a lo que se refiere al diseño y desarrollo de aplicaciones.

IMPACTO SOCIAL

La automatización de los procesos de control de No-Conformidades es una prioridad para la Organización que en este caso desea mejorar su nivel de productividad y con la implementación de este tipo de aplicaciones queremos dar a conocer nuevas alternativas y formas de controlar los procesos de producción para así mejorar la calidad del producto terminado, reducir los tiempos en la producción de las ordenes de fabricación, y disminuir la tasa de reclamos, de esta forma se mejorara la relación entre la Institución y el Cliente obteniendo así un cliente satisfecho.

4. PROBLEMATIZACION

PROBLEMATIZACION PRINCIPAL

Por lo general, la mayoría de las empresas presentan retrasos en sus operaciones por motivos que muchas veces no son claramente identificados, esto causa un gran impacto en procesos que son encadenados, provocando muchas veces una ruptura en la cadena del proceso, esto podría provocar atrasos e incrementos en los costos operacionales de un producto.

Si el problema no está bien identificado, es muy probable que el resto sea un esfuerzo en vano, ya que todas las partes que dependen de este proceso se verían afectadas.

Cualquiera que sea el origen del problema, es necesario proponer un análisis y solución a tiempo a las falencias encontradas.

5. OBJETIVOS

OBJETIVO GENERAL

Desarrollar un sistema que permita el control de sucesos o evidencias que se encuentran en algunos procesos denominados No-Conformidades, podríamos citar que la definición de una No-Conformidad es el incumplimiento de un requisito.

OBJETIVOS ESPECIFICOS

- Adquirir nuevos conocimientos sobre las áreas tecnológicas desconocidas por parte del desarrollador del proyecto, que sirvan de complemento para la formación académica y personal.
- Identificar la información necesaria, que permita la construcción de un modelo sólido para el desarrollo de la aplicación.
- Adquirir conocimientos sobre los lenguajes de programación y la configuración de servidores.
- Obtener experiencia en el manejo de Oracle.
- Proporcionar mecanismos de almacenamiento seguro de información mediante procesos de Backups y Logs de auditoría.

6. TEORIA REFERENCIAL O MARCO CONCEPTUAL

El software que se va a utilizar para el desarrollo de esta Tesis será:

Llaves Simples: Es un método para intercambiar información en forma segura, donde existe un único código o llave, tanto para encriptar como para desencriptar mensajes por ejemplo:

El usuario **A** desea enviarle un mensaje al usuario **B** entonces:

- El usuario **A** debe enviarle al usuario **B** una copia de su llave secreta.
- El usuario **B** encripta un mensaje con la llave secreta del usuario **A**.
- El usuario **A** decodifica el mensaje con su llave secreta.

Oracle : Es básicamente una herramienta Cliente – Servidor, es muy potente y sin duda una de las mejores Base de Datos que existen en el mercado mundial, tiene muchas características que garantizan la integridad y seguridad de los datos.

Oracle hace uso de los recursos del sistema informático en todas las arquitecturas de hardware para garantizar su aprovechamiento al máximo en ambientes cargados de mucha información.

VENTAJAS DE UTILIZACION.

Mayor Estabilidad: Los datos tienden a ser más estables que los procesos y en consecuencia la aplicación será más fácil de mantener.

Integridad con otras aplicaciones: Difícilmente una aplicación será completamente independiente de las otras. La mejor forma de lograr la integración es tener cuidado en los datos que se comparten.

Multiplataforma: Los códigos internos de Oracle son iguales a los códigos establecidos en todas las plataformas de Sistemas Operativos.

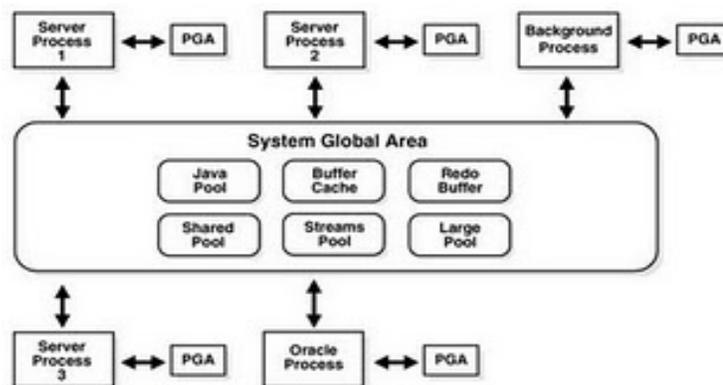
Escalabilidad y Portabilidad: Oracle corre en computadoras personales, microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo.

SQL*Plus y Plsql Developer: Para poder escribir sentencias SQL al servidor de Oracle.

Oracle soporta todas las funciones que se esperan de un servidor "serio": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.

METODOLOGIA ORACLE

Arquitectura Servidor Oracle



En un servidor se pueden crear varias instancias, pero se recomienda solo una porque cada instancia consume muchos recursos. Por cada instancia de Oracle se recomienda que se tenga una sola Base de Datos.

La arquitectura tiene mucha preponderancia como pieza clave a la hora de diseñar una solución informática.

Lo importante es comprender que para la arquitectura existen varias perspectivas.



La capa **Cliente** lo conformara la aplicación, que representa la interacción de los usuarios, quien realiza peticiones al Servidor de Aplicaciones.

La capa **Servidor de Aplicaciones** es donde residen los programas que se ejecutan recibiendo las peticiones del usuario enviando respuestas tras el proceso. Esta capa se comunica con la capa Cliente para recibir las solicitudes y presentar los resultados, y con la capa Servidor de Datos para solicitar al Gestor de Datos almacenamiento o recuperación de datos

La capa **Servidor de Datos** es donde residen los datos que lo conformara la propia Base de Datos de Oracle quien recibe peticiones de almacenamiento y recuperación de información desde la capa Servidor de Aplicaciones.

Pasos para el desarrollo de una aplicación con Oracle:

Conexión con una instancia de Oracle: de todos los servicios creados durante la instalación Oracle solo nos interesa básicamente dos:

- El servicio relacionado con la instancia y la Base de Datos, cuyo nombre tiene la siguiente estructura **OracleServiceXXX**, donde **XXX** identifica el nombre de la instancia.
- El servicio relacionado con la disponibilidad del servidor para el acceso remoto, el nombre de este servicio es **Listener**, Este servicio reside en memoria y se encarga de recibir las llamadas que llegan a la base de datos desde la red, y de pasárselas a esta.

Estos dos servicios deben estar ejecutándose y su verificación se puede ver en la ventana de servicios.

Conexión Local Utilizando SQL*Plus: Es la conexión que permite al usuario comunicarse con el servidor.

Una vez instalado Oracle en la Capa de Datos se trabaja con procedimientos, funciones, desencadenantes y paquetes que hacen más eficiente el desarrollo de las aplicaciones.

Tomado de www.oracle.com

7. PROPUESTA DE TRABAJO

Fase 1

1. INVESTIGACION DE CAMPO

- 1.1 Recopilación de Información sobre las No-conformidades en una Empresa.
- 1.2 Recopilación de Información sobre la utilización de la herramienta.
- 1.3 Recopilación de Información complementaria sobre los temas desconocidos en lo que refiere al manejo de la Herramienta de desarrollo Oracle.

Fase 2

2. ANALISIS Y DISEÑO DE LA APLICACIÓN.

2.1 Especificación de Requisitos de Software

- 2.1.1** Elaboración del Modelo Entidad – Relación.
- 2.1.2** Elaboración de los Diagramas de Procesos.

Fase 3

3. ESTUDIO Y REVISIÓN DE LAS HERRAMIENTAS PARA EL DESARROLLO DE LA APLICACIÓN.

- 3.1 Configuración del servidor en la plataforma Windows XP.
- 3.2 Configuración de los Equipos que tendrán la aplicación.

Fase 4

4. PROGRAMACIÓN DE LA APLICACIÓN

- 4.1 Instalación de Oracle
- 4.2 Desarrollo de la aplicación con la información recopilada.

Fase 5

5. SEGURIDAD DE DATOS.

- 5.1 Aplicación de seguridades utilizando métodos de encriptación en los módulos del proyecto.

Fase 6

6. PRUEBAS Y PUESTA EN OPERACIÓN DE LA APLICACIÓN.

- 6.1 Verificación del funcionamiento de la aplicación.
- 6.2 Corrección de errores de la aplicación en caso de encontrarse.
- 6.3 Puesta en producción de la aplicación.
- 6.4 Ingreso de datos y pruebas operativas.

8. ESQUEMA TENTATIVO DE CONTENIDOS DE LA TESIS.

1. Introducción

1.1 Introducción

1.2 Objetivos

1.2.1 Objetivo General.

1.2.2 Objetivos Específicos.

1.3 Concepto de Metodología de Oracle.

1.4 Conceptos de Programación con Herramientas Forms y Reports.

1.5 Conceptos de Modelos de Base de Datos.

1.6 Introducción a la Arquitectura de Aplicación Tres Capas.

2. Análisis del Sistema.

2.1 Conocimiento de Requerimientos.

2.1.1 Especificación de Requerimientos de Software.

2.1.2 Funciones del Sistema

2.2 Construcción del Modelo Conceptual.

2.3 Registro de los términos en el diccionario del modelo.

3. Diseño del Sistema.

3.1 Modelo de Datos.

3.1.1 Modelo Entidad – Relación.

3.1.2 Diccionario de Datos.

3.1.3 Esquema Físico de la Base de Datos.

3.2 Modelo de Procesos.

3.2.1 Diagrama de Contexto.

3.2.2 Diagrama de Descomposición.

3.2.3 Diseño de interfaces

4. Desarrollo de la Aplicación.

4.1 Creación de las definiciones de la Base de Datos.

4.1.1 Definición de los atributos.

4.2 Programación en Oracle.

5. Implementación y Pruebas.

5.1 Configuración del Servidor.

5.2 Instalación de la aplicación.

5.3 Corrección de errores encontrados.

6. Documentación

2.1 Manual Técnico.

2.2 Manual de Usuario.

7. Conclusiones y recomendaciones.

8. PROCEDIMIENTOS METODOLOGICOS.

Para la recopilación de información nos hemos basado en la técnica de investigación documental utilizando:

- Entrevistas: Para conocer los requisitos que deberá cumplir la aplicación a desarrollar, se empleara esta técnica a las siguientes personas:
 - Ing. Gina Motta. (Gerente de SIG).
 - Ing. José Millán Anuar. (Presidente Corporativo).
 - Dr. Geovani Coellar (Gerente General).
- Libros : Para obtener conocimientos con el objetivo de captar información sobre Diseño de Base de Datos, Lenguajes de Programación.
- Internet: La navegación será de gran utilidad para buscar información sobre las herramientas que se utilizarán ya que permite obtener manuales y bibliografía de actualidad.

9. RECURSOS.

RECURSOS HUMANOS

Con el propósito de alcanzar los objetivos planteados en este diseño las personas que participaran en la elaboración del sistema son:

- Director de Tesis
 - Ing. Lenin Erazo Garzón.
- Asesoría en el Diseño de Tesis
 - Mgs. Carlos Alberto Tirado.
- Desarrollador
 - Jackson Marcelo Serrano Tinitana.

RECURSOS TECNICOS

HARDWARE:

El hardware necesario, para que nuestra aplicación Web funciones depende de ciertos factores, como el trafico del sitio que se desea implementar, la complejidad de las aplicaciones, el tamaño de las tablas y los demás servicios que se están ejecutando en el servidor, los equipos con que contamos para el desarrollo de nuestro sistema son:

- Computador AMD Turion 64 Bits X2, Memoria RAM de 2Gb,, HD 160 Gb.
- Impresora SAMSUNG LASER MULT. SCX-4300
- Canon LiDE100 Color Image Scanner

SOFTWARE

- Microsoft Windows XP Professional Edition.
- Erwin Data Modeler.
- Microsoft Office 2007.
- Oracle 10g.
- Plsql Developer.
- Virtual Machine VM Ware.
- Utilitarios.

10. TIEMPO

- El tiempo estimado para la culminación de la Tesis es 6 meses.

11. RECURSOS FINANCIEROS

CANTIDAD	MATERIAL	VALOR UNITARIO	VALOR TOTAL	DESCRIPCION
2	Resma Papel Bond	7	14	Para la impresión de la documentación.
2000	Copias	0.02	40	Para fotocopias.
1	Toner	40	40	Para impresión.
1	Memoria Flash 2Gb	30	30	Para grabar información.
1	Telefono	50	50	Para comunicación.
1	Gastos Varios	100	100	Movilización, Agua, Luz, Teléfono.
1	Gastos Imprevistos	100	100	Gastos Adicionales.
1	Internet	100	100	Para investigación.

BIBLIOGRAFIA.

INGENIERIA DE SOFTWARE UN ENFOQUE PRACTICO QUINTA EDICION.

PRESMAN, Roger S.

Editorial Mc-Graw Hill, 2002.

LENGUAJE UNIFICADO DE MODELADO.

BOOCH, Grady; RUMBAUGH. James: JACOBSON, Ivar

Addison Wesley Iberoamericana. Madrid 1999.

FUNDAMENTOS DE BASE DE DATOS CUARTA EDICION.

SILBERSCHATZ Abraham; KORTH Henry F.

MC-Graw Hill, 2002.

DEVELOPER MANUAL AVANZADO DE FORMS Y REPORTS

PETER KO-LETZKE.

Editorial, Osborne Mac-Graw Hill.

SITIOS WEB:

- www.oracle.com
- www.osborne.com
- www.iso.org

ACTIVIDAD	CRONOGRAMA DE ACTIVIDADES																											
	MES 1				MES 2				MES 3				MES 4				MES 5				MES 6							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
INVESTIGACION DE CAMPO	■	■	■	■																								
Recopilación de Información sobre las No-Conformidades	■	■																										
Recopilación de Información sobre la herramienta		■	■																									
Recopilación de Información complementaria			■	■																								
ANALISIS Y DISEÑO DE LA APLICACION																												
Elaboración del Modelo Entidad Relación					■	■	■																					
Elaboración de los Diagramas de Procesos						■	■	■																				
ESTUDIO Y REVISION DE LAS HERRAMIENTAS PARA EL DESARROLLO DE LA APLICACION																												
Configuración del Servidor en la Plataforma Windows XP											■																	
Configuración de los Equipos que tendrán la aplicación											■																	
PROGRAMACION DE LA APLICACION																												
Instalación y configuración de Base de Oracle 10g											■																	
Instalación y configuración de Forms y Reports 10g											■																	
Configuración de Servidor de Impresión Report Server											■																	
Configuration de WebUtil											■																	
Desarrollo de la Aplicación con la información recopilada												■	■	■	■	■	■	■	■	■								
SEGURIDAD DE DATOS																												
Aplicación de seguridades utilizando métodos de encriptación en los módulos del proyecto																										■		
PRUEBAS Y PUESTA EN OPERACION DE LA APLICACION																												
Verificación del funcionamiento de la aplicación																											■	
Corrección de errores de la aplicación en caso de encontrarse																											■	
MANUAL DE USUARIO																												
Elaboración de manual de usuario																												■