



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**“SISTEMA DE GESTIÓN DE BASE DE DATOS PARA EL SISTEMA DE
SIMULACIÓN Y MONITOREO DEL RENDIMIENTO DE CENTRALES
TERMOELÉCTRICAS – LAPEM TERMO®”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

P R E S E N T A N :

Y O O J I A H

ZÁRRAGA MARTÍNEZ JESÚS ALFREDO

DIRECTOR DE TESIS: M.C. LUIS OCHOA TOLEDO

MÉXICO, D.F., MARZO 2009



A G R A D E C I M I E N T O

A M.C. LUIS OCHOA TOLEDO

Por darnos la oportunidad de trabajar en este proyecto, por todo el apoyo que nos brindaste primero como profesor y después como jefe, por todas y cada una de las cosas que aprendimos de ti, por brindarnos un lugar en tu equipo de trabajo y sobre todo por ser un gran profesor y una gran persona, MUCHAS GRACIAS.

A ING. GABRIELA BETZABÉ LIZÁRRAGA RAMÍREZ

Por todas y cada una de las clases que tuvimos la oportunidad de tomar con usted, por todas las enseñanzas, por toda la paciencia y asesoría brindada para la presente tesis, por todo el esfuerzo y dedicación que realiza como profesora para cada una de sus clases y por siempre estar preocupada para que nosotros como alumnos aprendamos bien lo que necesitaremos en el mundo laboral, MUCHAS GRACIAS.

A NUESTRA FACULTAD DE INGENIERÍA

Por permitirnos ser parte de tan honorable escuela, por brindarnos la oportunidad de tener una educación de alto nivel, por prepararnos muy bien para la vida profesional y por darnos la oportunidad de hacer realidad uno de los más grandes sueños de nuestra vida, MUCHAS GRACIAS.

DEDICATORIA

Para mi esposa:

Quiero dedicarte especialmente a ti todo el trabajo que realicé durante mi tesis, porque eres tú la mayor motivación y la mayor inspiración que tengo en mi vida, porque eres tú la persona más importante para mí, porque eres tú con quien quiero compartir y vivir todos mis sueños, ilusiones y logros, y de la misma forma los momentos difíciles, que son los que nos hacen más fuertes, tanto como personas como pareja. Y sobre todo porque por cada día que paso a tu lado, me enamoro más de ti. Muchas Gracias por estar a mi lado y por hacerme tan feliz. Para ti Miriam Pavón Marín.

Para mi padre:

Por apoyarme durante todos estos años que estuve estudiando, por brindarme la oportunidad de ser una mejor persona, por enseñarme que el camino para poder superarme personal y profesionalmente está en el esfuerzo y dedicación que ponga a cada uno de los aspectos de mi vida. Aunque estás lejos, siempre recuerdo lo que me has enseñado y lo que me has demostrado al salir adelante tú sólo a pesar de todos los obstáculos que tuviste en esta vida. Muchas Gracias papá, donde quiera que estés.

Para mi madre:

Por estar conmigo y apoyarme para que pudiera terminar mi carrera, por alentarme a realizar mis sueños y por siempre aceptar y respetar las decisiones que he tomado a lo largo de mi vida. Muchas gracias mamá.

Para mis hermanas y para mi hermano:

Por apoyarme y darme ánimos para que pudiera terminar mi carrera, y aunque todos estamos separados y casi no nos vemos, siempre voy a recordar todo el tiempo que convivimos cuando estábamos pequeños. Muchas Gracias por todo.

Para mi compañera de tesis:

Anita, te conocí desde el primer semestre de nuestra carrera, hemos tenidos una larga amistad de más de 5 años, hemos pasado por muchas cosas y por muchas materias juntos, y siempre nos entendimos muy bien, nos acoplamos bien trabajando en equipo al grado de casi nunca tener problemas. Quiero dedicarte a ti también todo este trabajo que realizamos juntos, gracias por todo, pero sobre todo por ser mi compañera y también mi amiga.

Para mi sobrina María del Rocío:

También quiero dedicarte a ti este trabajo, porque muchas veces estuviste conmigo cuando hacía mis tareas o estudiaba para mis exámenes, porque siempre me dedicas un rato para estar conmigo, y porque siempre has confiado en mí y me has apoyado para que termine mi carrera. Y sobre todo porque te quiero mucho, y sé que tu también a mí.

Para mis amigos:

Para Francisco Mejía Rivera, María del Rocío Ortiz Albarrán, Víctor Alfonso Luna Álvarez y para todos mis demás amigos, muchas gracias por estar conmigo y por todo su apoyo.

Para mis profesores:

Para todos y cada uno de los profesores que tuve a lo largo de mi carrera, porque de todos aprendí algo importante para aplicar tanto a mi vida profesional como personal, muchas gracias a todos.

Jesús Alfredo Zárraga Martínez

DEDICATORIA

A mis padres, porque a ellos les debo todo. Gracias a ellos soy quien soy hoy en día. Gracias por su comprensión y apoyo incondicional durante todos estos años.

A mis hermanas que siempre están a mi lado y compartimos los pequeños secretos que solo se viven entre hermanas. Clara, la pequeña de la familia, juntas hemos superado las dificultades y logrado adaptarnos a todo un mundo llamado México durante los últimos ocho años, y por supuesto mi hermana mayor, la rebelde de la familia, que a pesar de estar del otro lado del continente siempre me apoyas en todas mis decisiones.

A mis amigos, los que han pasado y los que se han quedado, porque todos ustedes han dejado alguna marca en mi vida; Tere, que a pesar de la distancia sigues siendo mi mejor amiga, Nora, siempre dispuesta a ayudar como sea, mis amigas argentinas Azu, Lau, Nani y Lu, que aunque no las veo hace años, siempre están cerca para escucharme e inspirarme, y Alfredo, un gran amigo, hicimos tantas cosas juntos a lo largo de la carrera y esta tesis es nuestro trabajo más importante.

Yoo Ji Ah

ÍNDICE TEMÁTICO

| | |
|---|----|
| Introducción | 14 |
| Capítulo 1. Introducción general. | 16 |
| 1.1 Objetivos | 16 |
| 1.2 Planteamiento del problema | 16 |
| 1.3 Justificación | 18 |
| Capítulo 2. Conceptos básicos | 20 |
| 2.1 Energía eléctrica | 20 |
| 2.2 Plantas de generación termoeléctrica | 22 |
| 2.3 CFE y las centrales termoeléctricas de México | 25 |
| 2.4 Conceptos fundamentales de bases de datos | 30 |
| 2.4.1 Concepto de base de datos | 32 |
| 2.4.2 Sistema de gestión de base de datos – SGBD | 32 |
| 2.4.3 Usuarios de la base de datos | 33 |
| 2.4.4 Arquitectura de una base de datos | 34 |
| 2.4.5 Modelos de datos | 36 |
| 2.4.6 Clasificación de los modelos de datos | 37 |
| 2.4.7 Modelo entidad-relación | 38 |
| 2.4.8 Modelo de datos relacional | 41 |
| 2.4.8.1 Propiedades de las relaciones del modelo relacional | 43 |
| 2.4.8.2 Tipos de relaciones del modelo relacional | 43 |
| 2.4.8.3 Llaves del modelo relacional | 44 |

| | | |
|---------------------------|---|-----------|
| 2.4.8.4 | Esquema de una relación del modelo relacional | 44 |
| 2.4.8.5 | Reglas de integridad | 45 |
| 2.4.8.6 | Manejo de los datos | 46 |
| 2.4.9 | Paso del modelo entidad-relación al modelo relacional | 47 |
| 2.5 | Base de datos en Access | 47 |
| 2.6 | Base de datos en SQL Server | 48 |
| 2.7 | Desarrollo en plataforma .NET | 49 |
| 2.7.1 | Conexión e interoperabilidad de bases de datos | 49 |
| 2.7.2 | Universal Data Access | 51 |
| 2.7.3 | Microsoft Data Access Components | 52 |
| 2.7.4 | ODBC | 52 |
| 2.7.5 | OLE DB | 53 |
| Capítulo 3. Diseño | | 54 |
| 3.1 | Descripción del Sistema de Simulación y Monitoreo del Rendimiento de Centrales Termoeléctricas: LAPEM – TERMO® | 54 |
| 3.1.1 | Antecedentes | 54 |
| 3.1.2 | Solución propuesta | 56 |
| 3.1.3 | Objetivos del sistema LAPEM – TERMO® | 58 |
| 3.1.4 | Módulos del sistema LAPEM – TERMO® | 58 |
| 3.2 | Descripción del sistema de gestión de base de datos para el Sistema de Simulación y Monitoreo del Rendimiento de Centrales Termoeléctricas LAPEM – TERMO® | 62 |
| 3.2.1 | Diseño de la base de datos para el sistema LAPEM – TERMO® | 64 |
| 3.2.2 | Diseño de la interfaz gráfica para la base de datos del sistema LAPEM – TERMO® | 78 |

| | |
|--|-----|
| Capítulo 4. Desarrollo | 87 |
| 4.1 Desarrollo de la base de datos en Microsoft Access | 87 |
| 4.2 Desarrollo de la base de datos en Microsoft SQL Server | 97 |
| 4.3 Seguridad | 109 |
| 4.4 Acceso a la base de datos | 110 |
| | |
| Capítulo 5. Prueba de software e implementación del sistema | 132 |
| 5.1 Señales de entrada | 133 |
| 5.2 Señales de salida | 134 |
| 5.3 Coeficientes de desviaciones del ciclo | 134 |
| | |
| Capítulo 6. Conclusiones | 138 |
| | |
| Anexo 1. Diccionario de datos | 142 |
| | |
| Anexo 2. Ejemplos de modelos de alto nivel | 152 |
| | |
| Bibliografía | 158 |
| | |
| Referencias de internet | 159 |

ÍNDICE DE IMÁGENES

| | | |
|----------------|--|----|
| Imagen 2.2.1 | Esquema de una central termoeléctrica | 23 |
| Imagen 2.2.2 | Centrales termoeléctricas tipo vapor | 24 |
| Imagen 2.3.1 | Sectores de consumo de electricidad en México | 26 |
| Imagen 2.3.2 | Centrales generadoras de electricidad en México | 27 |
| Imagen 2.3.3 | Centrales termoeléctricas del sistema LAPEM – TERMO® | 29 |
| Imagen 2.4.4.1 | Arquitectura de una base de datos | 35 |
| Imagen 2.4.7.1 | Conceptos del modelo entidad-relación | 39 |
| Imagen 2.7.1.1 | Arquitectura de aplicación distribuida | 50 |
| Imagen 2.7.4.1 | Interfaz de un ODBC | 52 |
| Imagen 3.1.4.1 | Estructura funcional del sistema LAPEM – TERMO® | 59 |
| Imagen 3.1.4.2 | Operación funcional del sistema LAPEM – TERMO® | 60 |
| Imagen 3.2.1 | DBMS del sistema LAPEM – TERMO® | 63 |
| Imagen 4.1.1 | Creando una base de datos en Access 1 | 88 |
| Imagen 4.1.2 | Creando una base de datos en Access 2 | 88 |
| Imagen 4.1.3 | Creando una tabla en Access 1 | 89 |
| Imagen 4.1.4 | Creando una tabla en Access 2 | 89 |
| Imagen 4.1.5 | Creando una tabla en Access 3 | 90 |
| Imagen 4.1.6 | Creando una tabla en Access 4 | 90 |
| Imagen 4.1.7 | Creando una tabla en Access 5 | 90 |
| Imagen 4.1.8 | Creando una tabla en Access 6 | 90 |
| Imagen 4.1.9 | Creando una tabla en Access 7 | 90 |
| Imagen 4.1.10 | Creando una tabla en Access 8 | 91 |
| Imagen 4.1.11 | Creando una tabla en Access 9 | 91 |

| | | |
|---------------|---|-----|
| Imagen 4.1.12 | Tabla "Plantas" en vista Diseño | 92 |
| Imagen 4.1.13 | Tabla "Plantas" en vista Hojas de Datos | 92 |
| Imagen 4.1.14 | Tabla "Unidades" en vista Diseño | 93 |
| Imagen 4.1.15 | Tabla "Unidades" en vista Hojas de Datos | 93 |
| Imagen 4.1.16 | Tabla "Equipos" en vista Diseño | 94 |
| Imagen 4.1.17 | Tabla "Equipos" en vista Hojas de Datos | 94 |
| Imagen 4.1.18 | Base de Datos del sistema LAPEM – TERMO® con sus 12 tablas | 95 |
| Imagen 4.2.1 | Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005 | 98 |
| Imagen 4.2.2- | Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005 | 98 |
| Imagen 4.2.3 | Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005 | 99 |
| Imagen 4.2.4 | Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005 | 99 |
| Imagen 4.2.5 | Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005 | 100 |
| Imagen 4.2.6 | Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005 | 100 |
| Imagen 4.2.7 | Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005 | 101 |
| Imagen 4.2.8 | Base de Datos del Sistema LAPEM – TERMO® en Microsoft SQL Server 2005 | 102 |
| Imagen 4.2.9 | Diseño de la tabla Plantas en Microsoft SQL Server 2005 | 103 |
| Imagen 4.2.10 | Contenido de la tabla Plantas en Microsoft SQL Server 2005 | 104 |
| Imagen 4.2.11 | Diseño de la tabla Unidades en Microsoft SQL Server 2005 | 105 |
| Imagen 4.2.12 | Contenido de la tabla Unidades en Microsoft SQL Server 2005 | 106 |

| | | |
|---------------|--|-----|
| Imagen 4.2.13 | Diseño de la tabla Equipos en Microsoft SQL Server 2005 | 106 |
| Imagen 4.2.14 | Contenido de la tabla Equipos en Microsoft SQL Server 2005 | 107 |
| Imagen 4.3.1 | Modulo de validación de usuarios del sistema LAPEM-TERMO® | 109 |
| Imagen 4.3.2 | Modulo de validación de usuarios del sistema LAPEM-TERMO® | 109 |
| Imagen 4.4.1 | Tecnología ADO.NET | 111 |
| Imagen 4.4.2 | Menú de configuración del sistema LAPEM-TERMO® | 119 |
| Imagen 4.4.3 | Menú de configuración de Plantas del sistema LAPEM-TERMO® | 119 |
| Imagen 4.4.4 | Reporte de Datos Generales de Referencia del sistema LAPEM-TERMO® | 123 |
| Imagen 4.4.5 | Menú para respaldar en Access en el sistema LAPEM-TERMO® | 124 |
| Imagen 4.4.6 | Menú para respaldar en Archivo Plano en el sistema LAPEM-TERMO® | 129 |
| Imagen 5.1 | Menú Configuración del sistema LAPEM-TERMO® | 132 |
| Imagen 5.2 | Señales de entrada del generador de vapor-vapor de la unidad 2 de Manzanillo II | 133 |
| Imagen 5.3 | Configuración de señales de salida en el sistema LAPEM-TERMO® | 134 |
| Imagen 5.4 | Configuración de los coeficientes de las ecuaciones en el sistema LAPEM-TERMO® | 135 |
| Imagen 5.5 | Configuración del modo automático en el sistema LAPEM-TERMO® | 135 |

ÍNDICE DE CUADROS

| | |
|--|----|
| Cuadro 2.1.1 – Tipos de Plantas Generadoras de Energía Eléctrica | 22 |
| Cuadro 2.3.2 - Capacidad efectiva instalada por tipo de generación al mes de junio de 2008 | 27 |
| Cuadro 2.3.3 – Centrales Termoeléctricas del Sistema LAPEM-TERMO®. | 28 |
| Cuadro 2.3.4 – Centrales Termoeléctricas no incluidas en el Sistema “LAPEM-TERMO®” | 30 |
| Cuadro 2.4.8.4.1 – Tabla de un ejemplo del Modelo Relacional | 45 |
| Cuadro 3.2.1.1-Tipos de datos que soporta el SQL de Microsoft® Jet versión 4.0 | 75 |

ÍNDICE DE DIAGRAMAS

| | |
|---|-----|
| Diagrama 3.2.1.1-Modelo de la Base de Datos para el Sistema LAPEM-TERMO® | 77 |
| Diagrama 3.2.2.1-Modelo Jerárquico de la Interfaz Gráfica del Sistema LAPEM-TERMO® | 81 |
| Diagrama 3.2.2.2-Diagrama de Flujo del Nivel 0 de la GUI del Sistema LAPEM-TERMO® | 82 |
| Diagrama 3.2.2.3-Diagrama de Flujo del Nivel 1 de la GUI del Sistema LAPEM-TERMO® | 83 |
| Diagrama 3.2.2.4-Diagrama de Flujo del Nivel 2 de la GUI del Sistema LAPEM-TERMO® | 84 |
| Diagrama 3.2.2.5-Diagrama de Flujo del Nivel 3 de la GUI del Sistema LAPEM-TERMO® | 85 |
| Diagrama 3.2.2.6-Diagrama de Flujo del Nivel 4 de la GUI del Sistema LAPEM-TERMO® | 85 |
| Diagrama 3.2.2.7-Diagrama de Flujo del Nivel 5 de la GUI del Sistema LAPEM-TERMO® | 86 |
| Diagrama 4.1.19 -Diagrama de la Base de Datos del sistema LAPEM-TERMO® en Microsoft Access. | 96 |
| Diagrama 4.2.15- Diagrama de la Base de Datos del sistema LAPEM-TERMO® en Microsoft SQL Server 2005. | 108 |
| Diagrama 5.6- Modelo de prueba para el sistema LAPEM-TERMO® | 137 |

INTRODUCCIÓN

La generación de energía en la actualidad es un factor muy importante para el crecimiento económico de cualquier país. Entre los tipos de energía más importantes por su amplio uso y facilidad de generación destaca la energía eléctrica. Para generarla existen muchas maneras, pero la más popular en México y en el mundo es la generación de energía eléctrica por medio de centrales termoeléctricas, las cuales se basan en el principio de mover una turbina conectada a un generador eléctrico, a base de la presión generada por la quema de combustibles fósiles, como la madera o el carbón. Este proceso de generación sobresale de los demás por su bajo costo y facilidad de implementación, lo cual ha hecho que más de la mitad de la producción total de energía eléctrica en México se realice de esta manera.

Es importante darnos cuenta de lo trascendental que son las centrales termoeléctricas en nuestro país, así como de la importancia de su correcto funcionamiento y de lo fundamental que resulta el tener un buen manejo de ellas, ya que un desperfecto (como puede ser una falla parcial o total en la central), una baja en la producción o un mal empleo de los recursos disponibles, puede impactar directamente en la jornada laboral de miles de mexicanos, lo que se traduce en pérdidas económicas que afectan el desarrollo del país.

Por ello, en la presente tesis se expone el análisis, diseño, desarrollo e implementación de un **SISTEMA DE GESTIÓN DE BASE DE DATOS** para el proyecto: “**SISTEMA DE SIMULACIÓN Y MONITOREO DEL RENDIMIENTO DE CENTRALES TERMOELÉCTRICAS – LAPEM TERMO®**”; el cual atiende la necesidad antes planteada, que es la de tener un buen manejo de las centrales termoeléctricas de México para lograr eficacia y eficiencia en la producción de energía eléctrica.

Para empezar habrá que definir el concepto general de los requerimientos que la **COMISIÓN FEDERAL DE ELECTRICIDAD – CFE** requerirá para el monitoreo y simulación de sus centrales termoeléctricas, así como los objetivos principales que se deberán cumplir, el planteamiento del problema y de su solución debidamente justificada. Todo esto se llevará a cabo dentro del primer capítulo de ésta tesis.

Más adelante, dentro del capítulo dos, será necesario abordar algunos conceptos fundamentales que estén relacionados con la generación de energía termoeléctrica, para así poder entender de mejor manera las características más importantes de las centrales termoeléctricas que existen en México. También se aborda el tema de los conceptos fundamentales de las bases de datos, que son y para que nos sirven, así como la manera de diseñarlas correctamente. También se tratará la forma de desarrollar bases de datos en dos plataformas: Access y SQL Server 2005, ya que en estos dos manejadores se desarrollará la base de datos para el sistema LAPEM-TERMO®. Además y como parte final de este capítulo, se dará una breve explicación del desarrollo de sistemas en .NET, ya que será la plataforma en la que se llevará a cabo este proyecto.

En el CAPÍTULO tres se trata el análisis y diseño del Sistema de Gestión de Base de Datos. Para ello, primero se tendrán que definir los requerimientos dentro de la etapa de análisis, tomando en cuenta las plataformas elegidas tanto para el desarrollo de la base de datos como para el desarrollo del sistema. Después se realizará el diseño conceptual para cada módulo que se va a integrar dentro del Sistema de Gestión de Base de Datos, así como de la manera de comunicarse con el sistema LAPEM-TERMO®.

Con el Sistema de Gestión de Base de Datos diseñado, se procederá al desarrollo del mismo, utilizando las dos plataformas descritas en el CAPÍTULO dos, así como las normas y estándares más utilizados. También se utilizará la técnica de programación orientada a objetos y la plataforma .NET para poder comunicar la base de datos con la interfaz gráfica del sistema LAPEM-TERMO®. Todo ello dentro del capítulo cuatro de la presente tesis.

Con lo anterior realizado, se llevará a cabo las pruebas de software del sistema así como la implementación del mismo; todo esto se desarrollará en el CAPÍTULO cinco. Las pruebas incluirán monitoreo y simulaciones dentro de una implementación en la central termoeléctrica de Manzanillo, dentro de sus unidades I y II.

En el último capítulo se mencionaran los resultados que se obtuvieron al terminar de desarrollar la tesis, se discutirá si se cumplieron los objetivos o no, se mencionaran los resultados de la implantación del sistema y de las pruebas realizadas en la central termoeléctrica de Manzanillo, y también se detallaran las posibles mejoras que puedan sugerirse para versiones futuras.

CAPÍTULO 1

INTRODUCCIÓN GENERAL

1.1 Objetivos

El principal objetivo de ésta tesis es analizar, diseñar, desarrollar e implementar el **SISTEMA DE GESTION DE BASE DE DATOS** para el proyecto: “**SISTEMA DE SIMULACIÓN Y MONITOREO DEL RENDIMIENTO DE CENTRALES TERMOELÉCTRICAS – LAPEM TERMO®**”.

1.2 Planteamiento del problema

Las Centrales Termo Eléctricas son aquellas que emplean la combustión del carbón, petróleo o gas natural para generar la energía eléctrica. Son consideradas las centrales más económicas y rentables, su utilización está muy extendida en México al proporcionar alrededor del 75% del consumo nacional por lo que su correcto funcionamiento es trascendental para la estabilidad del mismo.

EL SISTEMA DE SIMULACIÓN Y MONITOREO DEL RENDIMIENTO DE CENTRALES TERMOELÉCTRICAS – LAPEM-TERMO® es un proyecto solicitado por el Laboratorio de Pruebas de Equipos y Materiales “LAPEM” de la Comisión Federal de Electricidad “CFE” al Centro de Ciencias Aplicadas y Desarrollo Tecnológico “CCADET” de la Universidad Nacional Autónoma de México “UNAM” para diseñar, desarrollar e implementar un sistema de simulación y monitoreo del rendimiento de centrales Termo Eléctricas para facilitar la evaluación de pruebas del régimen térmico, Consumo Térmico Unitario (CTU) y costos Termo Económicos, a partir de los modelos matemáticos proporcionadas por gente del “LAPEM”. El sistema tendrá la capacidad de operar totalmente en línea con los sistemas de adquisición de datos de las plantas Termo Eléctricas, para ello se desarrollara una versión StandAlone para instalarse en cada planta y una versión Web para monitoreo de toda la red de centrales y se instalara en las instalaciones del LAPEM.

El problema que busca resolver esta propuesta dentro del proyecto LAPEM-TERMO® es implantar un sistema de gestión de Base de Datos que permita operar en tiempo real y de forma integral la información necesaria para la simulación y monitoreo en línea de una planta Termo Eléctrica, logrando con ello contar con una herramienta útil para las Centrales Termo Eléctricas del país que ayude a mejorar su funcionamiento y aumentar su eficiencia y productividad, para lograr este objetivo se definen una serie de actividades que serán las siguientes:

1. Análisis de las Plantas Termo Eléctricas del país y detección de parámetros operativos dentro del modelo matemático.
2. Desarrollo del Sistema de Gestión de Base de Datos.
 - 2.1. Definición de las tablas de la Base de Datos en base a la información de cada planta Termoeléctrica.
 - 2.2. Integración de los mecanismos de comunicación entre la interfaz de usuario del sistema de simulación con la información contenida en la Base de Datos.
 - 2.3. Desarrollo y construcción de la Base de Datos en Microsoft Access para la versión StandAlone y en SQL Server 2005 para la versión WEB.
 - 2.4. Generación de reportes y gráficas para el sistema de simulación en base a la información contenida en la Base de Datos.
 - 2.5. Desarrollo de las interfaces de usuario que interactúen con la Base de Datos.
3. Realizar pruebas del sistema
 - 3.1. Pruebas para verificar la correcta integridad de la Base de Datos en operación con el sistema de simulación.
 - 3.2. Pruebas de integración en tiempo real del sistema de adquisición de datos (OPC) con el sistema LAPEM TERMO®.
 - 3.3. Pruebas generales para todo el sistema verificando su buen funcionamiento.
 - 3.4. Instalación y Validación de la información en la Central Termo Eléctrica de Manzanillo II, unidades 1 y 2.

4. Redacción de la documentación del sistema de Base de Datos, así como de la integración con el sistema LAPEM TERMO®.
5. Redacción de la Tesis

1.3 Justificación

Para desarrollar este sistema debemos hacer dos tipos de elecciones:

- 1.- Elegir el lenguaje de programación más adecuado para la parte del front end del sistema (es decir, el que se encargara del control y correcta ejecución de las instrucciones brindadas por el usuario), así como para desarrollar la interfaz de usuario gráfica (GUI).
- 2.- Elegir la forma de trabajar con las dos herramientas en las cuales construiremos la Base de Datos, ya que ésta se encargará de almacenar la información del sistema, tarea primordial ya que sin información el Sistema de Gestión de Base de Datos y el Sistema LAPEM-TERMO® no sirve de nada.

En base a los dos puntos de elección anteriores podemos empezar a analizar las alternativas disponibles para la primera:

Existen varios lenguajes de programación poderosos que nos permiten realizar aplicaciones de este tipo, los cuales están distribuidos mediante herramientas comerciales. Estas herramientas también incluyen una parte de desarrollo gráfico para presentarlas con atractivas interfaces de usuario (como ventanas, menús, íconos, botones, etc.).

En la actualidad existen dos plataformas de desarrollo de sistemas de este tipo. La primera, de la empresa Sun Microsystems que utiliza su lenguaje de programación Java y que está disponible en entornos de desarrollo integrado (IDE)¹ como por ejemplo NetBeans o Eclipse y el otro de la empresa Microsoft llamado “.NET

¹ Un **entorno de desarrollo integrado** o, en inglés, *Integrated Development Environment ('IDE')*, es un entorno de programación compuesto por un conjunto de herramientas para un programador que ha sido empaquetado como un programa de aplicación

Framework” el cual busca ser una plataforma de desarrollo de software en varios lenguajes de programación (Visual Basic, C#, J#, C++) y cuyo entorno de desarrollo comercial se llama Visual Studio.

Nosotros elegiremos la segunda alternativa, utilizando el lenguaje de programación “C#”, por la razón de que cuenta con dos grandes ventajas:

- **Cuenta con un desarrollo rápido de aplicaciones**, lo cual es importante ya que además de ser una herramienta poderosa, nos permite realizar de manera más sencilla y rápida una aplicación evitando dar más tiempo a la etapa de desarrollo del sistema que a la etapa de análisis y diseño (como sabemos es muy recomendable que sea al contrario, ya que dedicando más tiempo a un buen análisis y a un buen diseño, se realizará más rápido la etapa de desarrollo).
- **Cuenta con una poderosa, atractiva y sencilla manera de diseñar ventanas, formularios y demás detalles para la interfaz gráfica del usuario**; lo cual brinda una ventaja importante en comparación a otras que no pueden llegar a ser tan sencillas de manejar.

Para la segunda elección que es de donde sale la propuesta del Sistema de Gestión de Base de Datos requerido para el sistema LAPEM-TERMO® en cuanto a la plataforma a utilizar se encuentra acotada en función de los requerimientos por parte del cliente para desarrollarse en la plataforma Windows, con el manejador de Base de Datos Microsoft Access y SQL Server

Con esta elección limitada no tenemos desacuerdo, ya que ambas son buenas herramientas, poderosas, sencillas de manejar, fáciles de instalar y muy comerciales, motivos por los que CFE pudo haberlas elegido.

La metodología de diseño de Base de Datos que se empleara será la relacional para soportar el tipo de jerarquías que se manejan dentro de la información que se pretende maneje el Sistema de Gestión de Base de Datos para el Sistema LAPEM –TERMO®.

Con estas dos elecciones hechas quedan definidas las herramientas con las cuales buscaremos resolver nuestro problema planteado en la sección 1.2.

CAPÍTULO 2

CONCEPTOS BÁSICOS

A lo largo de este capítulo se abordarán los conceptos más importantes que se utilizan como base durante el desarrollo de ésta tesis. En el capítulo 1 ya se mencionó el panorama general de los conceptos a exponer y los cuales se detallan a continuación.

2.1 Energía Eléctrica

La energía eléctrica en el mundo en el que hoy vivimos juega un papel muy importante. Sería imposible vivir sin las comodidades que nos proporciona, pues ya hace muchos años que se ha convertido en parte de la vida del hombre. Sin ella sería imposible imaginar los niveles de progreso que el mundo ha alcanzado, y sería aun más difícil imaginarse que el mundo siga progresando sin ella.

Pero ¿qué es la energía eléctrica? Toda la materia está compuesta por átomos, y éstos a su vez por partículas más pequeñas: electrones, protones y neutrones. Los protones y neutrones permanecen en el núcleo del átomo, mientras los electrones giran alrededor de éste. Los protones tienen una carga positiva, los electrones una carga negativa y los neutrones, como su nombre lo dice, tienen una carga neutra.

Algunos materiales están formados por átomos que pierden fácilmente sus electrones, y cuando pasa esto los electrones pueden pasar de un átomo a otro. Hablado en términos sencillos podríamos decir que la energía eléctrica (ó electricidad) no es otra cosa que los electrones en movimiento, así cuando éstos se mueven de un átomo a otro se crea una corriente de electricidad. Esto es lo que pasa en los cables que transportan la energía eléctrica a nuestros hogares.

Ya hemos visto que es la energía eléctrica, pero la pregunta que surge ahora es ¿cómo se genera? Existen varias formas de hacerlo, y en todas se utilizan los recursos de nuestro medio ambiente, desde el flujo del agua en ríos o cascadas, el calor producido al quemar combustibles fósiles (como el carbón, la madera, derivados del petróleo, gas natural, etc.), la geotermia (el calor interno de la Tierra), la energía nuclear hasta las corrientes de viento (energía eólica) y los rayos del sol (energía solar).

Existen distintos tipos de plantas generadoras de energía eléctrica, que varían dependiendo el recurso del medio ambiente que aprovechen para generar la electricidad (alguno de los antes mencionados). Pero hablando más estrictamente y obedeciendo al Principio de Conservación de la Energía (el cual nos dice que la energía no se crea ni se destruye, solo se transforma), lo que hacen la mayoría de estas plantas es jugar con las leyes de la Física para transformar algún tipo de energía, como la eólica o la solar, en energía eléctrica. Los tipos de plantas generadoras más importantes hoy en día son las siguientes:

| NOMBRE | DESCRIPCIÓN | ENERGÍA DE ORIGEN |
|-----------------------------------|--|--|
| Plantas Hidroeléctricas | Funcionan en base al flujo ó caída del agua, el cual se aprovecha para mover una turbina conectada a un generador de energía eléctrica. | Energía Cinética, Energía Potencial |
| Plantas Termoeléctricas | Funcionan en base al calor y vapor de agua de una caldera, obtenido al quemar combustibles fósiles. El vapor se eleva a una gran presión lo que hace mover una turbina conectada a un generador de energía eléctrica. | Energía Calorífica, Energía Cinética. |
| Plantas Geotermoeléctricas | Funcionan de igual manera que las plantas termoeléctricas, pero en lugar de combustibles se utiliza el calor interno de la Tierra (principalmente el vapor seco que se emana de las fracturas en el suelo) para mover la turbina y el generador. | Energía Geotérmica |
| Plantas Nucleoeléctricas | Utilizan la energía nuclear del átomo para producir vapor de agua y así mover las turbinas y generadores que dan origen a la energía eléctrica. | Energía Nuclear |

| | | |
|-------------------------------|--|----------------|
| Plantas Eoloeléctricas | Utilizan la fuerza de las corrientes de los vientos para que ésta mueva unos aerogeneradores que producen la energía eléctrica al girar. | Energía Eólica |
| Plantas Termosolares | Utilizan el calentamiento de un fluido mediante radiación solar, en un ciclo termodinámico tradicional para así mover la turbina y el generador que dan origen a la energía eléctrica. | Energía Solar |

Cuadro 2.1.1 - Tipos de Plantas Generadoras de Energía Eléctrica

2.2 Plantas de Generación Termoeléctrica

En México aproximadamente el 75% de la energía eléctrica se genera en plantas termoeléctricas, las cuales, como ya hemos visto, funcionan a través de la quema de algún combustible fósil para producir vapor de agua en una caldera; el vapor es elevado a una gran presión y llevado a una turbina que está conectada a un generador, y cuando éste gira convierte el movimiento en electricidad. Después de que el vapor pasa por la turbina, es llevado a una torre de enfriamiento, donde se condensa y se convierte de nueva cuenta en agua líquida. Al ocurrir esto, el agua se lleva de nuevo a la caldera para así repetir el proceso indefinidamente.

Cuando una planta puede utilizar indistintamente dos tipos de combustibles fósiles, se le denomina “Dual”, por ejemplo cuando puede consumir carbón ó gas natural.

Existen plantas termoeléctricas llamadas de “Ciclo Combinado”, las cuales utilizan principalmente gas natural como combustible (también en algunos casos gasóleo o carbón preparado). En ellas, los gases calientes obtenidos de la combustión del gas natural que pasaron por la turbina, se vuelven a utilizar introduciéndolos a otras calderas que generan vapor de agua para mover nuevamente una turbina y un segundo generador.

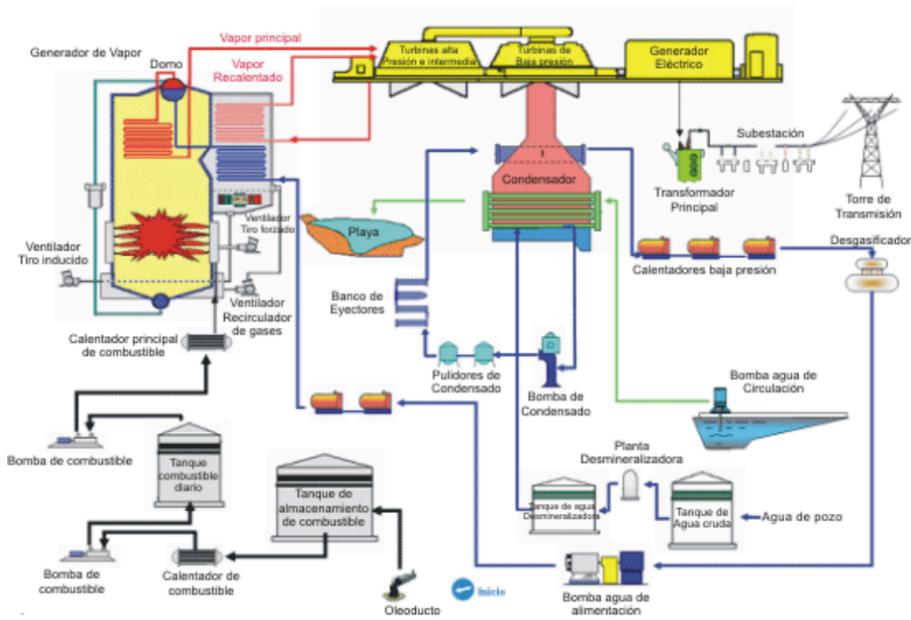


Imagen 2.2.1 – Esquema de una central termoeléctrica, (tomado de www.cfe.gob.mx).

Después de explicar todo lo anterior y entender el cómo funcionan este tipo de plantas, solo podría resaltar una duda, ¿cómo funciona el generador?, ó ¿cómo se produce la energía eléctrica después de que se hace mover la turbina?

En todos los casos anteriores, la turbina está unida por su eje al generador, es decir, cuando gira la turbina gira de la misma forma el generador. Este último contiene un rotor bobinado que gira dentro de un campo magnético estacionario con espiras (embobinado) de un largo y grueso cable. Cuando giran el eje de la turbina y el magneto que está dentro del generador, se produce una corriente de electricidad en el cable. Lo anterior se explica debido al fenómeno llamado “electromagnetismo”, que descrito de manera sencilla consiste en que cuando un cable o cualquier material conductor de electricidad se mueve a través de un campo magnético, cortando líneas de fuerza magnética, se produce una corriente de energía eléctrica en el cable o conductor empleado.

Dicho de otra forma, se podría decir que un generador es como un motor eléctrico, pero empleado al revés, pues en lugar de usar energía eléctrica para hacer girar el motor, el eje de la turbina hace girar el motor para producir electricidad.

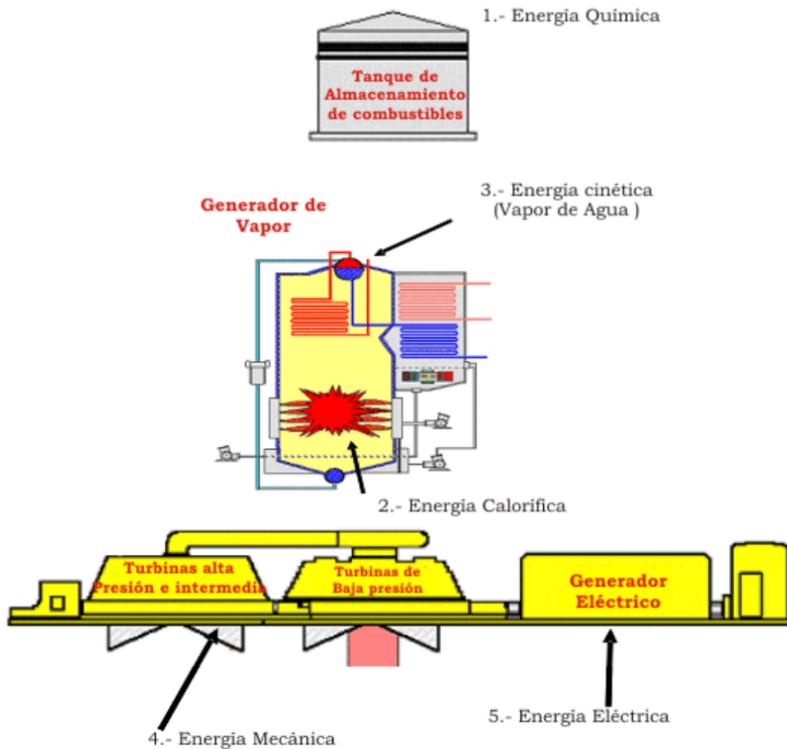


Imagen 2.2.2 -Centrales termoeléctricas tipo vapor: Estas centrales utilizan el poder calorífico de combustibles derivados del petróleo (combustóleo, diesel y gas natural), para calentar agua y producir vapor con temperaturas del orden de los 520°C y presiones entre 120 y 170 kg/cm², para impulsar las turbinas que giran a 3600 r.p.m. (revoluciones por minuto), (tomado de www.cfe.gob.mx).

Las centrales termoeléctricas son las más abundantes en México y en el mundo, ello se debe a que son las centrales más baratas de construir, especialmente las de carbón, y a la simplicidad de su construcción.

Su rendimiento es bajo comparado con el rendimiento ideal, y a pesar de haberse realizado con el tiempo mejoras a la eficiencia, ésta se mantiene entre un 30 y 40%, es decir, del total de la energía liberada en la combustión, solo del 30 al 40% se convierte en electricidad. Las centrales termoeléctricas de ciclo combinado por su parte, son más eficientes, ya que alcanzan un 50% de eficiencia, aumentando así la electricidad generada y con ello las ganancias (datos tomados de www.cfe.gob.mx y http://es.wikipedia.org/wiki/Central_termoel%C3%A9ctrica).

Otro inconveniente de las centrales termoeléctricas es que su uso de combustibles fósiles durante el proceso de generación, genera la emisión de gases de efecto invernadero y de lluvia ácida a la atmósfera junto con partículas en el aire (en el caso del carbón) que pueden contener metales pesados. Por último, otra desventaja importante, es que su rentabilidad se ve afectada a la disponibilidad de los recursos fósiles que emplea, pues como todos sabemos, son una fuente de energía finita.

2.3 CFE y las Centrales Termoeléctricas de México

El problema más serio que tiene la energía eléctrica es que no puede almacenarse, sino que debe ser transmitida y utilizada justo en el momento en que se genera. En México, como ya sabemos, la fuente de generación de energía eléctrica más importante y más utilizada, es la que se obtiene por medio de las plantas o centrales termoeléctricas. En cuanto se produce la electricidad en éstas plantas, una enorme red de cables tendidos e interconectados a lo largo y ancho del país, se encargan de hacerla llegar casi instantáneamente a todos los lugares de consumo, ya sea hogares, fábricas, oficinas, talleres, comercios, etc.

Ésta red de distribución también está interconectada con los otros tipos de plantas generadoras (hidroeléctricas, geotermoeléctricas, eoloeléctricas, nucleoeeléctricas, etc.), que aunque representan en conjunto un porcentaje menor al de las plantas termoeléctricas (alrededor del 25%), su funcionamiento también es importante para que la energía eléctrica llegue a todos los rincones del país.

Miles de trabajadores vigilan día y noche que no se produzcan fallas en este servicio, y cuando ocurren (debido a vientos, lluvias, rayos, entre otras causas), ellos mismos acuden a reparar las líneas para restablecer el suministro de energía. Para ello, hay centros de monitoreo, estratégicamente ubicados para mantener una vigilancia permanente en toda la red.

Todo este trabajo es realizado en México por la “Comisión Federal de Electricidad –CFE”, quien desde hace ya más de 50 años, genera, transmite, distribuye y comercializa la electricidad para todos los mexicanos. Su compromiso es ofrecer servicios de excelencia, garantizando altos índices de calidad en todos sus procesos, al nivel de las mejores empresas eléctricas del mundo.

La CFE proporciona un servicio a 25.9 millones de clientes en nuestro país, lo que nos dice que aproximadamente 80 millones de mexicanos cuentan con energía eléctrica en sus hogares (aproximadamente un 80% de la población total del país).

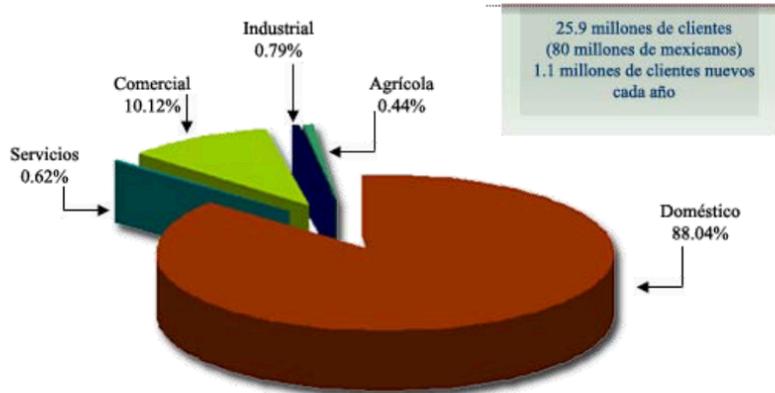


Imagen 2.3.1 – Sectores de Consumo de Electricidad en México, (tomado de www.cfe.gob.mx).

Para poder darle el servicio a tantos mexicanos, la CFE cuenta con 156 centrales generadoras de todos los tipos (también existen 21 centrales generadoras independientes, las cuales brindan servicio a fábricas e industrias del sector privado, en total, incluyendo éstas últimas serían 177 en todo el país.), predominando en cantidad las centrales termoeléctricas.

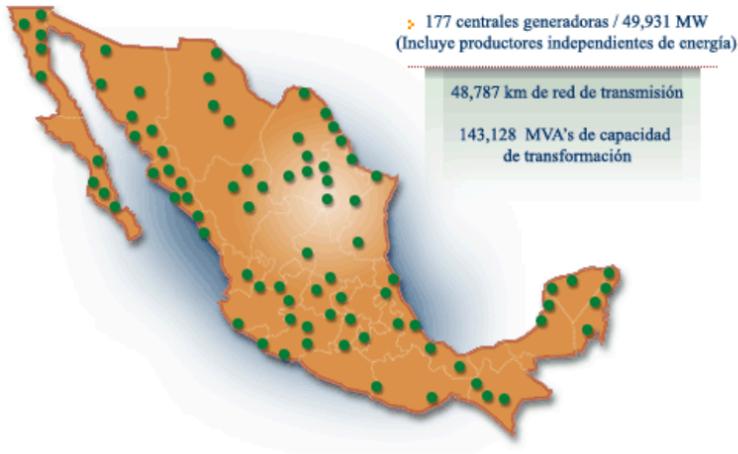


Imagen 2.3.2 – Centrales Generadoras de Electricidad en México, (tomado de www.cfe.gob.mx).

Estas 177 centrales generadoras de todo el país cuentan con cierta capacidad instalada dependiendo el tipo de generación con la que trabajan. En la siguiente tabla se desglosa la capacidad de todas estas centrales.

| Tipo de Generación | Capacidad efectiva en MW |
|--|--------------------------|
| Termoeléctrica | 22,404.69 |
| Hidroeléctrica | 11,054.90 |
| Carboeléctrica | 2,600.00 |
| Geotermoeléctrica | 964.50 |
| Eoloeléctrica | 85.48 |
| Nucleoeléctrica | 1,364.88 |
| Termoeléctrica (Productores independientes) | 11,456.90 |
| Total | 49,931.34 |

Cuadro 2.3.2 - Capacidad efectiva instalada por tipo de generación al mes de junio de 2008, (tomado de www.cfe.gob.mx).

Del cuadro anterior podemos observar que la suma de la capacidad efectiva de plantas termoeléctricas (incluyendo los productores independientes y las plantas carboeléctricas pues también utilizan el mismo principio de generación) es de 36 461.59 MW de un total de 49 931.34 MW, lo que representa un 73.023%, que es 3 cuartas partes del total de capacidad efectiva generada en nuestro país (ó un 75%, como antes se había mencionado).

Las principales centrales termoeléctricas de México (sin incluir las de productores independientes) y con las cuales va a trabajar el Sistema de Gestión de Base de Datos para el Sistema LAPEM –TERMO® y cuyo empleo queda preestablecido como requerimiento de CFE para este proyecto, son las siguientes:

CENTRALES TERMOELÉCTRICAS DEL SISTEMA “LAPEM-TERMO®”

1. Central Termoeléctrica de Altamira (Estación de Colonias, municipio de Altamira, Tamaulipas, México).
2. Central Termoeléctrica Carbón II (Coahuila, México).
3. Central Termoeléctrica Francisco Villa (Delicias, Chihuahua, México).
4. Central Termoeléctrica Guadalupe Victoria (Villa Juarez, municipio de Lerdo, Durango, México).
5. Central Termoeléctrica Guaymas II (Guaymas, Sonora, México).
6. Central Termoeléctrica Lerma (Campeche, México).
7. Central Termoeléctrica Manzanillo I (Colima, México).
8. Central Termoeléctrica Manzanillo II (Colima, México).
9. Central Termoeléctrica Mazatlán II (Sinaloa, México).
10. Central Termoeléctrica Mérida II (Yucatán, México).
11. Central Termoeléctrica Petacalco (Km. 28 Carretera Lázaro Cardenas-Zihuatanejo, municipio La Union, Guerrero, México).
12. Central Termoeléctrica Poza Rica (Tihuatlan, Veracruz, México).
13. Central Termoeléctrica Puerto Libertad (Puerto Libertad, Pitiquito, Sonora, México).
14. Central Termoeléctrica Punta Prieta II (La Paz, Baja California, México).
15. Central Termoeléctrica Rio Bravo (Coahuila, México).
16. Central Termoeléctrica Rio Escondido (Rio Escondido, Tamaulipas, México).
17. Central Termoeléctrica Rosarito (Rosarito, Baja California Norte, México).
18. Central Termoeléctrica Salamanca (Salamanca, Guanajuato, México).
19. Central Termoeléctrica Samalayuca (Chihuahua, México).
20. Central Termoeléctrica Topolobampo II (Los Mochis, Sinaloa, México).
21. Central Termoeléctrica Tula (Tula de Allende, Hidalgo, México).
22. Central Termoeléctrica Tuxpan (Veracruz, México).
23. Central Termoeléctrica Valladolid (Yucatán, México).
24. Central Termoeléctrica Valle de México (Estado de México, México).
25. Central Termoeléctrica Villa de Reyes (San Luis Potosí, México).

Cuadro 2.3.3 – Centrales Termoeléctricas del Sistema LAPEM-TERMO®.

Todas las centrales anteriores, distribuidas a todo lo largo del territorio nacional representan casi el 70% de la capacidad efectiva instalada de las centrales tipo termoeléctricas (el restante 30% corresponde a las centrales del sector privado o productores independientes, y a las centrales termoeléctricas que la CFE no incluyo para este proyecto). Por ende, resulta obvia la importancia que tiene el correcto funcionamiento de estas centrales para el día a día de todos los habitantes del territorio nacional, tanto en sus hogares, oficinas, fábricas, talleres y demás lugares de trabajo.

Las centrales del cuadro anterior están distribuidas en México como ilustra la siguiente imagen:



Imagen 2.3.3 – Centrales Termoeléctricas del Sistema LAPEM-TERMO®.

Aparte de las centrales anteriores, México también cuenta con otras centrales termoeléctricas que por decisiones que atañen a CFE no entraron dentro de la lista para el proyecto LAPEM – TERMO®, algunas de las cuales son las siguientes:

**CENTRALES TERMOELÉCTRICAS NO INCLUIDAS EN EL SISTEMA
“LAPEM-TERMO®”**

- 1. Central Termoeléctrica Nachi-Cocom, Mérida, Yucatán, México.**
- 2. Central Termoeléctrica Guaymas I, Guaymas, Sonora, México.**
- 3. Central Termoeléctrica La Laguna, Gómez Palacio, Durango, México.**
- 4. Central Termoeléctrica Monterrey, San Nicolás de los Garza, Nuevo León, México.**
- 5. Central Termoeléctrica San Jerónimo, Monterrey, Nuevo León, México.**

Cuadro 2.3.4 – Centrales Termoeléctricas no incluidas en el Sistema “LAPEM-TERMO®”

(tomado de www.cfe.gob.mx).

2.4 Conceptos Fundamentales de Base de Datos

Desde que el hombre invento la escritura empezó a surgir la necesidad de encontrar una forma de almacenar información. La primera manera que encontró fue con el labrado de la piedra y ya después con la invención del papel se fue haciendo más fácil el guardar datos relevantes. Pero todo esto cambio con la invención de la computadora personal, pues ahora la información comenzó a guardarse en dispositivos de almacenamiento. A principios de los años 80's los sistemas de cómputo utilizaban un medio distinto para almacenar información: “los archivos”. Éstos se definen como un conjunto de información que se almacena en algún medio de escritura que permite ser leído o accedido por una computadora. Los archivos brindan una manera de facilitar la organización de recursos usados para almacenar información en una o varias computadoras.

Los sistemas de cómputo que se basan en el empleo de archivos para almacenar información responden a las características de los llamados Sistemas Orientados a Procesos. En ellos se pone un énfasis muy importante al tratamiento que tienen los datos, los cuales son almacenados en archivos diseñados específicamente para la aplicación que va a hacer uso de ellos. Pero obviamente estos sistemas fueron desplazados por las desventajas que presentaban, entre ellas destacan el alto consumo de espacio en memoria secundaria y el aumento en los tiempos de los procesos (por la repetición de operaciones iguales sobre distintos archivos).

Lo anterior nos dice que el administrar y almacenar información en un sistema de manejo de archivos tiene los siguientes inconvenientes:

- Redundancia e inconsistencia de los datos.
- Dificultad en el acceso a los datos.
- Aislamiento de los datos.
- Problemas de integridad de los datos.

Ante todo esto surgió la necesidad de encontrar una mejor manera de almacenar una gran cantidad de información, que fuera más rápida y eficiente para así poder posteriormente consultar los datos y sobre todo actualizarlos.

El concepto de base de datos tal y como lo conocemos hoy en día, se empezó a definir más fuertemente en la década de los años setenta, cuando Edgar Frank Codd (23 de agosto de 1923 – 18 de abril de 2003), escribió un artículo llamado "A Relational Model of Data for Large Shared Data Banks" ("Un modelo relacional de datos para grandes bancos de datos compartidos"), en el cual definió el modelo relacional y publicó una serie de reglas para la evaluación de administradores de sistemas de datos relacionales. Para muchas personas este artículo marcó el nacimiento de las bases de datos relacionales.

Tomando como base los aportes del artículo de Codd, el multimillonario Larry Ellison comenzó un nuevo proyecto, que tenía el propósito de elaborar el primer software que administrara el almacenamiento de grandes cantidades de datos, este proyecto terminó con el desarrollo de la base de datos Oracle, el cual es un sistema de administración de base de datos, que se destaca por sus transacciones, estabilidad, escalabilidad y multiplataforma; y que hoy en día tiene una presencia comercial importante con versiones que se actualizan constantemente.

En un principio no se utilizó el modelo relacional ya que éste presentaba inconvenientes por el rendimiento, y por ello no podían ser competitivas con las bases de datos jerárquicas y de red, que también estaban empezando a tomar fuerza por aquellos años. Toda ésta tendencia cambio por un proyecto de la gigante International Business Machines – IBM, la cual desarrolló técnicas para la construcción de un sistema de bases de datos relacionales eficientes, el cual fue llamado "System R".

De ahí en adelante el desarrollo de nuevos sistemas para la administración de base de datos empezó a surgir de todos lados, surgiendo de distintos tipos pero siendo el relacional el más eficiente y el más usado.

2.4.1 Concepto de Base de Datos

Existen muchas definiciones de Base de Datos, algunas de ellas son más completas que otras, pero siempre se observa una serie de coincidencias entre ellas, y en una que otra se omite aspectos importantes que otra de ellas tiene.

Una colección aleatoria de datos no debe considerarse propiamente una base de datos, sino que debe ser un conjunto de datos lógicamente coherentes, que representen algún aspecto de la vida real y cuyas modificaciones se vean reflejadas en los datos que almacena. Se diseña, se desarrolla, se implementa y se carga de datos para un propósito específico. Está dirigida a un grupo de usuarios en específico y tiene un conjunto de ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

Por ello podemos definir a una **Base de Datos** como una colección de datos integrados, es decir, relacionados entre sí, almacenados en memoria secundaria (no volátil) y con una redundancia controlada. Los datos que almacena la base de datos que son compartidos por los usuarios a quien está dirigida, deben de mantenerse independientes de las aplicaciones, y su definición (estructura de la base de datos) única y almacenada junto con los datos, se debe de basar en un modelo de datos, el cual debe de permitir captar las interrelaciones y las restricciones existentes en el mundo real, donde los procedimientos de actualización y recuperación, comunes y bien determinados facilitarán la seguridad de los datos de la base.

2.4.2 Sistema de Gestión de Base de Datos – SGBD

Un Data Base Management System –DBMS (ó Sistema de Gestión de Base de Datos – SGBD) consiste en un conjunto de datos relacionados entre sí y un conjunto de aplicaciones que sirven para poder acceder a esos datos. Dicho de otra manera, un DBMS es un software encargado de servir como interfaz entre la base de datos, las aplicaciones que la utilizan y los usuarios.

El principal objetivo de un DBMS es el de crear un entorno para administrar de manera sencilla, ordenada y segura la información de la base de datos.

Un DBMS nos ofrece numerosas ventajas al momento de manejar los datos, entre las cuales podemos destacar las siguientes:

Evita:

- La dificultad para tener acceso a los datos.
- La redundancia e inconsistencia de información.
- Aislamiento de datos.
- Problemas de seguridad.
- Problemas de integridad (limitantes de consistencia).

Otras ventajas:

- Conseguir la completitud de información.
- Permite usuarios múltiples.
- Mantener la independencia de datos.
- Lograr la abstracción de datos.
- Soportar vistas de datos.
- Brindar facilidades para el desarrollo y mantenimiento de aplicaciones.
- Mejorar la eficiencia en las diferentes funciones que soportan el acceso a los datos.

También cuenta con una serie de desventajas que se mencionan a continuación:

- Costo del manejador y otros costos imprevistos.
- Necesidad de personal capacitado.
- Recursos de equipo de cómputo.
- Vulnerabilidad.

2.4.3 Usuarios de la Base de Datos

La base de datos y el DBMS interactúan entre sí con varios tipos de usuarios, a continuación se definen cada uno de ellos.

- **Administrador de la Base de Datos** – (Data Base Administrator-DBA) :
Monitorea el rendimiento del DBMS y administra su uso, se encargan de actividades como la creación y prueba de respaldos, verifica la integridad de los datos, define e implementa controles de acceso a los datos (seguridad), entre otras.

- **Diseñadores de la Base de Datos:** Se encargan del diseño lógico y físico de la base de datos además de su construcción, a partir de las necesidades de los usuarios finales.
- **Analistas de la Base de Datos:** Formulan los requerimientos de los usuarios y determinan las aplicaciones que debe de proveer el sistema de base de datos.
- **Programadores de la Base de Datos:** Interactúan con el DBMS implementando las aplicaciones y consultas (ó también denominados queries) que debe de proveer el sistema de base de datos.
- **Usuarios Finales de la Base de Datos:** Interactúan con el sistema de base de datos y con el DBMS a través de una interfaz gráfica para utilizar las aplicaciones que éste provee.

2.4.4 Arquitectura de una Base de Datos

Una característica muy importante de los DBMS y de las bases de datos en general, es que deben proporcionar a los usuarios finales una visión abstracta de los datos cuando éstos sean requeridos. Todo esto nos lleva a una “arquitectura de base de datos” que debe poseer el sistema para poder cumplir con esta característica.

La arquitectura estándar de una base de datos cuenta con tres niveles básicos: nivel interno (también llamado físico o inferior), nivel conceptual (ó nivel medio) y nivel externo (ó de visión); cada uno de los cuales corresponde a una vista diferente.

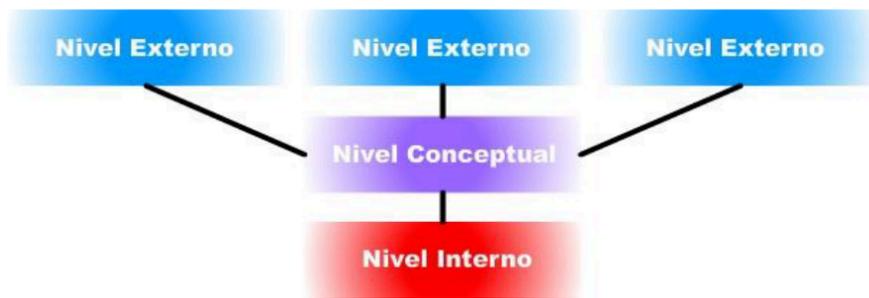


Imagen 2.4.4.1: Arquitectura de una Base de Datos

- **Nivel Interno:** En este nivel se brinda la visión física de los datos almacenados dentro de la base de datos, es decir, de los dispositivos de almacenamiento como los discos, las direcciones físicas, los índices, los punteros, etc. Los diseñadores de la base de datos física tienen la responsabilidad de este nivel, los cuales deciden que dispositivos físicos almacenarán los datos, que métodos de acceso se utilizarán para buscar y actualizar los datos, etc.
- **Nivel Conceptual:** En este nivel se lleva a cabo el diseño conceptual de la base de datos, el cual debe incluir una lista de las necesidades de los usuarios y el tipo de datos que se requiere almacenar para satisfacer dichas necesidades. El resultado de este nivel es el llamado "esquema ó diagrama conceptual", el cual es una sencilla descripción de todos los datos a utilizar y las relaciones que existen entre ellos.
- **Nivel Externo:** Este nivel consiste en todas las vistas que el usuario final tiene de la base de datos. Para cada grupo de usuarios se tendrá una vista diferente, y la unión de todas las vistas conformarán el nivel externo.

***Datos de la arquitectura tomados del artículo "Fundamentos de Base de Datos" realizado por Violeta Chang Camacho.

2.4.5 Modelos de Datos

Todos los datos que queremos almacenar dentro de una base de datos se deben de organizar atendiendo a una necesidad, como mencionamos anteriormente, deben de ser lógicamente coherentes y que representen una solución a un problema de la vida real.

Cuando se intenta representar un problema en particular de la vida real surgen siempre en el entorno tres conceptos básicos: modelos de datos, esquema y diagrama.

Un modelo de datos es una herramienta conceptual que permite describir las características, propiedades y el cómo se encuentran estructurados los datos de una realidad o problema que se desea especificar.

Un modelo de datos se compone de tres elementos:

- **Estructuras:** Permiten definir las características esenciales de los datos.
- **Restricciones:** Permiten describir las reglas que en todo momento se deben de cumplir en la base de datos. Pueden ser inherentes, implícitas y explícitas.
- **Operaciones:** Indican el cómo pueden ser utilizados los datos para satisfacer requerimientos.

Muchas veces se confunde el término diagrama con el de esquema, y se utilizan tanto uno como el otro de igual manera, pero hay una diferencia importante entre ellos que debemos mencionar.

Un esquema es una representación de un problema en particular utilizando un modelo de datos. Mientras que un diagrama es la representación gráfica de un esquema, es decir, un diagrama es la parte gráfica de un esquema.

2.4.6 Clasificación de los Modelos de Datos

La clasificación más usada para los modelos de datos se ha ido definiendo conforme a la propia evolución de las bases de datos y de los sistemas comerciales para manejar las mismas (los ya mencionados DBMS). Ésta clasificación se basa en tres diferentes tipos de modelos de datos:

- **Modelos de Alto Nivel:**¹ También son llamados “Conceptuales” ó “Semánticos”. Éstos modelos permiten representar problemas del mundo real a través de estructuras que visualizan a los datos de la forma más cercana a como el hombre los percibe. Ejemplos de este tipo de modelos son:
 - * Modelo Entidad-Relación
 - * Modelo Orientado por Objetos
 - * Modelo Lógico
 - * Modelo Infológico
- **Modelos de Implementación, Lógicos ó Tradicionales:** Este tipo de modelos nos permiten representar los problemas de la vida real a través de estructuras que pueden ser entendidas por usuarios pero que son más cercanas a la forma en la cual se almacenan en el computador. Ejemplos de éste tipo de modelos son:
 - * Los modelos de datos implementados en los DBMS's.
 - * Modelo Jerárquico.
 - * Modelo de Red.
 - * Modelo Relacional.

¹ Ver anexo #2

- **Modelos de Bajo Nivel:** También son llamados “Modelos Físicos”. Estos modelos ya son más abstractos, pues nos proporcionan una forma de representar el cómo están almacenados los datos en memoria secundaria. Cada DBMS maneja un modelo de bajo nivel diferente, lo que aumenta más la dificultad de los modelos físicos.

2.4.7 Modelo Entidad - Relación

El modelo con el cual se va a trabajar para la solución del planteamiento del problema de ésta tesis es el “Modelo Entidad-Relación”. Este modelo se empleará como punto de partida para el capítulo tres del presente documento; por ello es prudente empezar a conocerlo y darle un énfasis sobre los demás.

El Modelo Entidad-Relación (*E-R* "Entity Relationship") es el modelo conceptual más utilizado en el mundo actual para el diseño de bases de datos. Es una herramienta de apoyo muy importante para el diseño de sistemas de información.

El Modelo Entidad-Relación se basa en una percepción del mundo real que consiste en una colección de objetos llamados entidades, sus propiedades y las relaciones entre éstos objetos; los cuales dan origen a un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. En un principio este modelo solo incluía los conceptos de entidad, atributo y relación, pero más tarde se añadieron otros conceptos como el de atributo compuesto y las jerarquías de generalización.

Este modelo fue propuesto por Peter Chen en 1976, quien tenía la intención de visualizar los objetos que pertenecen a la Base de Datos como **entidades**, las cuales tienen ciertos **atributos** y se relacionan entre ellas mediante **relaciones** (tomado de http://es.wikipedia.org/wiki/Peter_Chen).

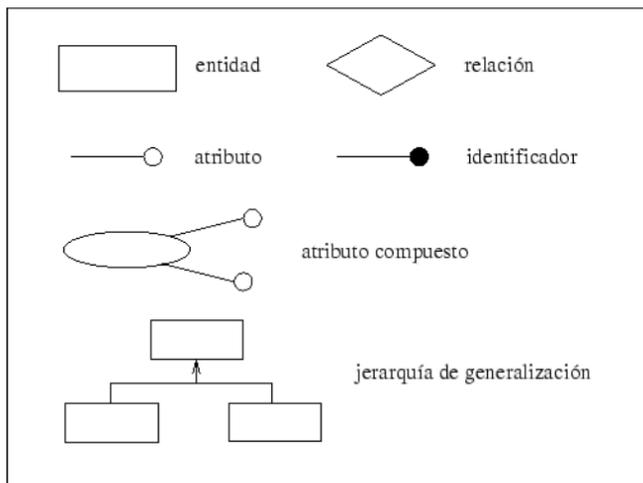


Imagen 2.4.7.1 - Conceptos del Modelo Entidad-Relación.

- **Entidad:** Una entidad se refiere a cualquier tipo de objeto o concepto sobre el cual se recoge información; por ejemplo: empleados, coches, clientes, empresas, productos, etc. Se representan gráficamente mediante rectángulos y su nombre aparece en su interior (el nombre de una entidad solo puede aparecer una vez en el modelo entidad-relación). Existen dos tipos de entidades:
 - * **Entidad Débil:** Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad (entidad fuerte).
 - * **Entidad Fuerte:** Una entidad fuerte es una entidad cuya existencia depende de sí misma y no de ninguna otra entidad.
- **Relación:** Una relación se refiere a una correspondencia o asociación entre dos o más entidades. Se representan gráficamente mediante rombos con su nombre en su interior.
- **Atributo:** Un atributo se refiere a una característica de interés o un hecho sobre una entidad o una relación, es decir, representan las propiedades básicas de las entidades y las relaciones. Gráficamente se representan mediante círculos u óvalos (que cuelgan de las entidades o relaciones) con

su nombre en su interior. Los atributos pueden ser simples o compuestos; los **atributos simples** son aquellos que tienen un solo componente, mientras los **atributos compuestos** son aquellos que tienen varios componentes, cada uno con un propio significado.

- **Identificador:** Toda entidad debe tener al menos un identificador, mientras que las relaciones carecen de ellos. Un identificador de una entidad es un atributo que determina de modo único cada ocurrencia de esa entidad. Debe cumplir con una regla básica: **“No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador”**.
- **Jerarquía de generalización:** La jerarquía de generalización se utiliza para agrupar un cierto número de entidades que tienen el mismo grado de ocurrencia. Es decir, una entidad E es una generalización de un grupo de entidades $E_1, E_2, E_3 \dots E_n$, si cada ocurrencia de cada una de esas entidades es también una ocurrencia de E.

Una ventaja importante del modelo entidad-relación es que mediante una serie de procedimientos se puede pasar de este modelo a otros, como por ejemplo el **modelo relacional**.

El modelo entidad-relación no es la única técnica que existe para el modelado de datos utilizando diagramas, pero sí es la más utilizada. Para describir más claramente en qué consiste ésta técnica, a continuación se mencionan una serie de pasos para poder realizarla:

1. Se parte de una descripción textual del problema ó sistema de información a realizar, es decir, definir los requisitos.
2. Se hace una lista de los sustantivos y verbos que aparecen en dicha descripción.
3. Los sustantivos son posibles entidades o atributos.
4. Los verbos son posibles relaciones.
5. Analizando las frases de los requisitos se determina la cardinalidad de las relaciones (uno a uno, uno a muchos, muchos a muchos).

6. Se elabora el diagrama(s) entidad-relación.
7. Se completa el modelo con una lista de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

Al observar los pasos anteriores se puede observar que ésta técnica es muy rudimentaria, por lo que se necesita cierto entrenamiento y experiencia para lograr buenos modelos de datos.

Todo el proceso de modelado no termina después de realizar el modelo entidad-relación, sino que son necesarias otras técnicas para lograr un modelo que se pueda implementar directamente en una base de datos. Algunas de éstas técnicas son:

- Transformación de relaciones múltiples a binarias.
- Normalización (algunas relaciones pueden transformarse en atributos y viceversa).
- **Conversión en tablas (en el caso de utilizar una Base de Datos Relacional).**
- Entre otras no menos importantes.

2.4.8 Modelo de Datos Relacional

El Modelo Relacional es un modelo lógico en el cual se basan la mayoría de los DBMS que se usan hoy en día. Surgió en 1970 cuando E.F. Codd lo dio a conocer. En aquellos días, se utilizaba un enfoque distinto para la estructura de las bases de datos, ya que se utilizaban punteros físicos (direcciones de disco) para poder relacionar registros de distintos ficheros. Por ellos es fácil imaginar lo vulnerable que eran aquellas bases de datos a los cambios en el entorno físico de los sistemas; los cuales se basaban en el modelo de red y en el modelo jerárquico (mencionados anteriormente), que representan la llamada primera generación de los DBMS.

El Modelo Relacional representa la segunda generación de los DBMS. En él todos **los datos están estructurados a nivel lógico en forma de tablas conformadas por filas y columnas** (aunque a nivel físico pueden tener una estructura completamente diferente).

La base del Modelo Relacional se centra en el concepto matemático de relación, que gráficamente se representa mediante una tabla. Codd utilizó una terminología de las matemáticas como base: “la teoría de conjuntos” y la “lógica de predicados”. **Una relación es una tabla conformada por columnas y filas**, donde el DBMS solo necesita que el usuario perciba a la base de datos como un conjunto de tablas relacionadas entre sí. **Un atributo es el nombre de una columna de una relación**. En el Modelo Relacional las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos. Una relación se representa gráficamente como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas corresponden a los campos o atributos de esos registros.

Un dominio es el conjunto de valores legales de uno o varios atributos. Los dominios representan una característica muy importante del modelo relacional. Cada atributo de una Base de Datos Modelo Relacional se define sobre un dominio, pudiendo haber varios atributos definidos sobre el mismo dominio. El concepto de dominio es importante porque permite que el usuario defina, en un lugar común, el significado y la fuente de los valores que los atributos pueden tomar. Esto hace que haya más información disponible para el sistema cuando éste va a ejecutar una operación relacional, de modo que las operaciones que son semánticamente incorrectas, se pueden evitar.

El Modelo Relacional cuenta con varios elementos, aparte de los ya mencionados anteriormente, los cuales son los siguientes:

- **Tupla:** Una tupla es una fila de una relación. Los elementos de una relación son las tuplas o filas de la tabla.
- **Grado de la Relación:** El grado de una relación es el número de atributos que contiene.
- **Cardinalidad de una Relación:** La cardinalidad de una relación es el número de tuplas que contiene. Puede variar debido a la constante inserción de relaciones y al borrado de tuplas.

Este modelo tiene una sencilla estructura lógica pero con un fundamento teórico importante, del cual carecen los modelos de la primera generación. El modelo relacional, como todo modelo de datos, tiene que ver con tres aspectos de los datos:

- Estructura de los datos.
- Integridad de los datos.
- Manejo de los datos.

Se dice que un Modelo Relacional bien realizado da origen a una base de datos relacional. **Una base de datos relacional se define como un conjunto de relaciones normalizadas.**

2.4.8.1 Propiedades de las Relaciones del Modelo Relacional

Las relaciones, base del Modelo Relacional, cuentan con una serie de propiedades o características, las cuales debemos de conocer para poder trabajar con este modelo. A continuación se mencionan las más importantes:

- Cada relación tiene un sólo nombre y éste es distinto del nombre de todas las demás.
- Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor. Cuando ocurre esto se dice que las relaciones están *normalizadas*.
- No existen dos atributos con el mismo nombre.
- El orden de los atributos no importa.
- Cada tupla es distinta de las demás, es decir, no hay tuplas duplicadas.
- El orden de las tuplas no importa, es decir, las tuplas no están ordenadas.

2.4.8.2 Tipos de Relaciones del Modelo Relacional

Dentro de un DBMS relacional pueden existir varios **tipos de relaciones** (aunque no todos manejan todos los tipos), las cuales son:

- **Relaciones Base:** Son las propias del diseño de la base de datos.
- **Relaciones de Vista:** Son relaciones virtuales que se crean dependiendo del tipo de usuario que haga uso de la base de datos.
- **Relaciones Instantáneas:** Son relaciones creadas con cierta regularidad pero que no pertenecen al diseño de la base de datos.

- **Relaciones de resultado de consulta:** Estas relaciones surgen cuando realizamos una consulta.
- **Relaciones temporales:** Estas relaciones las creamos al entrar al sistema y se borran al momento en que salimos.

2.4.8.3 Llaves del Modelo Relacional

Debido a que en una relación no existen tuplas repetidas, éstas se pueden distinguir unas de otras, es decir, se pueden identificar de modo único. La forma de identificarlas es mediante los valores de sus atributos. En este entorno entra el concepto de llaves. Existen varios tipos:

- **Superllave:** Atributo ó conjunto de atributos mediante los cuales distingo una entidad ó relación.
- **Llave Candidato:** Es una superllave mínima. Es decir, si tenemos el atributo o conjunto de atributos "A" de la relación "R", A es llave candidato de R si y solo si nunca hay dos tuplas en la relación R con el mismo valor de A.
- **Llave Primaria:** Llave candidato elegida por el diseñador de la base de datos. También es llamada "Primary Key - PK".
- **Llave Foránea:** Es un atributo o conjunto de atributos de una relación cuyos valores coinciden con los valores de la llave primaria de alguna otra relación. Representan relaciones entre datos. También es llamada "Foreign Key - FK".

2.4.8.4 Esquema de una relación del Modelo Relacional

Para poder representar el esquema de una base de datos relacional se debe conocer el nombre de sus relaciones, el de sus atributos, sus llaves primarias y sus llaves foráneas.

En el esquema aparecen los nombres de las relaciones seguidos de los nombres de sus atributos encerrados entre llaves. Las llaves primarias son los atributos subrayados. Las llaves foráneas son los atributos que aparecen doblemente subrayados.

En todos los casos los atributos de la relación aparecen acompañados del tipo de variable empleada para su almacenamiento, indicando su tamaño (el tipo de variable puede variar dependiendo del DBMS que se utilice). Ejemplo:

Jugador {ClaveJugador: CHAR(5), NombreJugador: VARCHAR(30), EdadJugador: NUMBER(2), ClaveEquipo: CHAR(5)}

Conociendo las relaciones y sus atributos representados de la forma anterior, podemos proceder a la creación de las tablas, las cuales adoptarán el nombre de la relación como su propio nombre, y el nombre de sus atributos serán el nombre de sus columnas. Una vez creada la tabla se puede proceder a insertar las tuplas, que es en sí, la información que contendrá la tabla. Retomando el ejemplo anterior, su tabla quedaría así:

| JUGADOR | | | |
|-------------------|------------------|-------------|------------------|
| ClaveJugador (PK) | NombreJugador | EdadJugador | ClaveEquipo (FK) |
| A1001 | Salvador Cabañas | 27 | BAG12 |
| B1256 | Rafael Márquez | 28 | AGD57 |

Cuadro 2.4.8.4.1 – Tabla de un ejemplo del Modelo Relacional

2.4.8.5 Reglas de Integridad

Una vez definida la estructura de datos del modelo relacional podemos pasar a analizar las llamadas reglas de integridad, en las cuales, los datos almacenados en dichas estructuras, las deben de cumplir para garantizar que son correctos.

Antes de empezar a mencionar dichas reglas, debemos conocer el concepto de “nulo”. Un nulo se refiere a cuando en una tupla un atributo es desconocido. Un nulo no representa el valor cero ni una cadena vacía, sino que estos valores tienen significado. El nulo implica ausencia de información, ya sea porque al insertar la tupla se desconocía el valor del atributo, o porque para dicha tupla el valor de ese atributo en particular no tiene sentido.

Cabe aclarar que no todos los DBMS aceptan valores nulos, pero en caso de que sí, éstos deben de tratarse de una forma diferente a los demás.

A la primera regla de integridad se le conoce como “**Regla de Integridad de Entidades**”. Ésta nos dice que ninguno de los atributos que componen la llave primaria de una relación puede ser nulo.

A la segunda regla de integridad se le conoce como “**Regla de Integridad Referencial**”. Ésta nos dice que si en una relación existe una llave foránea, su valor debe de coincidir con el valor de la llave primaria a la que hace referencia. Un complemento para esta regla es que no deben de existir llaves foráneas con valores nulos, ya que con ello se perdería inmediatamente la referencia.

2.4.8.6 Manejo de los Datos

La tercera parte de un modelo de datos relacional es la manipulación de los datos. Existen varios lenguajes empleados por diferentes DBMS relacionales para manejar las relaciones. Algunos de ellos son **procedurales**, lo que quiere decir que el usuario dice al sistema exactamente como debe de manipular a los datos. Por otra parte, otros son **no procedurales**, en los cuales el usuario dice que datos necesita en lugar de decir cómo deben de obtenerse.

En este apartado se presentan el Álgebra Relacional y el Cálculo Relacional, los cuales fueron definidos por Codd como base de los lenguajes relacionales. Se puede decir que el Álgebra Relacional es un lenguaje procedural de alto nivel, mientras que el Cálculo Relacional es un lenguaje no procedural. Sin embargo ambos lenguajes son equivalentes, ya que para cada expresión de álgebra se puede encontrar una expresión equivalente para el cálculo, y viceversa. Ambos también se utilizan para medir la potencia de los lenguajes relacionales; ya que si un lenguaje permite obtener cualquier relación que se pueda derivar mediante algunos de éstos dos lenguajes, se dice que es relacionalmente completo.

Tanto el Álgebra como el Cálculo Relacional son lenguajes formales no muy “amigables”, pero se deben de estudiar porque sirven para ilustrar las operaciones básicas que todo lenguaje de manejo de datos debe de ofrecer. Además han servido como base para el desarrollo de otros lenguajes relacionales de manejo de datos de más alto nivel.

2.4.9 Paso del Modelo Entidad-Relación al Modelo Relacional

Como ya se había mencionado anteriormente, mediante una serie de pasos ó reglas es posible pasar del Modelo Entidad – Relación al Modelo Relacional. Este procedimiento es muy importante, ya que brinda una metodología para pasar de un esquema a otro, lo que es muy útil porque para comenzar cualquier diseño para una base de datos es muy práctico empezar con el Modelo Entidad-Relación; mientras que para comenzar la implementación es más sencillo a partir del Modelo Relacional. Por ello combinar estas dos técnicas resulta una metodología muy poderosa y eficaz.

Las reglas necesarias para pasar de un modelo a otro son los siguientes:

1. Toda entidad fuerte se convierte en una relación.
2. Todo tipo de relación muchos a muchos se transforma en una relación.
3. Dentro de los tipos de relación uno a muchos tenemos dos casos:
 - a) Cuando no contiene atributos descriptivos se realiza una propagación de la llave, del uno al muchos.
 - b) Cuando tiene atributos descriptivos se crea una relación.
4. En las relaciones uno a uno se hace una propagación de la llave dependiendo de la entidad de mayor interés.
5. En entidades débiles sus atributos son absorbidos por la relación de la entidad fuerte, desapareciendo la entidad débil.

2.5 Base de Datos en Access

Access es un programa desarrollado por la empresa Microsoft. Está enfocado para ser un programa de gestión de base de datos relacionales, es decir, para ser un DBMS relacional. Es un programa muy popular en las pequeñas y medianas empresas, en las que sus bases de datos son relativamente pequeñas y no requieren de muchas transacciones. Otra ventaja de este DBMS, es que permite integrarse con las demás herramientas de Microsoft, en particular las de la suite de Office, que como todos sabemos, es de las más usadas en todo el mundo.

Existen varias versiones de este programa, siendo la última Microsoft Access 2007, versión que es muy llamativa por la nueva y mejorada interfaz que maneja, así como por conservar la sencillez de su uso, aspecto que beneficia a usuarios que no conocen mucho sobre el diseño de las bases de datos. Por ello muchas veces se ha dicho que Access es un DBMS visual. Incluye aplicaciones de bases de datos listas para usarse, pues están preparadas para tareas muy comunes como archivar documentación, generar reportes, ordenar libros, etc.

Sin embargo, Access contiene herramientas de diseño y programación reservadas a los usuarios de mayor experiencia. Por ejemplo, cuenta con la opción de trabajar directamente una consulta, actualización o borrado en código SQL. También nos permite obtener datos de otras aplicaciones, como Microsoft Excel, y además nos permite exportar nuestra base de datos a otros manejadores como Microsoft SQL Server.

Access es una aplicación de un sistema de gestión de base de datos que utiliza por defecto el motor de base de datos Microsoft Jet. Esto quiere decir que Access en realidad es una interfaz preparada para gestionar cualquier base de datos, no sólo una base de datos Access. Los sistemas Windows disponen de ODBC, que es una pasarela para poder acceder a cualquier base de datos. A través de ODBC podemos conectarnos a cualquier base de datos utilizando un lenguaje de programación o una aplicación como Access. Por ejemplo, podríamos conectar con una base de datos SQL Server o MySQL desde Access.

Lo que normalmente llamamos una base de datos Access, en realidad es una base de datos Jet, que está incluida en cualquier sistema Microsoft. Con Access podemos crear este tipo de bases de datos, que generalmente se guardan en archivos con extensión “.mdb”, y gestionarlas a través de la aplicación. Access dispone de numerosos métodos y asistentes para definir cualquier elemento de la base de datos, como las tablas, las consultas, los formularios, etc. y realizar la administración y gestión de los datos.

2.6 Base de Datos en SQL Server

De igual manera que Access, SQL Server es un programa desarrollado por la empresa de software Microsoft. Es un sistema de gestión de base de datos relacionales basado en el lenguaje “Transact-SQL” y también en “Sybase IQ”. Es capaz de poner a disposición de los usuarios grandes cantidades de información de manera simultánea, representando la respuesta de Microsoft a otros DBMS muy poderosos, como Oracle, MySQL, etc.

La estrategia que utiliza Microsoft con SQL Server, es que ésta sea la herramienta de base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación muy rápido y sencillo para programadores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.

Sus características son:

- Cuenta con soporte de transacciones.
- Tiene escalabilidad, estabilidad y seguridad.
- Maneja procedimientos almacenados.
- Maneja un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor.
- Permite administrar información de otros servidores de datos.

SQL Server ofrece herramientas integradas de desarrollo para el motor de base de datos, extracción, transformación y carga de datos, minería de datos, OLAP e informes que están directamente integrados con Microsoft Visual Studio (otra herramienta de Microsoft para el desarrollo de aplicaciones y sistemas de información) para ofrecer capacidades de desarrollo de aplicación de extremo a extremo.

La última versión de este DBMS es Microsoft SQL Server 2008, la cual ofrece una gama de mejoras y novedades orientadas a la administración integral de bases de datos. Entre ellas, destacan el soporte para la gestión basada en políticas, capacidades de auditoría, almacenamiento de datos a gran escala, datos geoespaciales, y servicios de análisis y reporting avanzados.

2.7 Desarrollo en plataforma .NET

2.7.1 Conexión e interoperabilidad de bases de datos

Los proyectos que se desarrollan actualmente tienen en general una arquitectura de aplicación distribuida, dividiéndose en tres módulos básicos:

1. Interfaz con los usuarios

2. Manipulación de datos
3. Base de datos.

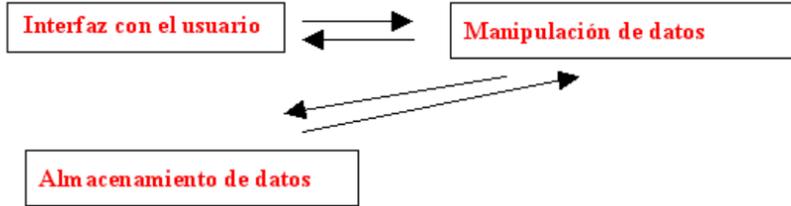


Imagen 2.7.1.1 – Arquitectura de aplicación distribuida.

Como podemos observar, uno de los aspectos de esta arquitectura es la comunicación con la fuente de datos.

Las empresas realizan sus operaciones diarias dependiendo de múltiples fuentes de datos, incluyendo servidores de base de datos, registros de archivo desfasados, correspondencia electrónica, documentos de productividad personal (hojas de cálculo, informes o presentaciones), y servidores de publicación basados en información.

Además, Internet se ha convertido en los últimos años en uno de los principales medios de difusión de información. Debido a su popularidad cada vez es mayor el número de proyectos que se desarrollan tomando a Internet como punto de partida.

El acceso a través de Internet posibilita una forma rápida y cómoda de acceder a la información almacenada en la base de datos con el fin de recuperarla, modificarla o insertarla. Se despliega así un enorme gama de posibilidades para el diseño de páginas Web: catálogos en línea, con información siempre actualizada sobre productos; información sobre visitantes, que puede actualizarse dinámicamente a medida que éstos navegan por el sitio Web; generación al vuelo de páginas, en función de selecciones o gustos del cliente, o del momento del día, o cualquier otro criterio; puede conservar el contenido de un sitio web en una base de datos, que se cargue dinámicamente cuando el usuario solicita una página, de forma que separe el diseño y formato de las páginas de su contenido; servicios en línea, como correo Web, acceso remoto a base de datos corporativas, noticias, guías de servicios y un largo etcétera.

Ahora bien, antes de que todo esto fuera posible, el principal problema fue que las diferentes plataformas de bases de datos (Oracle, SQL Server, Access, etc) no funcionan de igual manera, entonces, si escribimos una aplicación para acceder a las tablas de una base de datos de Access, y después queremos que la misma aplicación, y sin reescribir nada, utilice tablas de SQL Server u otra DB cualquiera, sería inútil ya que nuestra aplicación, diseñada para un motor concreto, no sabría dialogar con el otro.

Por lo tanto para solucionar este problema ha sido necesario crear nuevas tecnologías y estándares de interoperabilidad para que las soluciones empresariales funcionen con éxito independientemente de su forma o ubicación, permitiendo a los usuarios acceder de forma transparente a los datos de la empresa y modificarlos.

2.7.2 Universal Data Access

La solución propuesta por Microsoft, es conocida como Universal Data Access (UDA), que ofrece un modelo de programación consistente para acceder a cualquier tipo de datos, independientemente de su ubicación en la empresa.

Esta estrategia utiliza componentes COM (Component Object Model) que permite a los desarrolladores crear complejas aplicaciones utilizando una serie de pequeños objetos de software (componentes COM). Los ejemplos de un componente podrían ser un procedimiento de autorización de tarjeta de crédito, o normativas empresariales para calcular el coste de expediciones. El modelo de programación COM acelera el proceso de desarrollo capacitando a múltiples equipos de desarrollo para trabajar en distintas partes de una aplicación simultáneamente.

Las tecnologías que comprenden la estrategia Universal Data Access permiten a las organizaciones integrar varias fuentes de datos, crear soluciones de fácil mantenimiento, y utilizar su elección de herramientas, aplicaciones y servicios de plataformas. Para potenciar las inversiones, Universal Data Access no requiere el movimiento de datos costoso y prolongado en un sólo almacén de datos, ni requiere comprometerse con un único proveedor de productos de datos. En su lugar, Universal Data Access se basa en especificaciones abiertas de industria con un amplio soporte de industria, y funciona con todas las principales plataformas establecidas.

2.7.3 Microsoft Data Acces Components

Los Microsoft Data Access Components MDAC (ó Componentes de Acceso a Datos de Microsoft) son las tecnologías clave que habilitan el Universal Data Access (ó Acceso Universal a Datos). Mediante MDAC, las aplicaciones cliente/servidor conducidas por datos implementadas en la Web o en una LAN (Local Area Network ó Red de Área Local) pueden integrar fácilmente información de varias fuentes, tanto si son relacionales como si no. Estas tecnologías incluyen Open Database Connectivity - ODBC (Conectividad Abierta de Base de Datos) y Object Linking and Embedding for Databases – OLE DB ("Enlace e incrustación de objetos para bases de datos"), entre otros.

2.7.4 ODBC

Es un estándar de la industria para el tratamiento de datos relacionales mediante la sintaxis de consultas SQL en orígenes de datos dispares. ODBC presenta la ventaja de ser un estándar internacional que permite tratar un conjunto amplio de orígenes de datos relacionales a través de numerosos controladores ODBC de Microsoft y de otros proveedores.

La interfaz de ODBC posibilita que las aplicaciones accedan a datos almacenados en prácticamente cualquier sistema de gestión de bases de datos relacionales, ya que por un lado provee de unas características siempre homogéneas, y por el otro permite distintos controladores que aseguran la conectividad de la aplicación desarrollada con diferentes bases de datos.

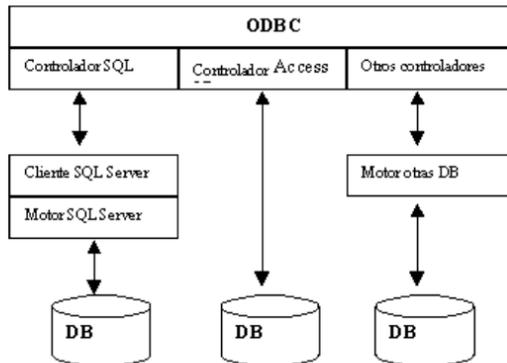


Imagen 2.7.4.1 - Interfaz de un ODBC.

ODBC es soportado por Microsoft, productos de desarrollo de aplicaciones de terceros, y aplicaciones de productividad de usuario final. Microsoft también ha implementado una cantidad de unidades ODBC para acceder a varios almacenamientos de datos de respaldo.

La interfaz de acceso a datos de Microsoft Open Database Connectivity continúa ofreciendo una forma unificada de acceder a datos relacionales como parte de la especificación OLE DB.

2.7.5 OLE DB

OLE DB es una interfaz de programación a nivel de sistema para trasladar datos dentro de la empresa. Es una especificación abierta diseñada para fundamentarse en ODBC aportando un estándar abierto para acceder a todo tipo de datos. Mientras que ODBC se planificó para acceder a bases de datos relacionales, OLE DB se diseñó para fuentes de información similares, relacionales o no.

Los componentes de OLE DB consisten en proveedores de datos (que contienen y exponen los datos), consumidores de datos (que utilizan los datos), y componentes de servicio como procesadores de consulta y motores de cursor (que reúnen y clasifican los datos). Las interfaces OLE DB se diseñaron para ayudar a distintos componentes a integrarse sin dificultades, para que los proveedores de componentes OLE DB puedan comercializar rápidamente productos de alta calidad.

Los proveedores de datos OLE DB implementan un conjunto de interfaces OLE DB que ofrecen una funcionalidad básica, permitiendo que otros proveedores de datos de OLE DB, componentes de servicios y aplicaciones de consumidor interoperen de forma de manera estándar y predecible. Los consumidores de datos pueden ser cualquier programa de software que requiera el acceso a una gran variedad de datos, lo que incluye herramientas de desarrollo, aplicaciones de productividad personal, productos de servicios de bases de datos, o componentes de servicios de OLE DB. Los componentes de servicios de OLE DB implementan la funcionalidad no implementada de forma nativa por algunos proveedores de datos OLE DB.

Universal Data Access permite desarrollar aplicaciones genéricas de consumidor OLE DB que accedan a fuentes de datos de respaldo de forma única y uniforme. Esto permite a las empresas extraer recursos y migrarlos de un almacenamiento de datos de respaldo a otro según sea necesario.

CAPÍTULO 3

DISEÑO

3.1 Descripción del Sistema de Simulación y Monitoreo del Rendimiento de Centrales Termoeléctricas: LAPEM – TERMO®

3.1.1 Antecedentes

El Sistema de Simulación y Monitoreo del Rendimiento de Centrales Termoeléctricas LAPEM-TERMO® surge principalmente de la necesidad de realizar pruebas de evaluación térmica dentro de las centrales termoeléctricas del país.

Ésta necesidad entró a un nivel crítico cuando hubo un aumento considerable en los servicios propuestos a la oficina de Sistemas Térmicos e Hidráulicos del LAPEM (Laboratorio de Pruebas de Equipos y Materiales, perteneciente a la Comisión Federal de Electricidad-CFE), la cual no contaba con la capacidad, tanto técnica como de personal, para realizar este incremento en los servicios.

Por ello el sistema informático con el que contaba LAPEM debía ser transformado casi en su totalidad para que pudiera cubrir todos los servicios propuestos.

El sistema actual cuenta con varios programas con una tarea específica para cada uno. El programa para realizar las pruebas necesita capturar manualmente diversos datos de operación, como por ejemplo, valores de temperatura y presión de diferentes equipos, con los cuales al combinarse con una serie de formulaciones, se reproducen las condiciones de eficiencia en cada uno de los equipos con los que cuenta una central termoeléctrica.

Por tal motivo, en este programa se necesita validar la información que se obtiene de la planta, labor que recae en un programa de validación realizado en el lenguaje de programación EES (Engineering Equation Solver).

Una vez validados estos valores, el sistema procede a la ejecución de la metodología de cálculo en la parte de prueba, la cual da como resultado una serie de datos con sus respectivos valores, entre ellos:

- Régimen Térmico
- CTU - Consumo Térmico Unitario
- Eficiencia de Turbinas
- Generador de vapor
- AE - Aproximación de Enfriamiento
- DTT – Diferencia de Temperatura Terminal
- Etc.

Al tener todos estos valores se procede a una etapa de comparación. En ella todos estos valores tienen que ser comparados contra valores de diseño, con el objetivo de saber que tan desviados se encuentran, y así poder mejorarlos posteriormente.

Para poder realizar un comparativo de los datos encontrados mediante el programa de pruebas, se elaboró una metodología de cálculo en la cual con datos del fabricante, balances de masa, de energía y algunos otros procesos, reprodujeran los datos característicos del Régimen Térmico, CTU y eficiencias de cualquier planta.

Al realizar el comparativo entre el programa de prueba y el de diseño se tomó como base el flujo de vapor principal, este valor tiene que ser introducido en forma manual para realizar los cálculos matemáticos. Ya teniendo el estado actual (valor de prueba) y el estado de diseño (valor de referencia), era necesaria contar con un tercer programa en el cual se mostrará la desviación del Régimen Térmico y la potencia de los parámetros más representativos de la central.

La limitación principal que LAPEM tiene en todo este proyecto, es que la entrada de datos es manual, además de que el sistema tiene limitaciones para su crecimiento y actualización, por lo que es muy necesario migrar a otra plataforma de programación que sea más flexible, que permita la operación en tiempo real y que se independice de la actualización de las licencias de uso.

3.1.2 Solución propuesta

Ante todo el problema planteado anteriormente, surge la necesidad de actualizar el sistema LAPEM, rediseñándolo totalmente y desarrollándolo en una nueva plataforma de programación. Toda esta labor no puede ser realizada por la oficina de Sistemas Térmicos e Hidráulicos del LAPEM, por lo que tiene que ser delegada a una segunda instancia, ya sea del sector público ó privado, que sea experta en el desarrollo de sistemas.

El Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET) es una entidad académica perteneciente a la Coordinación de la Investigación Científica de la UNAM que se caracteriza por el perfil multidisciplinario de sus académicos. Esta conjunción de conocimientos en diversas áreas de la ingeniería y de las ciencias físico-matemáticas le confiere al Centro una capacidad inusual en el medio universitario para afrontar proyectos interdisciplinarios de investigación y desarrollo dirigidos a la solución de problemas relevantes en nuestro entorno.¹

Con el fin de minimizar la dispersión de intereses que puede generarse en una comunidad multidisciplinaria, el CCADET definió cuatro campos prioritarios en los cuales deben concentrarse la actividad académicas de sus integrantes, independientemente de la especialidad de cada quien. Estos campos son: ¹

- Instrumentación científica
- Micro y nanotecnologías
- Tecnologías de la información
- Enseñanza de la ciencia y la tecnología

Tomando en cuenta el tercer campo antes mencionado y sabiendo que las Tecnologías de Información “se refieren al estudio, diseño, desarrollo, implementación, soporte o dirección de los sistemas de información computarizados”, podemos decir que el CCADET es una entidad totalmente calificada para realizar el proyecto que la oficina de Sistemas Térmicos e Hidráulicos del LAPEM necesita.

¹ Tomado de <http://www.cinstrum.unam.mx/>, Lunes 8 de Septiembre de 2008, 18:05 hrs.

Ante todo esto, podemos concluir que el CCADET cuenta con personal calificado en el área de modelado y simulación de sistemas, así como también en el diseño y desarrollo de sistemas informáticos, por lo que la Comisión Federal de Electricidad-CFE le entregó la responsabilidad de llevar a cabo el proyecto LAPEM-TERMO®.

Al tener el proyecto en sus manos, CCADET propuso el diseño de un sistema LAPEM-TERMO® que pueda ser operado en 3 modos distintos:

- Modo Manual
- Modo Semiautomático
- Modo Automático

Para lograr lo anterior el sistema se desarrollará en 2 versiones:

1. Versión Stand-Alone
2. Versión Web

La primera versión es la que será instalada y utilizada en todas y cada una de las centrales termoeléctricas del país que maneje el sistema LAPEM-TERMO® y podrá ser operada por los usuarios de los 3 modos distintos para la alimentación de datos, es decir, el usuario podrá introducir la información de forma manual, dato por dato, podrá también cargar la información de una serie de archivos en un formato previamente definido y también podrá estar operando el sistema en modo automático conectado al sistema de adquisición de datos. Un factor importante a considerar es que el modo de operación dependerá del nivel de automatización en la adquisición de datos con que cuente cada central.

La segunda versión será utilizada dentro de la Intranet de CFE. Estará instalada en un servidor de LAPEM y mediante una dirección WEB se podrá acceder desde la dirección general de CFE o desde cualquier otro punto con los permisos correspondientes para poder obtener información y realizar simulaciones en línea sobre la planta termoeléctrica que el usuario desee.

Todos los modelos matemáticos, de validación y cálculo con datos de fabricantes se basarán en las capacidades de la gente del LAPEM, pues ellos serán quien los proponga.

3.1.3 Objetivos del Sistema LAPEM-TERMO®

Para comenzar el proyecto integrando todas las consideraciones antes mencionadas, CCADET se propuso los siguientes objetivos:

1. Diseñar, desarrollar, implementar e integrar un sistema de simulación y monitoreo del rendimiento de centrales termoeléctricas para facilitar la evaluación de pruebas del régimen térmico, CTU y costos termo económicos en centrales termoeléctricas, a partir de los modelos matemáticos y código fuente del sistema actual con que cuenta el LAPEM. El sistema tendrá la capacidad de operar totalmente en línea con los sistemas de adquisición de datos de las plantas termoeléctricas.
2. Integración en una plataforma Web del sistema de simulación LAPEM-TERMO® de todas las plantas termoeléctricas para la evaluación de pruebas térmicas desde cualquier nodo de la CFE y del LAPEM.
3. Capacitación del personal de CFE en el uso del sistema.

3.1.4 Módulos del Sistema LAPEM-TERMO®

El sistema LAPEM-TERMO® es un desarrollo integral de un sistema informático de simulación monitoreo para realizar pruebas de evaluación térmica en las centrales termoeléctricas del país.

La estructura funcional de operación del sistema la podemos apreciar en la figura 3.1.4.1.

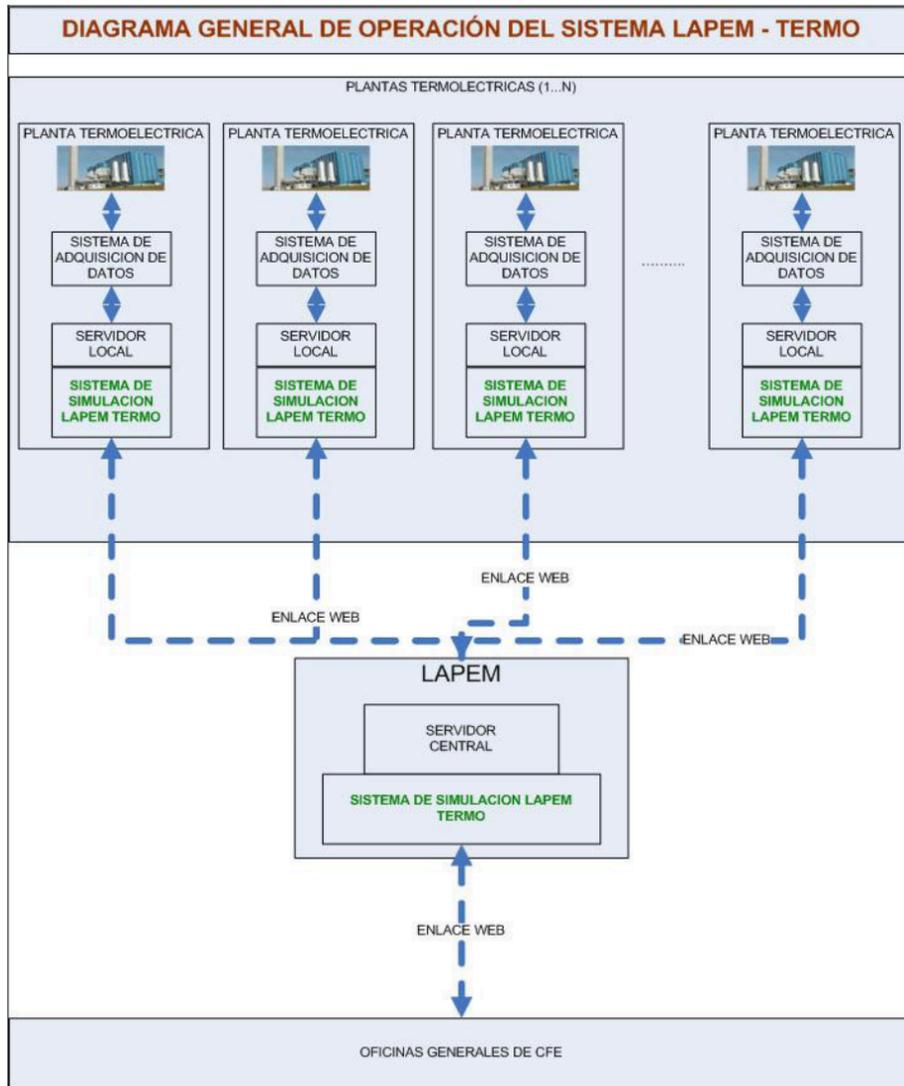


Imagen 3.1.4.1: Estructura funcional del sistema LAPEM -TERMO®

Esta estructura fue propuesta al personal del LAPEM, en la cual se aprecia el funcionamiento global del sistema.

Con base en lo anterior, la operación funcional propuesta para el sistema se aprecia en la siguiente figura 3.1.4.2.

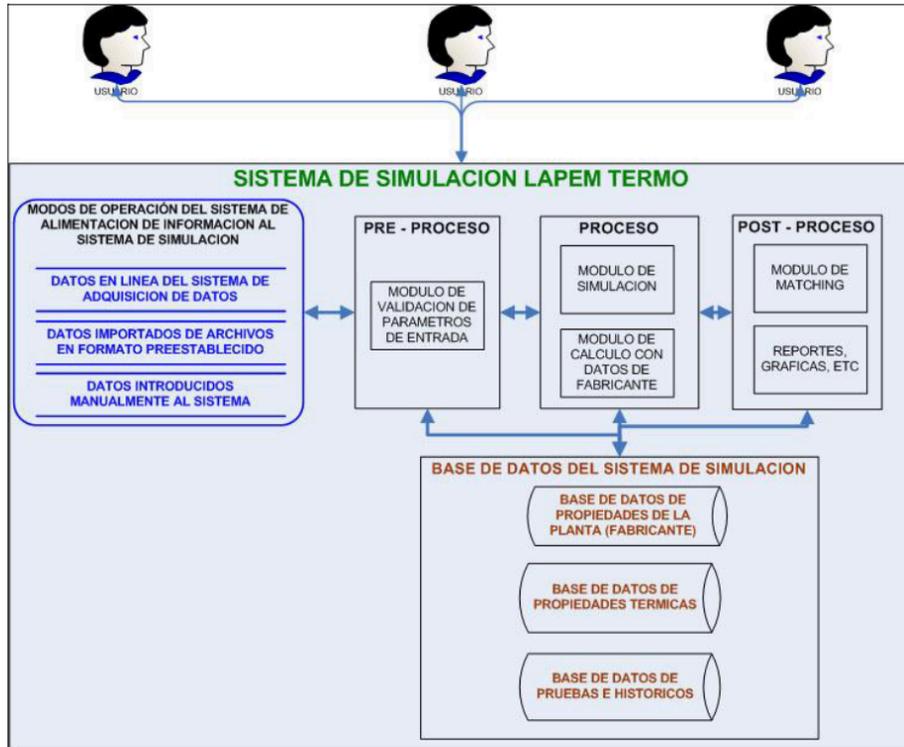


Imagen 3.1.4.2: Operación funcional del sistema LAPEM-TERMO®

El alcance del proyecto se describe a través del diseño y desarrollo de los componentes del sistema de simulación y monitoreo, cada uno de los cuales se va a considerar como entregables, y se mencionan a continuación.

1. Modulo de Entrada de Datos: Este modulo permitirá la captura de parámetros y variables necesarias para operar el funcionamiento del sistema. Este operará de 3 formas, una en forma manual donde el usuario tendrá unas interfaces para la captura de las variables requeridas en función de la planta, la otra forma es que el usuario podrá elegir la forma de ingresar los datos de las variables a través de un archivo de datos, el cual tendrá una estructura definida con la opción de diversos formatos como txt, csv, xls, etc.; y la tercera opción realizara una lectura de los datos de las bases de datos del sistema de adquisición de datos que se tenga implementado la planta.

2. Modulo de Validación de parámetros de entrada: Este modulo se encargará de que los datos proporcionados por el usuario o por el sistema en línea de adquisición de datos, se encuentre dentro de los rangos permitidos para su operación de la planta, estos rangos permitidos estarán en función de la información del fabricante de la planta.

3. Modulo de Simulación: Este modulo se encargara de realizar la simulación de todo el proceso termoeléctrico para evaluar el régimen térmico, las CTU, la eficiencia, etc. Este modulo empleará una herramienta embebida para el manejo de las propiedades térmicas del vapor.

4. Modulo de Calculo con datos de fabrica: Este modulo realizara cálculos con los parámetros proporcionados por el fabricante de la planta, de acuerdo a las condiciones en que se haya realizado el calculo de la simulación del modulo 3.

5. Modulo de Matching: Este modulo tendrá la función de realizar una comparación de los resultados generados por los módulos 3 y 4, para posteriormente generar reportes de eficiencia y de régimen térmico.

6. Modulo de la Interfase Humano-Máquina: Este modulo permite la configuración de cada central según sus condiciones de operación particulares, así como la presentación de los valores de los diversos parámetros de operación y resultados para cada equipo de una central. Este desarrollo se hará en el lenguaje de programación C#.

7. Modulo de Resultados, Reportes, Graficas, etc.: Este modulo le brindara al usuario del sistema de simulación la capacidad de generar reportes y graficas del proceso de simulación, y también la capacidad de exportar los reportes y graficas o formatos comerciales para uso ejecutivo de los mismos resultados. Este modulo tendrá la posibilidad de exportar archivos en formatos txt, csv, xls, etc.

8. Modulo de Base de Datos: El modulo de base de datos tendrá estrecha comunicación con los demás módulos ya que en el se almacenaran 3 tipos de información, la cual será por una lado, información referente a propiedades de la planta proporcionados por el fabricante, información de propiedades térmicas utilizadas en el proceso de calculo y simulación e información referentes a las distintas pruebas realizadas al sistema, así como un histórico para efectos de reportes ejecutivos. Para ese desarrollo se empleará SQL Server y Microsoft Access.

9. Versión Stand-Alone: Instalador de la versión Stand-Alone del sistema, desarrollado en visual Basic.

10. Versión Web: Instalador de la versión Web del sistema, desarrollado en ASP.NET.

11. Capacitación: Se realizará un taller para la capacitación de personal de CFE en el uso del sistema.

3.2 Descripción del Sistema de Gestión de Base de Datos “DBMS” par el Sistema de Simulación y Monitoreo del Rendimiento de Centrales Termoeléctricas: LAPEM – TERMO®

Como vimos en el capítulo dos, un Sistema de Gestión de Base de Datos –DBMS es un software encargado de servir como interfaz entre el usuario, la base de datos y las aplicaciones que la utilizan; manejando de manera clara, sencilla y ordenada un conjunto de datos que representan información relevante para el usuario.

Trasladando este concepto al sistema LAPEM-TERMO®, podemos decir que el DBMS de este sistema administrará la base de datos que almacenará toda la información que se necesite para poder realizar las pruebas de evaluación térmica, que es el objetivo central del sistema LAPEM-TERMO®.

Observando la figura 3.1.4.2 y el módulo 8 del tema anterior, podemos observar que el sistema requiere almacenar en la base de datos tres tipos de datos:

1. Datos de propiedades de la planta: Información referente a propiedades de la planta proporcionados por el fabricante.
2. Datos de propiedades térmicas: Información de propiedades térmicas utilizadas en el proceso de cálculo y simulación
3. Datos de pruebas e históricos: Información referente a las distintas pruebas realizadas al sistema, así como un histórico para efectos de reportes ejecutivos.

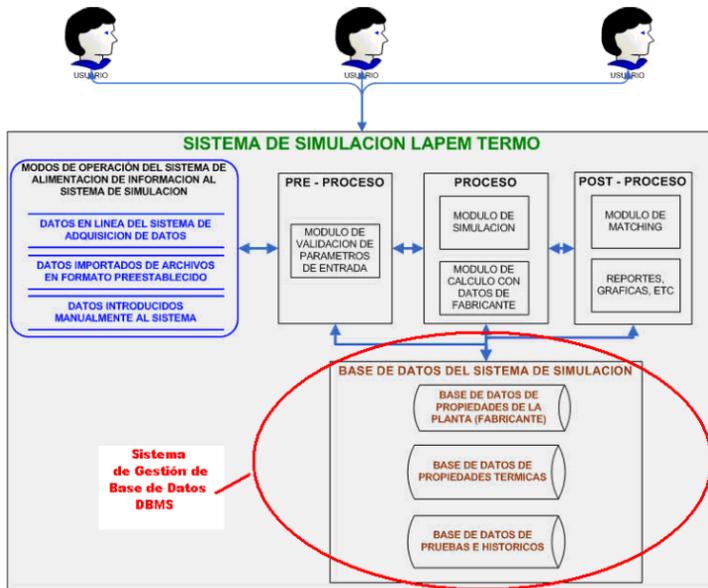


Imagen 3.2.1: DBMS del sistema LAPEM-TERMO®

3.2.1 Diseño de la Base de Datos para el Sistema LAPEM-TERMO®

El primer paso que tenemos que realizar para construir el DBMS para nuestro sistema LAPEM-TERMO® será el diseñar la base de datos que administrará.

Para diseñar la base de datos se tomó como base el Modelo Relacional, el cual se describió en el capítulo 2 de ésta tesis, y en el que, como recordaremos, representa los datos y las relaciones entre los datos mediante una colección de tablas, cada una de las cuales tiene un número de columnas con nombres únicos. Se utilizó este modelo debido a la sencillez que tiene al momento de diseñar y estructurar la base de datos, lo cual representa una característica muy asequible a los usuarios, ya que ésta sencillez también se manifiesta en las operaciones de consulta y actualización sobre la base de datos.

Para poder representar el esquema de una base de datos relacional se debe conocer primeramente el nombre de sus relaciones, después necesitamos saber el nombre de sus atributos, y por último, ya conociendo sus atributos, saber cuál o cuáles de ellos serán sus llaves primarias y cuales sus llaves foráneas.

Teniendo en cuenta lo anterior, podemos decir que el primer paso del Modelo Relacional es establecer las relaciones que estarán presentes en la base de datos. Como ya se mencionó anteriormente, este modelo se centra en el concepto de relación, la cual podemos definir como una tabla conformada por columnas y filas.

Entonces empezaremos estableciendo el esquema de las relaciones que utilizaremos en la base de datos del sistema LAPEM-TERMO®.

Partiendo de los tres tipos de datos que va a almacenar la base de datos del sistema LAPEM-TERMO®, podemos empezar a desglosar las relaciones que estarán presentes en ella:

1. **PLANTAS:** El sistema LAPEM-TERMO® estará presente en varias centrales termoeléctricas del país, por ello será necesario crear una relación llamada "Plantas" para que el sistema pueda ubicar correctamente a que central pertenece un dato en específico. EL sistema LAPEM-TERMO® trabajará con 25 centrales termoeléctricas en total.

2. **UNIDADES:** Cada central termoeléctrica cuenta con una o varias unidades, por eso será necesario crear una relación llamada “Unidades”, ya que dentro de una misma planta pueden existir dos unidades con características diferentes, y por ende manejan datos diferentes. El sistema LAPEM-TERMO® trabajará con 80 unidades en total, las cuales se distribuyen dentro de las 25 centrales termoeléctricas del sistema.

3. **EQUIPOS:** Cada una de las 80 Unidades de las centrales termoeléctricas cuenta con cierto número de equipos, los cuales pueden variar de unidad a unidad y de planta a planta. Por lo anterior se creará una relación llamada “Equipos”, ya que cada uno de estos equipos maneja información única, de tal manera que si existen equipos repetidos, éstos no necesariamente tienen que manejar la misma información. El sistema LAPEM-TERMO® trabajará con 1382 equipos en total distribuidos dentro de las 80 Unidades del sistema.

4. **IMAGEN _ EQUIPOS:** Cada uno de los 1382 equipos del sistema LAPEM-TERMO®, contará con una imagen, la cual contendrá un diagrama del equipo, en el que se describirán tanto las partes del equipo, como el valor de las variables que maneja. Por ello es muy importante mostrar todos estos datos dentro de una imagen, ya que para el usuario del sistema será más fácil ver los datos del equipo en cuestión de ésta manera. Como son muchos equipos en el sistema, también tendremos muchas imágenes, por lo que es conveniente crear una relación llamada IMAGEN_EQUIPOS, la cual servirá como un catálogo de la relación llamada EQUIPOS (en términos de un diagrama Entidad-Relación, podríamos decir que IMAGEN_EQUIPOS es una entidad débil).

5. **USUARIOS:** El sistema LAPEM-TERMO® trabajará con 3 tipos de usuarios (“LAPEM-CONFIGURACIÓN”, “DIRECTIVO” E “ING. PRUEBAS”), los cuales tendrán distintos privilegios sobre el sistema y tendrán acceso a él mediante una contraseña cifrada. En el sistema se podrán dar de alta y de baja usuarios, por lo que se debe crear una relación llamada “Usuarios” para poder almacenar toda la información de cada uno de ellos.

6. **UNIDAD_M:** Debido a que el sistema LAPEM-TERMO® es un sistema para el monitoreo y simulación de centrales termoeléctricas, resulta obvio que maneja muchos tipos de datos y por ende, de magnitudes. Para cada una de éstas magnitudes, el sistema tendrá la opción de poder manejar los

datos respectivos en diferentes unidades de medida; por ejemplo, para la magnitud de presión, el sistema podrá expresar los datos en unidades como los Pa, kPa, MPa, bar, mmbar, atm, etc., donde todas y cada una de las unidades anteriores, son unidades de presión. Por ello será necesario crear una relación llamada UNIDAD_M, en la que se almacenarán todas las magnitudes que utilizará el sistema, así como todas las unidades de medida en las que se podrá trabajar cada una de esas magnitudes.

7. **UNIDAD_SALIDA:** Como se mencionó en el punto anterior, el sistema maneja varios tipos de magnitudes y varios tipos de unidades para cada una de las magnitudes. Con esto establecido, podemos decir que el encargado de la central termoeléctrica será quien decida en que unidades de medida requiere los datos obtenidos de los monitoreos y simulaciones. Es aquí donde entra el concepto de “unidades de salida”, el cual sirve para decir en que tipo de unidad para cada una de las magnitudes manejadas, estarán expresados los resultados. Por ello será necesario crear una relación llamada UNIDAD_SALIDA, la cual servirá para almacenar el tipo de unidad elegido para cada magnitud del sistema.

8. **VARIABLES_PRUEBA:** Una parte muy importante del sistema LAPEM-TERMO® es que contará con un programa que capture datos de operación dentro de la central termoeléctrica (por ejemplo, valores de temperatura ó presión de diferentes equipos), los cuales al combinarse con una serie de formulaciones, se reproducen las condiciones de eficiencia en cada uno de los equipos con los que cuenta una central termoeléctrica. A éstos datos se les dio el nombre de “variables de prueba”, por lo que será necesario crear una relación llamada VARIABLES_PRUEBA para poder almacenar toda esta información.

9. **VARIABLES_REFERENCIA:** Otra parte muy importante del sistema LAPEM-TERMO® será el programa que obtiene los valores de diseño, para lo cual se elaboró una metodología de cálculo en la cual con datos del fabricante, balances de masa, de energía y algunos otros procesos, reprodujeran los datos característicos del Régimen Térmico, CTU y eficiencias de cualquier planta. Por ello será necesario crear una relación llamada VARIABLES_REFERENCIA, que nos servirá para guardar todos los datos de éstas variables.

10. **VARIABLES_DESVIACIONES:** Como ya se mencionó, al tener todos estos valores (los de prueba y los de referencia), se procede a una etapa de comparación. En ella todos estos valores de prueba tienen que ser comparados contra valores de diseño, con el objetivo de saber que tan desviados se encuentran, y así poder mejorarlos posteriormente. Por ello necesitamos una relación llamada VARIABLES_DESVIACIONES, que nos ayudará a almacenar los datos de todas éstas desviaciones.

11. **RESULTADOS_PRUEBA_H:** Todos los resultados que obtiene el programa que maneja los valores de prueba y todos los que se deducen con estos últimos, necesitan almacenarse como datos históricos. Por ésta razón será necesario crear una relación llamada RESULTADO_PRUEBA_H, la cual guardará todos los datos obtenidos en tiempo real (que con el tiempo pasarán a ser históricos), y los cuales ayudarán a tomar decisiones dentro de la central termoeléctrica.

12. **RESULTADOS_REFERENCIA_H:** De la misma forma que la relación anterior, se tiene que crear una relación llamada RESULTADOS_REFERENCIA para almacenar todos los datos históricos que se obtienen de y con los datos de diseño (ó referencia), ya que, como sabemos, estos sirven principalmente para calcular las desviaciones, y por tanto son muy importantes y es conveniente llevar un registro completo de ellos.

Al terminar de establecer las relaciones que existirán dentro de la base de datos, ahora, como segundo paso, toca mencionar los atributos que llevará cada una de éstas relaciones, los cuales se establecerán básicamente mediante los datos que necesitamos saber de cada una de esas relaciones, los cuales servirán tanto para identificarlas unas de otras, así como para poder realizar las pruebas de evaluación térmica dentro de cada central termoeléctrica.

Después de un largo análisis se definieron los atributos que llevará cada una de las relaciones, los cuales son los siguientes:

1. PLANTAS:

- ID_Planta
- Nombre_Planta
- Domicilio

- Teléfono
- Entidad
- Super_Int
- E-mail
- Descripción
- Imagen
- X
- Y

2. UNIDADES:

- ID_Unidad
- Nombre_Unidad
- Descripción
- Imagen
- N_TurbinasI
- Potencia
- U_Potencia
- N_Extracciones
- T_Coeficientes
- OPC
- Intervalo
- T_Intervalo
- N_Lecturas
- Validación

3. EQUIPOS:

- ID_Equipo
- Nombre__Equipo
- Descripción
- X1
- Y1
- X2
- Y2
- HABILITADO
- SELEC

4. IMAGEN _ EQUIPOS:

- ID_IMAGEN
- IMAGEN

5. USUARIOS:

- Nivel
- Usuario
- Clave
- Nombre_Usuario
- Descripción
- Tipo

6. UNIDAD _M:

- ID_UNIMED
- TIPO
- MAGNITUD
- NOM_UNIDAD

7. UNIDAD _SALIDA:

- ID_UNIMED
- MAGNITUD

8. VARIABLES _PRUEBA:

- ID_VARIABLE
- ID_EQUIPO
- Código
- Descripción
- Clave_Unidad
- ID_UNIMED
- OPCIONES_OPC
- DEFAULT_OPC

- VALOR_FIJO
- VALOR
- VALORMIN
- VALORMAX
- VALORDEF
- ID_UNIMEDT
- X
- Y
- X1
- Y1
- TAG
- Tipo

9. VARIABLES_REFERENCIA:

- ID_EQUIPO
- ID_VARIABLE
- CODIGO
- DESCRIPCION
- VALOR_100
- VALOR_75
- VALOR_50
- VALOR_25
- CLAVE_UNIDAD
- ID_UNIMED
- VALOR
- ID_UNIMEDT
- X
- Y
- X1
- Y1
- TIPO

10. VARIABLES_DESVIACIONES:

- ID
- ID_UNIDAD
- ID_VARIABLE
- Nombre
- EC100
- EC75

- EC50
- EC25
- C100A
- C100B
- C100C
- C100D
- C100E
- C100F
- C100G
- C75A
- C75B
- C75C
- C75D
- C75E
- C75F
- C75G
- C50A
- C50B
- C50C
- C50D
- C50E
- C50F
- C50G
- C25A
- C25B
- C25C
- C25D
- C25E
- C25F
- C25G

11. RESULTADOS_PRUEBA_H:

- ID
- FECHA
- HORA
- ID_VARIABLE
- CODIGO
- ID_UNIMED
- VALOR
- TIPO

12.RESULTADOS_REFERENCIA_H:

- ID
- FECHA
- HORA
- ID_VARIABLE
- CODIGO
- ID_UNIMED
- VALOR

Después de lo anterior y como tercer paso, toca establecer completamente el esquema de todas las relaciones mencionadas.

Como se mencionó en el capítulo 2, en el esquema de una relación aparecen los nombres de las relaciones seguidos de los nombres de sus atributos encerrados entre llaves. Las llaves primarias son los atributos subrayados. Las llaves foráneas son los atributos que aparecen doblemente subrayados. En todos los casos los atributos de la relación aparecen acompañados del tipo de variable empleada para su almacenamiento (el tipo de variable puede variar dependiendo del DBMS que se utilice).

El esquema de las relaciones para la base de datos del sistema LAPEM-TERMO queda de la siguiente manera:

PLANTAS {ID_Planta NUMBER, Nombre_Planta CHAR, Domicilio CHAR, Telefono CHAR, Entidad CHAR, Super_Int CHAR, E-mail CHAR, Descripcion CHAR, Imagen OLE OBJECT, X NUMBER, Y NUMBER}

UNIDADES {ID_Unidad NUMBER, ID_Planta NUMBER, Nombre_Unidad CHAR, Descripcion CHAR, Imagen OLE OBJECT, N_TurbinasI NUMBER, Potencia NUMBER, U_Potencia CHAR, N_Extracciones NUMBER, T_Coeficientes NUMBER, OPC CHAR, Intervalo NUMBER, T_Intervalo NUMBER, N_Lecturas NUMBER, Validacion BOOLEAN}

EQUIPOS {ID_Equipo NUMBER, ID_Unidad NUMBER, Nombre_Equipo CHAR, Descripcion CHAR, ID_Imagen NUMBER, X1 NUMBER, Y1 NUMBER, X2 NUMBER, Y2 NUMBER, Habilitado NUMBER, Selec NUMBER}

IMAGEN _ EQUIPOS {ID_Imagen NUMBER, Imagen OLE OBJECT}

USUARIOS {Nivel NUMBER, Usuario CHAR, Clave CHAR, Nombre_Usuario CHAR, Descripcion CHAR, Tipo NUMBER}

UNIDAD_M {ID_UniMed NUMBER, Tipo NUMBER, Magnitud CHAR, Nom_Unidad CHAR}

UNIDAD_SALIDA {ID_Unidad NUMBER, ID_UniMed NUMBER, Magnitud CHAR}

VARIABLES_PRUEBA {ID_VariableP NUMBER, ID_Equipo NUMBER,Codigo CHAR, Descripcion CHAR, Clave_Unidad NUMBER, ID_UniMed NUMBER, Opciones OPC NUMBER, Default OPC NUMBER, Valor_Fijo NUMBER, Valor NUMBER, ValorMin NUMBER, ValorMax NUMBER, ValorDef NUMBER, ID_UniMedT NUMBER, X NUMBER, Y NUMBER, X1 NUMBER, Y1 NUMBER, TAG NUMBER, Tipo CHAR}

VARIABLES_REFERENCIA {ID_Equipo NUMBER, ID_VariableR NUMBER, Codigo CHAR, Descripcion CHAR, Valor_100 CHAR, Valor_75 CHAR, Valor_50 CHAR, Valor_25 CHAR, Clave_Unidad NUMBER, ID_UniMed NUMBER, Valor NUMBER, ID_UniMedT NUMBER, X NUMBER, Y NUMBER, X1 NUMBER, Y1 NUMBER, Tipo NUMBER}

VARIABLES_DESVIACIONES {ID_Desviacion AUTOINCREMENT, ID_Unidad NUMBER, ID_VariableD CHAR, Nombre CHAR, EC100 NUMBER, EC75 NUMBER, EC50 NUMBER, EC25 NUMBER, C100A NUMBER, C100B NUMBER, C100C NUMBER, C100D NUMBER, C100E NUMBER, C100F NUMBER, C100G NUMBER, C75A NUMBER, C75B NUMBER, C75C NUMBER, C75D NUMBER, C75E NUMBER, C75F NUMBER, C75G NUMBER, C50A NUMBER, C50B

NUMBER, C50C NUMBER, C50D NUMBER, C50E NUMBER, C50F NUMBER, C50G NUMBER, C25A NUMBER, C25B NUMBER, C25C NUMBER, C25D NUMBER, C25E NUMBER, C25F NUMBER, C25G NUMBER}

RESULTADOS_PRUEBA_H {ID_ResPH AUTOINCREMENT, Fecha CHAR, Hora CHAR, ID_VariableP NUMBER, Codigo CHAR, ID_UniMed NUMBER, Valor NUMBER, Tipo CHAR}

RESULTADOS_REFERENCIA_H {ID_ResRH AUTOINCREMENT, Fecha CHAR, Hora CHAR, ID_VariableR NUMBER, Codigo CHAR, ID_UniMed NUMBER, Valor NUMBER}

Para poder entender los tipos de datos de las relaciones anteriores observemos la siguiente tabla, la cual muestra los tipos de datos que soporta el SQL de Microsoft® Jet versión 4.0 (como ya se mencionó desde el capítulo 1, la base de datos del sistema LAPEM-TERMO® se realizará tanto en SQL Server como en Access, ambos productos de Microsoft, por ello se tomará como base ésta tabla). Los sinónimos son palabras equivalentes al tipo de dato indicado. El tamaño indica cuánto ocupará una columna del tipo indicado.

| <u>Tipo de dato</u> | <u>Sinónimos</u> | <u>Tamaño</u> | <u>Descripción</u> |
|---------------------|--|---------------------|---|
| BINARY | VARBINARY BINARY VARYING BIT VARYING | 1 byte por carácter | Se puede almacenar cualquier tipo de datos en un campo de este tipo. Los datos no se traducen (por ejemplo, a texto). La forma en que se introducen los datos en un campo binario indica cómo aparecerán al mostrarlos. |
| BIT | BOOLEAN LOGICAL LOGICAL1 YESNO | 1 byte | Valores Sí y No, y campos que contienen solamente uno de dos valores. |
| TINYINT | INTEGER1 BYTE | 1 byte | Un número entero entre 0 y 255. |
| COUNTER | AUTOINCREMENT | | Se utiliza para campos contadores cuyo valor se incrementa automáticamente al crear un nuevo registro. |
| MONEY | CURRENCY | 8 bytes | Un número entero comprendido entre - 922.337.203.685.477,5808 y 922.337.203.685.477,5807. |
| DATETIME | DATE TIME | 8 bytes | Una valor de fecha u hora entre los años 100 y 9999 |

| | | | |
|------------------|--|--|--|
| UNIQUEIDENTIFIER | GUID | 128 bits | Un número de identificación único utilizado con llamadas a procedimientos remotos. |
| DECIMAL | NUMERIC DEC | 17 bytes | Un tipo de datos numérico exacto con valores comprendidos entre 1028 - 1 y - 1028 - 1. Puede definir la precisión (1 - 28) y la escala (0 - precisión definida). La precisión y la escala predeterminadas son 18 y 0, respectivamente. |
| REAL | SINGLE FLOAT4 IEEESINGLE | 4 bytes | Un valor de coma flotante de precisión simple con un intervalo comprendido entre - 3,402823E38 y - 1,401298E-45 para valores negativos, y desde 1,401298E-45 a 3,402823E38 para valores positivos, y 0. |
| FLOAT | DOUBLE FLOAT8 IEEEDOUBLE NUMBER | 8 bytes | Un valor de coma flotante de precisión doble con un intervalo comprendido entre - 1,79769313486232E308 y - 4,94065645841247E-324 para valores negativos, y desde 4,94065645841247E-324 a 1,79769313486232E308 para valores positivos, y 0. |
| SMALLINT | SHORT INTEGER2 | 2 bytes | Un entero corto entre - 32.768 y 32.767. |
| INTEGER | LONG INT INTEGER4 | 4 bytes | Un entero largo entre - 2.147.483.648 y 2.147.483.647. |
| IMAGE | LONGBINARY GENERAL OLEOBJECT | Lo que se requiera | Desde cero hasta un máximo de 2.14 gigabytes. Se utiliza para objetos OLE. |
| TEXT | LONGTEXT LONGCHAR MEMO NOTE NTEXT | 2 bytes por carácter. (Consulte las notas). | Desde cero hasta un máximo de 2.14 gigabytes. |
| CHAR | TEXT(n) ALPHANUMERIC CHARACTER STRING VARCHAR CHARACTER VARYING NCHAR NATIONAL CHARACTER NATIONAL CHAR NATIONAL CHARACTER VARYING NATIONAL CHAR VARYING | 2 bytes por carácter. (Consulte las notas). | Desde cero a 255 caracteres. |

Cuadro 3.2.1.1-Tipos de datos que soporta el SQL de Microsoft® Jet versión 4.0 (Tomado de http://www.aulaclie.es/sql/a_8_1_1.htm, 6 de Octubre de 2008, 17:40 hrs.).

Ya con toda la información que nos proporcionan los esquemas de las relaciones (atributos de la relación, llaves primarias, llaves foráneas, etc.), y sabiendo que una relación es una tabla conformada por columnas y filas, podemos empezar a visualizar la forma del como quedarían relacionadas nuestras tablas

Teniendo establecidas las relaciones para la base de datos del sistema LAPEM-TERMO® como parte de Modelo Relacional, podemos proceder a realizar un diagrama que nos muestre como quedarían relacionadas nuestras tablas (recordemos, como se mencionó en el capítulo dos, que un diagrama es la representación gráfica de un esquema ó un modelo).

Hoy en día existen muchos tipos de software para poder realizar este tipo de diagramas, los cuales cuentan también con opciones para exportar el diagrama a algún DBMS y así crear instantáneamente la base de datos y con ello ahorrar mucho tiempo valioso.

El software que utilizaremos para realizar nuestro diagrama de relaciones será ERwin Data Modeler Versión 7.0 , que es uno de los más usados en el mundo actual. El diagrama queda de la siguiente manera:

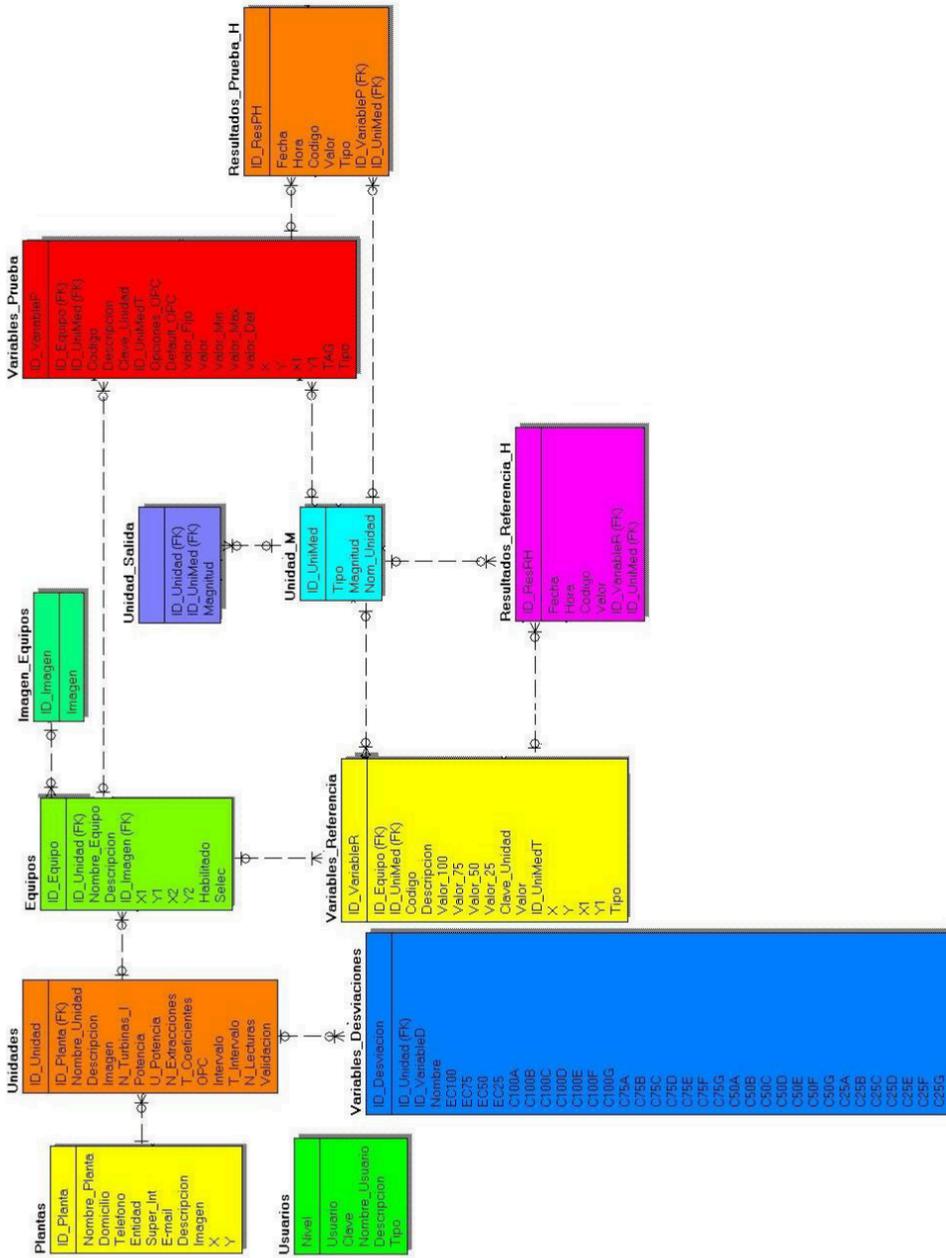


Diagrama 3.2.1.1-Modelo de la Base de Datos para el Sistema LAPEM-TERMO®

3.2.2 Diseño de la Interfaz Gráfica para la Base de Datos del Sistema LAPEM-TERMO®

Como ya se había mencionado antes, un Sistema de Gestión de Base de Datos - DBMS es un software encargado de servir como interfaz entre el usuario y la base de datos, manejando de manera clara y sencilla la información relevante para el usuario.

Una Interfaz Gráfica de Usuario (**Graphical User Interface-GUI**) en el ámbito de los sistemas de información podemos definirla como una herramienta de un sistema interactivo, la cual, a través del uso y representación del lenguaje visual, permite al usuario elegir comandos, iniciar programas, ver listas de archivos, guardar información, entre otras opciones. Estas elecciones comúnmente se hacen mediante iconos, listas o menús visuales, activando las elecciones a través del uso del teclado ó ratón de la computadora.

Tomando en cuenta lo anterior, es lógico pensar que para que un DBMS cumpla su propósito, este debe de contar con una buena interfaz gráfica de usuario. Es por ello que después de haber diseñado la base de datos del sistema LAPEM-TERMO®, toca como siguiente paso, diseñar la interfaz gráfica para manejarla.

Ésta interfaz gráfica correspondería a la versión Stand Alone del sistema LAPEM-TERMO® (como se mencionó al principio de este capítulo), mientras que para la segunda versión del sistema, es decir, la versión WEB, se tendrá una interfaz gráfica muy parecida pero con algunos cambios importantes debido a que será una versión para consultas y tomas de decisiones en línea. Teniendo esto en cuenta, cabe hacer la aclaración de que para la presente tesis solo se mencionará el diseño de la versión Stand Alone, que es la versión que se instalará localmente en todas y cada una de las centrales termoeléctricas del sistema LAPEM-TERMO®.

Para empezar a diseñar la interfaz gráfica el primer paso que tenemos que realizar es definir las opciones que ofrecerá el sistema para administrar la central termoeléctrica y así poder cubrir el principal objetivo del sistema LAPEM-TERMO® que es el de poder realizar las pruebas de evaluación térmica. Por ello se tomarán en cuenta los requerimientos que la CFE pidió a CCADET para que el sistema realice este tipo de pruebas.

Entre los requisitos que pidió la CFE destacan:

1. Opción para validar usuario.
2. Opción para seleccionar central termoeléctrica.
3. Opción para cambiar de tipo de usuario.
4. Opción para cambiar de central termoeléctrica.
5. Opción para crear, modificar o eliminar usuarios.
6. Opción para crear, modificar o eliminar plantas.
7. Opción para crear, modificar o eliminar unidades.
8. Opción para crear, modificar o eliminar equipos.
9. Opción para configurar coeficientes de desviaciones.
10. Opción para configurar señales de entrada.
11. Opción para configurar señales de salida.
12. Opción para configurar OPC.
13. Opción para respaldar datos de referencia o prueba.
14. Opción para restaurar datos de referencia o prueba.
15. Opción para configurar datos de diseño.
16. Opción para agregar, modificar o eliminar valores de referencia.
17. Opción para realizar simulaciones con valores de prueba.
18. Opción para realizar simulaciones con valores de referencia.
19. Opción para ver informes de simulaciones.
20. Opción para generar reportes de datos generales para cada simulación y monitoreo.
21. Opción para generar reporte de los datos de equipos de la última simulación ó monitoreo.
22. Opción para generar reportes históricos.
23. Opción para generar reportes gráficos de simulaciones con valores de prueba, referencia y/o desviaciones.

Como segundo paso realizaremos un diagrama jerárquico de bloques para representar las opciones que tiene que presentar la interfaz gráfica con base en los requerimientos anteriores, la cual también nos servirá para organizar jerárquicamente cada uno de éstas opciones.

Para lograr esto, agruparemos en diferentes niveles cada uno de los requerimientos de la lista anterior, tomando como base el orden en que se presentarían ante el usuario cada una de éstas opciones al utilizar la interfaz gráfica del sistema, es decir, tomando como el primer nivel la opción(es) que se presente(n) en primer plano cuando el usuario inicia el sistema LAPEM-TERMO®, y cómo el nivel más bajo la opción(es) que se considere(n) como el último paso que realizaría el usuario al utilizar el sistema LAPEM-TERMO®.

Utilizando ésta lógica se definieron 6 niveles para los 23 requisitos anteriores, agrupándose de la siguiente manera:

1. **NIVEL 0** - *NIVEL DE VALIDACIÓN DE USUARIO Y DE C.T. (Central Termoeléctrica):* Opción 1 y 2.
2. **NIVEL 1** - *NIVEL DE CAMBIO DE USUARIO Ó DE C.T.:* Opción 3 y 4.
3. **NIVEL 2** - *NIVEL DE CONFIGURACIÓN DEL SISTEMA:* Opción 5, 6, 7, 8, 9, 10, 11, 12, 13 y 14.
4. **NIVEL 3** - *NIVEL DE CONFIGURACION DE DATOS DE DISEÑO:* Opción 15 y 16.
5. **NIVEL 4** - *NIVEL DE REALIZACIÓN DE SIMULACIONES:* Opción 17, 18, 19, 20 y 21.
6. **NIVEL 5** - *NIVEL DE GENERACIÓN DE REPORTES HISTÓRICOS:* Opción 22 y 23.

Ahora que ya tenemos los 6 niveles definidos junto con las opciones que presentan cada uno de ellos, podemos proceder a realizar nuestro diagrama jerárquico, el cual queda de la siguiente manera:

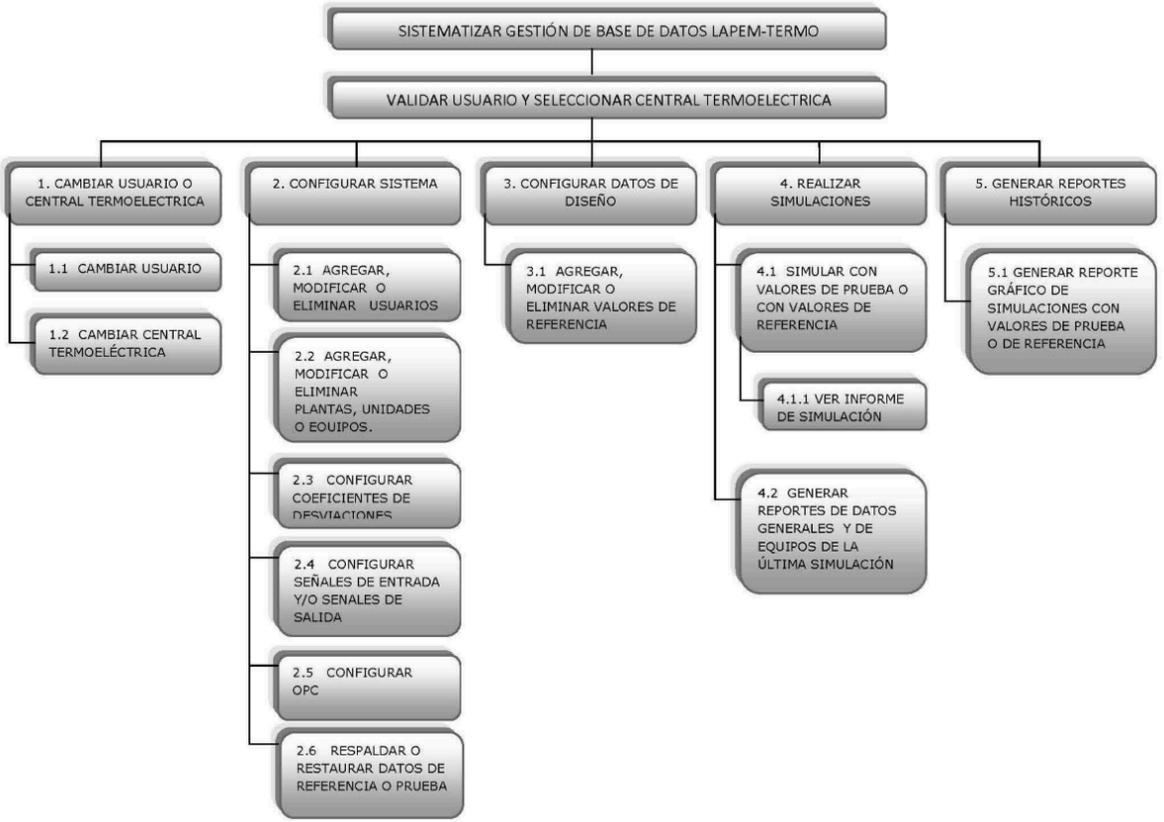


Diagrama 3.2.2.1-Modelo Jerárquico de la Interfaz Gráfica del Sistema LAPEM-TERMO®

Como tercer y último paso para terminar con el diseño de nuestra interfaz gráfica para la base de datos del sistema LAPEM-TERMO®, tenemos que definir como van a interactuar entre sí, la base de datos y la interfaz gráfica.

Para ello definiremos 6 diagramas de flujo (uno para cada nivel definido en el paso anterior), en donde se representará de manera gráfica y estructurada, la forma en que cada una de las opciones de cada nivel interactúan con la base de datos.

Para el NIVEL 0 tenemos el siguiente diagrama de flujo:

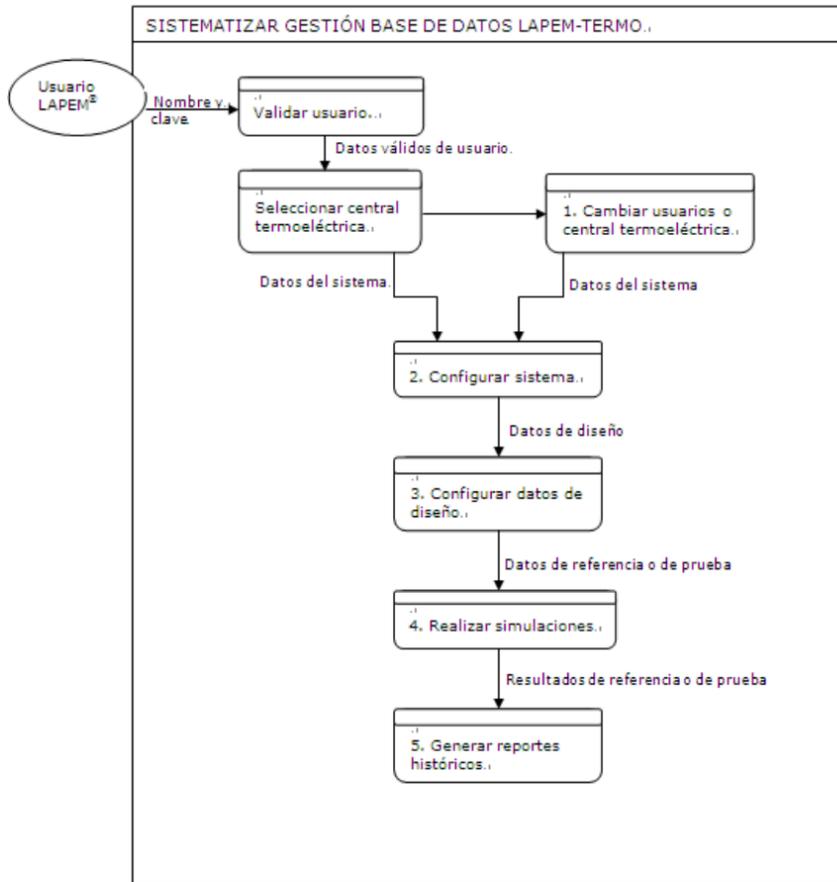


Diagrama 3.2.2.2-Diagrama de Flujo del Nivel 0 de la GUI del Sistema LAPEM-TERMO®

Para el NIVEL 1 tenemos el siguiente diagrama de flujo:

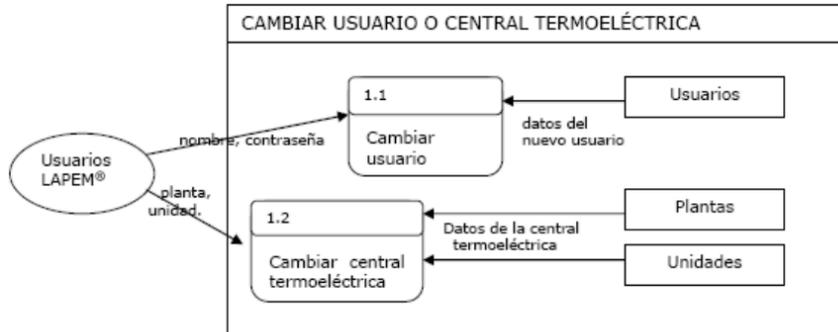


Diagrama 3.2.2.3-Diagrama de Flujo del Nivel 1 de la GUI del Sistema LAPEM-TERMO®

Para el NIVEL 2 tenemos el siguiente diagrama de flujo:

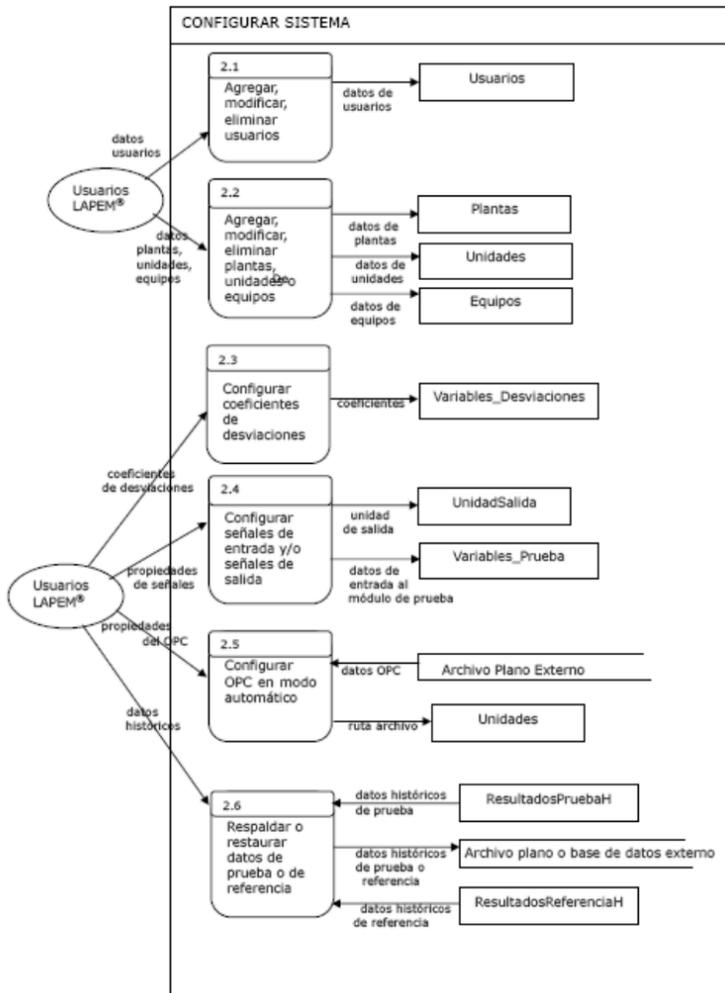


Diagrama 3.2.2.4-Diagrama de Flujo del Nivel 2 de la GUI del Sistema LAPEM-TERMO®

Para el NIVEL 3 tenemos el siguiente diagrama de flujo:

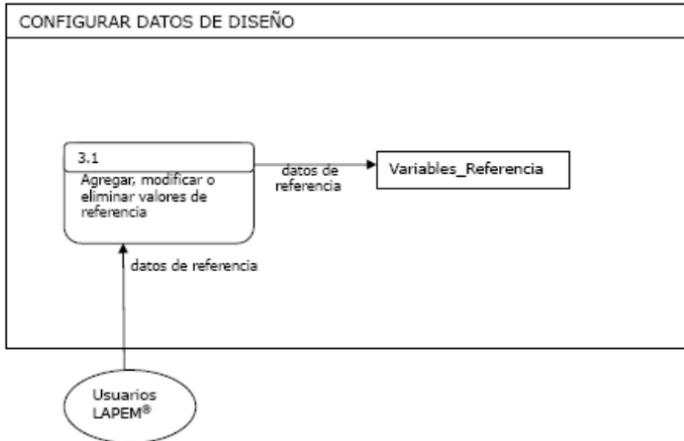


Diagrama 3.2.2.5-Diagrama de Flujo del Nivel 3 de la GUI del Sistema LAPEM-TERMO®

Para el NIVEL 4 tenemos el siguiente diagrama de flujo:

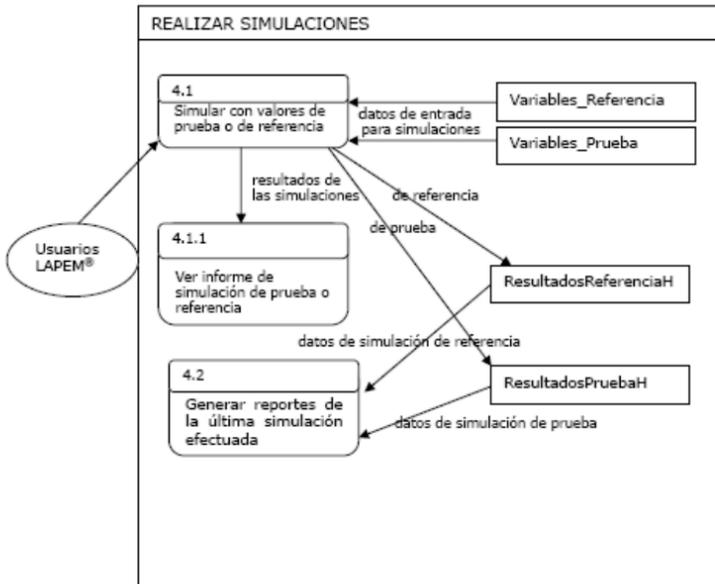


Diagrama 3.2.2.6-Diagrama de Flujo del Nivel 4 de la GUI del Sistema LAPEM-TERMO®

Por último, para el NIVEL 5 tenemos el siguiente diagrama de flujo:

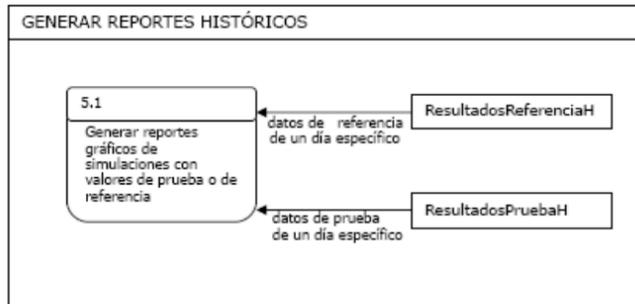


Diagrama 3.2.2.7-Diagrama de Flujo del Nivel 5 de la GUI del Sistema LAPEM-TERMO®

Una vez diseñada la base de datos y su interfaz gráfica para poder administrarla (que son los dos principales componentes de un DBMS), podemos proceder al siguiente capítulo de nuestra tesis, el cual pondrá en práctica todas la ideas planteadas en este CAPÍTULO de diseño, desarrollando tanto la base de datos como la interfaz gráfica, junto a algunos aspectos importantes que debe llevar todo DBMS, entre los que destacan la seguridad, la integridad y completitud de los datos, etc.

CAPÍTULO 4

DESARROLLO

Durante el CAPÍTULO anterior se abordaron varios aspectos muy importantes para la presente tesis, desde un análisis de requerimientos para el sistema LAPEM-TERMO®, hasta el diseño del Sistema de Gestión de Base de Datos para este sistema. En el análisis se trataron los antecedentes del proyecto así como las necesidades que dieron origen al mismo, se explicó en qué consiste y cuáles son sus objetivos. La parte del diseño se enfocó específicamente en dos puntos, el diseño de la base de datos y el diseño de la interfaz gráfica para el manejo de la misma (dos aspectos centrales de cualquier DBMS).

Ahora en este capítulo toca exponer la etapa de desarrollo del Sistema de Gestión de Base de Datos para el Sistema LAPEM-TERMO®.

Empezaremos por el aspecto central de cualquier DBMS: la base de datos. En los capítulos anteriores se mencionó que para el proyecto LAPEM-TERMO® van a existir dos versiones, la versión Stand Alone que será instalada en cada una de las 25 centrales termoeléctricas de México incluidas para este proyecto, y una versión Web que será instalada en un servidor central de la Comisión Federal de Electricidad, la cual permitirá que desde cualquier máquina de la red de INTRANET de CFE se pueda observar el funcionamiento de cualquiera de las 25 centrales antes mencionadas, así como tomar decisiones sobre cada una de ellas.

Como requisito de CFE para la versión Stand Alone la base de datos se desarrollará en Microsoft Access, mientras que para la versión Web, la base de datos se desarrollará en Microsoft SQL Server.

4.1 Desarrollo de la Base de Datos en Microsoft Access

Como primer paso de este capítulo desarrollaremos la base de datos en Microsoft Access versión 2003. Para ello tomaremos como base el modelo de base de datos y el esquema de las relaciones planteados en el CAPÍTULO anterior.

Una ventaja muy importante de este DBMS es que es un programa muy fácil de aprender a utilizar, ya que es un programa muy gráfico, es decir, no se requieren conocimientos de programación ó de SQL (Standard Query Language) para utilizarlo.

Se comenzará construyendo todas y cada una de las 12 tablas (ó relaciones) establecidas anteriormente mediante los esquemas de las relaciones del modelo relacional.

Para ello primero aprenderemos el procedimiento para crear una base de datos nueva en Microsoft Access, lo cual es bastante sencillo, pues solamente tenemos que abrir el programa, dar clic en Archivo y elegimos la opción “Nuevo”:

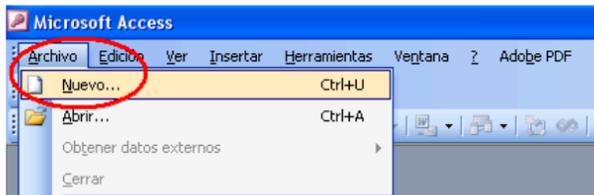


Imagen 4.1.1- Creando una Base de Datos en Access 1

Enseguida se abrirá un pequeño menú, en él seleccionamos la opción “Base de Datos en blanco”, con ello se abrirá un explorador en el que podemos elegir la ubicación donde deseamos guardar la base de datos, así como su nombre.

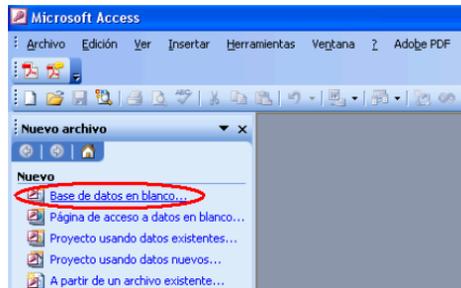


Imagen 4.1.2- Creando una Base de Datos en Access 2

Al hacer esto se abrirá una ventana con una serie de opciones, elegimos la opción “Tabla” lo cual abrirá otro pequeño menú, en él elegimos la opción “Crear una tabla en vista Diseño”:

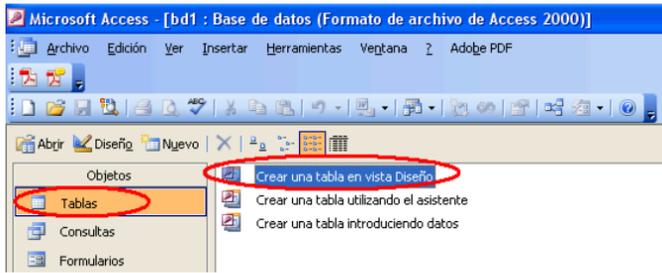


Imagen 4.1.3- Creando una Tabla en Access 1

Enseguida se abrirá una nueva ventana, en ella podemos introducir uno a uno los atributos o campos de nuestra tabla (ó relación), así como el tipo de dato que deseamos asignarle a cada uno, además de poder poner opcionalmente una pequeña descripción de a lo que hace referencia nuestro campo.

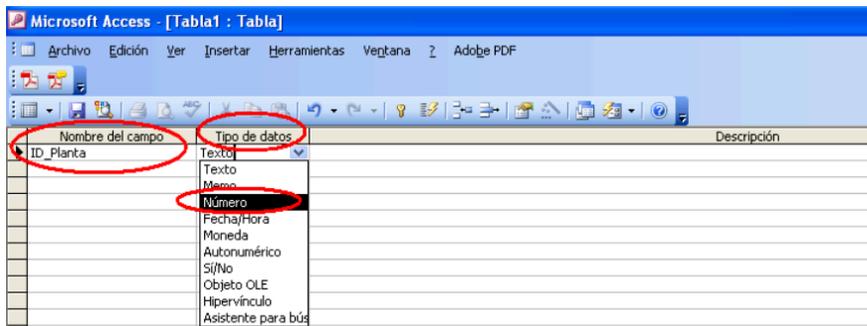


Imagen 4.1.4- Creando una Tabla en Access 2

Al terminar de introducir el campo y si deseamos que éste sea llave primaria, lo único que tenemos que hacer es situarnos sobre el nombre del campo, dar clic derecho con el Mouse y elegir la opción “Clave principal”, inmediatamente después aparecerá una imagen de una llave al costado izquierdo del nombre de nuestro campo, con lo que queda definido como llave primaria.

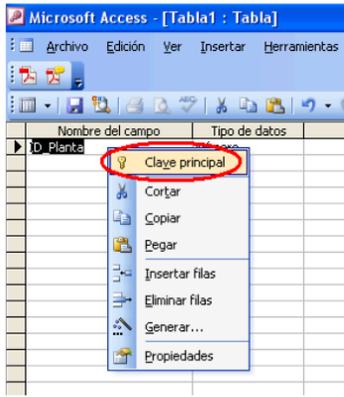


Imagen 4.1.5- Creando una Tabla en Access 3

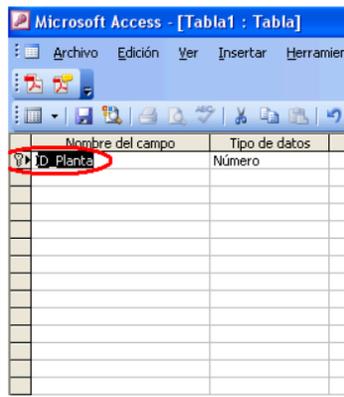


Imagen 4.1.6- Creando una Tabla en Access 4

Al terminar de introducir los datos de todos los campos de nuestra tabla cerramos la ventana, con lo cual se nos preguntara si deseamos guardar los cambios, damos clic en la opción "Sí", al hacer esto se abrirá una pequeña ventana en la que podemos darle el nombre que deseamos a nuestra tabla, al terminar damos clic en la opción "Aceptar".

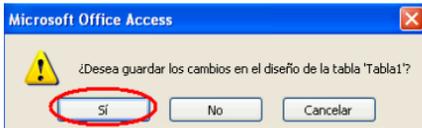


Imagen 4.1.7- Creando una Tabla en Access 5



Imagen 4.1.8- Creando una Tabla en Access 6

Con esto tendremos creada ya nuestra tabla y podemos observarla en nuestra ventana principal de Access:

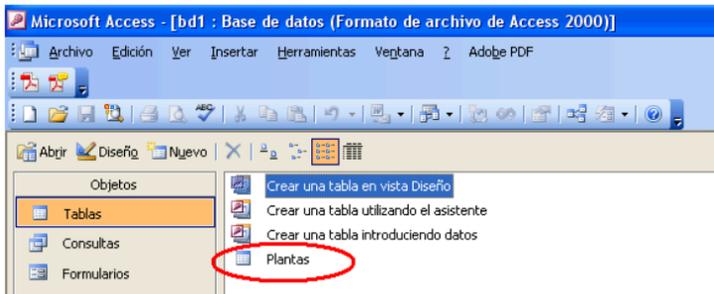


Imagen 4.1.9- Creando una Tabla en Access 7

Para volver a abrirla solamente tenemos que dar doble clic sobre ella, con lo que nuevamente se abrirá la ventana de vista Diseño en la que podemos realizar cambios en los campos o crear algunos que hagan falta. Ahora que si ya deseamos introducir tuplas lo único que tenemos que hacer es dar clic sobre el icono de “vista Hoja de Datos”:

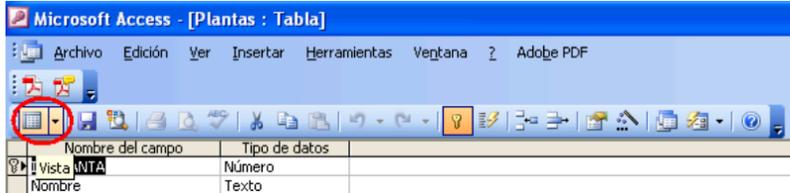


Imagen 4.1.10- Creando una Tabla en Access 7

Al hacer esto se abrirá la vista Hoja de Datos, en la que aparecerá una tabla con los campos creados en la vista Diseño, y en la cual podemos introducir las tuplas que deseemos.

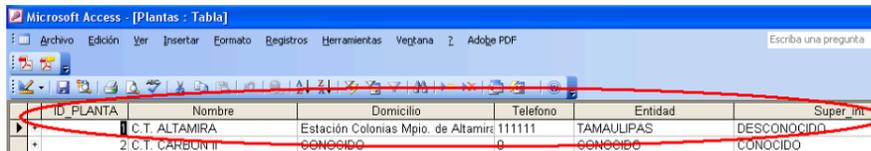
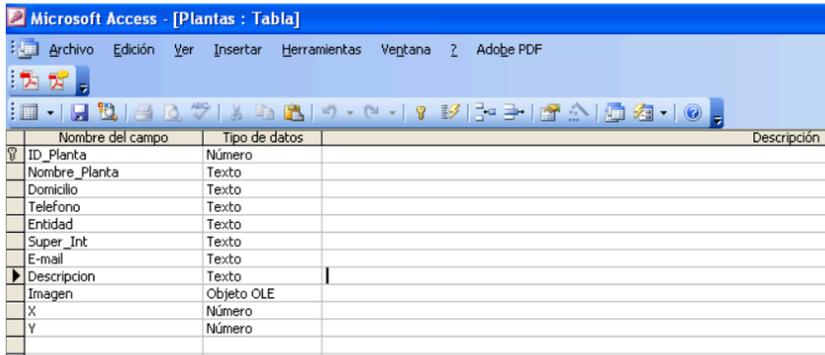


Imagen 4.1.11- Creando una Tabla en Access 8

Ahora que ya sabemos el procedimiento para crear una base de datos y para crear una tabla en Microsoft Access, podemos empezar a crear nuestras 12 relaciones planteadas en el CAPÍTULO anterior de la presente tesis.

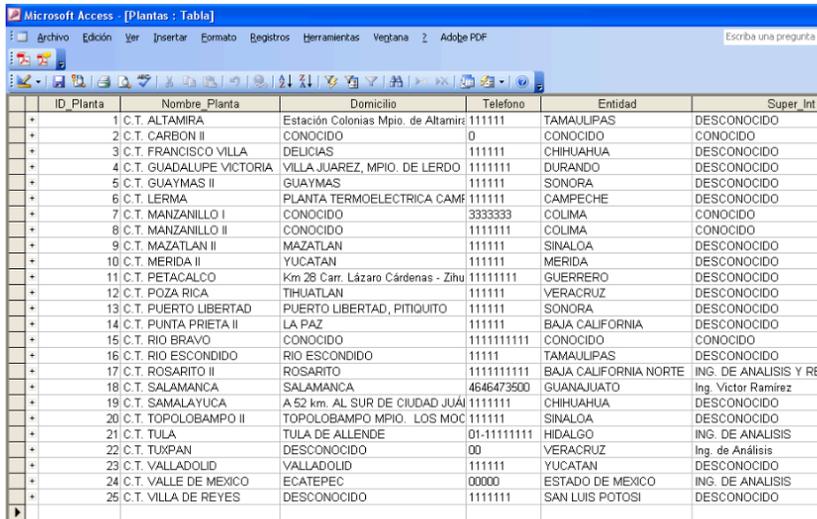
Para empezar con ello y como primer paso, creamos una base de datos nueva y le damos el nombre de “TERMO”. Después procedemos a crear una a una nuestras tablas. La primera es la tabla Plantas y en vista Diseño quedó de la siguiente manera:



| Nombre del campo | Tipo de datos | Descripción |
|------------------|---------------|-------------|
| ID_Planta | Número | |
| Nombre_Planta | Texto | |
| Domicilio | Texto | |
| Telefono | Texto | |
| Entidad | Texto | |
| Super_Int | Texto | |
| E-mail | Texto | |
| Descripción | Texto | |
| Imagen | Objeto OLE | |
| X | Número | |
| Y | Número | |

Imagen 4.1.12- Tabla "Plantas" en vista Diseño

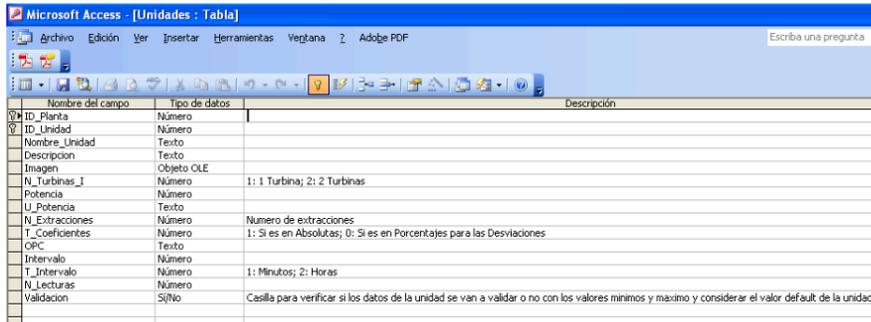
Mientras que en vista Hoja de datos quedó de la siguiente manera:



| ID_Planta | Nombre_Planta | Domicilio | Telefono | Entidad | Super_Int |
|-----------|-------------------------|-------------------------------------|-------------|-----------------------|-----------------------|
| 1 | C.T. ALTAMIRA | Estación Colonias Mpio. de Altamira | 1111111 | TAMAILIPAS | DESCONOCIDO |
| 2 | C.T. CARBON II | CONOCIDO | 0 | CONOCIDO | CONOCIDO |
| 3 | C.T. FRANCISCO VILLA | DEUCIAS | 111111 | CHIHUAHUA | DESCONOCIDO |
| 4 | C.T. GUADALUPE VICTORIA | VILLA JUAREZ, MPIO. DE LERDO | 11111111 | DURANGO | DESCONOCIDO |
| 5 | C.T. GUAYMAS II | GUAYMAS | 111111 | SONORA | DESCONOCIDO |
| 6 | C.T. LERMA | PLANTA TERMOCLECTRICA CAMF | 111111 | CAMPECHE | DESCONOCIDO |
| 7 | C.T. MANZANILLO I | CONOCIDO | 3333333 | COLIMA | CONOCIDO |
| 8 | C.T. MANZANILLO II | CONOCIDO | 11111111 | COLIMA | CONOCIDO |
| 9 | C.T. MAZATLAN II | MAZATLAN | 111111 | SINALOA | DESCONOCIDO |
| 10 | C.T. MERIDA II | YUCATAN | 111111 | MERIDA | DESCONOCIDO |
| 11 | C.T. PETACALCO | Km 28 Carr. Lázaro Cárdenas - Zihu | 11111111 | GUERRERO | DESCONOCIDO |
| 12 | C.T. POZA RICA | TIHUATLAN | 111111 | VERACRUZ | DESCONOCIDO |
| 13 | C.T. PUERTO LIBERTAD | PUERTO LIBERTAD, PITIQUITO | 111111 | SONORA | DESCONOCIDO |
| 14 | C.T. PUNTA PRIETA II | LA PAZ | 111111 | BAJA CALIFORNIA | DESCONOCIDO |
| 15 | C.T. RIO BRAVO | CONOCIDO | 1111111111 | CONOCIDO | CONOCIDO |
| 16 | C.T. RIO ESCONDIDO | RIO ESCONDIDO | 11111 | TAMAILIPAS | DESCONOCIDO |
| 17 | C.T. ROSARITO II | ROSARITO | 1111111111 | BAJA CALIFORNIA NORTE | ING. DE ANALISIS Y RE |
| 18 | C.T. SALAMANCA | SALAMANCA | 4646473500 | GUANAJUATO | Ing. Victor Ramirez |
| 19 | C.T. SAMALAYUCA | A 52 km. AL SUR DE CIUDAD JUÁ | 1111111 | CHIHUAHUA | DESCONOCIDO |
| 20 | C.T. TOPOLOBAMPO II | TOPOLOBAMPO MPIO. LOS MOC | 111111 | SINALOA | DESCONOCIDO |
| 21 | C.T. TULA | TULA DE ALLENDE | 01-11111111 | HIDALGO | ING. DE ANALISIS |
| 22 | C.T. TUXPAN | DESCONOCIDO | 00 | VERACRUZ | Ing. de Análisis |
| 23 | C.T. VALLADOLID | VALLADOLID | 111111 | YUCATAN | DESCONOCIDO |
| 24 | C.T. VALLE DE MEXICO | ECATEPEC | 00000 | ESTADO DE MEXICO | ING. DE ANALISIS |
| 25 | C.T. VILLA DE REYES | DESCONOCIDO | 1111111 | SAN LUIS POTOSI | DESCONOCIDO |

Imagen 4.1.13- Tabla "Plantas" en vista Hoja de Datos

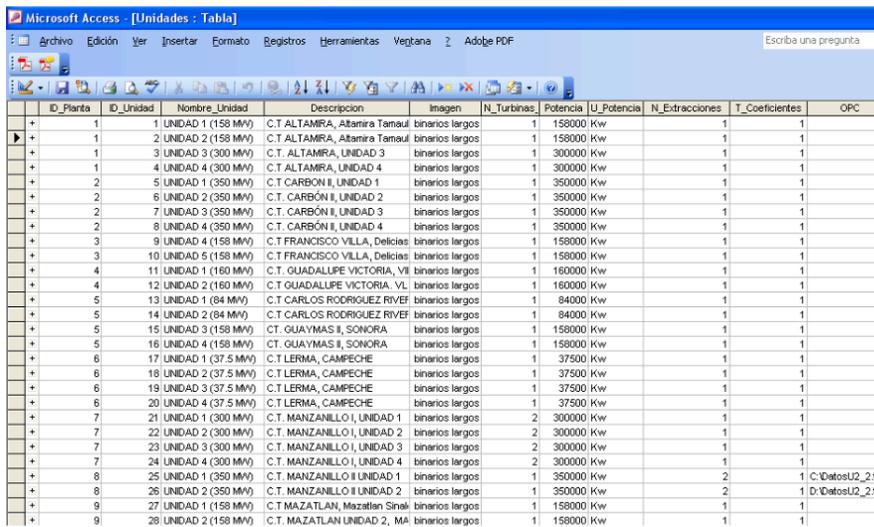
La segunda tabla es la denominada "Unidades", y en vista Diseño quedo así:



| Nombre del campo | Tipo de datos | Descripción |
|------------------|---------------|--|
| ID_Planta | Número | |
| ID_Unidad | Número | |
| Nombre_Unidad | Texto | |
| Descripción | Texto | |
| Imagen | Objeto OLE | |
| N_Turbinas_I | Número | 1: 1 Turbina; 2: 2 Turbinas |
| Potencia | Número | |
| U_Potencia | Texto | |
| N_Extracciones | Número | Numero de extracciones |
| T_Coeficientes | Número | 1: Si es en Absolutas; 0: Si es en Porcentajes para las Desviaciones |
| OPC | Texto | |
| Intervalo | Número | |
| T_Intervalo | Número | 1: Minutos; 2: Horas |
| N_Lecturas | Número | |
| Validacion | Sí/No | Casilla para verificar si los datos de la unidad se van a validar o no con los valores mínimos y máximo y considerar el valor default de la unidad |

Imagen 4.1.14- Tabla "Unidades" en vista Diseño

Ahora que en vista Hoja de Datos quedo así:



| ID_Planta | ID_Unidad | Nombre_Unidad | Descripción | Imagen | N_Turbinas | Potencia | U_Potencia | N_Extracciones | T_Coeficientes | OPC |
|-----------|-----------|--------------------|--------------------------------|-----------------|------------|-----------|------------|----------------|----------------|----------------|
| 1 | 1 | UNIDAD 1 (158 MW) | C.T. ALTAMIRA, Altamira Tamaul | binarios Iergos | 1 | 158000 Kw | | 1 | 1 | |
| 1 | 2 | UNIDAD 2 (158 MW) | C.T. ALTAMIRA, Altamira Tamaul | binarios Iergos | 1 | 158000 Kw | | 1 | 1 | |
| 1 | 3 | UNIDAD 3 (300 MW) | C.T. ALTAMIRA, UNIDAD 3 | binarios Iergos | 1 | 300000 Kw | | 1 | 1 | |
| 1 | 4 | UNIDAD 4 (300 MW) | C.T. ALTAMIRA, UNIDAD 4 | binarios Iergos | 1 | 300000 Kw | | 1 | 1 | |
| 2 | 5 | UNIDAD 1 (350 MW) | C.T. CARBÓN II, UNIDAD 1 | binarios Iergos | 1 | 350000 Kw | | 1 | 1 | |
| 2 | 6 | UNIDAD 2 (350 MW) | C.T. CARBÓN II, UNIDAD 2 | binarios Iergos | 1 | 350000 Kw | | 1 | 1 | |
| 2 | 7 | UNIDAD 3 (350 MW) | C.T. CARBÓN II, UNIDAD 3 | binarios Iergos | 1 | 350000 Kw | | 1 | 1 | |
| 2 | 8 | UNIDAD 4 (350 MW) | C.T. CARBÓN II, UNIDAD 4 | binarios Iergos | 1 | 350000 Kw | | 1 | 1 | |
| 3 | 9 | UNIDAD 4 (158 MW) | C.T. FRANCISCO VILLA, Delicias | binarios Iergos | 1 | 158000 Kw | | 1 | 1 | |
| 3 | 10 | UNIDAD 5 (158 MW) | C.T. FRANCISCO VILLA, Delicias | binarios Iergos | 1 | 158000 Kw | | 1 | 1 | |
| 4 | 11 | UNIDAD 1 (160 MW) | C.T. GUADALUPE VICTORIA, VII | binarios Iergos | 1 | 160000 Kw | | 1 | 1 | |
| 4 | 12 | UNIDAD 2 (160 MW) | C.T. GUADALUPE VICTORIA, VI | binarios Iergos | 1 | 160000 Kw | | 1 | 1 | |
| 5 | 13 | UNIDAD 1 (84 MW) | C.T. CARLOS RODRIGUEZ RIVER | binarios Iergos | 1 | 84000 Kw | | 1 | 1 | |
| 5 | 14 | UNIDAD 2 (84 MW) | C.T. CARLOS RODRIGUEZ RIVER | binarios Iergos | 1 | 84000 Kw | | 1 | 1 | |
| 5 | 15 | UNIDAD 3 (158 MW) | CT. GUAYMAS II, SONORA | binarios Iergos | 1 | 158000 Kw | | 1 | 1 | |
| 5 | 16 | UNIDAD 4 (158 MW) | CT. GUAYMAS II, SONORA | binarios Iergos | 1 | 158000 Kw | | 1 | 1 | |
| 6 | 17 | UNIDAD 1 (37.5 MW) | C.T. LERMA, CAMPECHE | binarios Iergos | 1 | 37500 Kw | | 1 | 1 | |
| 6 | 18 | UNIDAD 2 (37.5 MW) | C.T. LERMA, CAMPECHE | binarios Iergos | 1 | 37500 Kw | | 1 | 1 | |
| 6 | 19 | UNIDAD 3 (37.5 MW) | C.T. LERMA, CAMPECHE | binarios Iergos | 1 | 37500 Kw | | 1 | 1 | |
| 6 | 20 | UNIDAD 4 (37.5 MW) | C.T. LERMA, CAMPECHE | binarios Iergos | 1 | 37500 Kw | | 1 | 1 | |
| 7 | 21 | UNIDAD 1 (300 MW) | C.T. MANZANILLO I, UNIDAD 1 | binarios Iergos | 2 | 300000 Kw | | 1 | 1 | |
| 7 | 22 | UNIDAD 2 (300 MW) | C.T. MANZANILLO I, UNIDAD 2 | binarios Iergos | 2 | 300000 Kw | | 1 | 1 | |
| 7 | 23 | UNIDAD 3 (300 MW) | C.T. MANZANILLO I, UNIDAD 3 | binarios Iergos | 2 | 300000 Kw | | 1 | 1 | |
| 7 | 24 | UNIDAD 4 (300 MW) | C.T. MANZANILLO I, UNIDAD 4 | binarios Iergos | 2 | 300000 Kw | | 1 | 1 | |
| 8 | 25 | UNIDAD 1 (350 MW) | C.T. MANZANILLO II, UNIDAD 1 | binarios Iergos | 1 | 350000 Kw | | 2 | 1 | C:\DatosU2_2.t |
| 8 | 26 | UNIDAD 2 (350 MW) | C.T. MANZANILLO II, UNIDAD 2 | binarios Iergos | 1 | 350000 Kw | | 2 | 1 | D:\DatosU2_2.t |
| 9 | 27 | UNIDAD 1 (158 MW) | C.T. MAZATLAN, Mazatlan Sinal | binarios Iergos | 1 | 158000 Kw | | 1 | 1 | |
| 9 | 28 | UNIDAD 2 (158 MW) | C.T. MAZATLAN UNIDAD 2, MA | binarios Iergos | 1 | 158000 Kw | | 1 | 1 | |

Imagen 4.1.15- Tabla "Unidades" en vista Hoja de Datos

La tabla "Equipos" es la tercera en la lista y una vez creada en vista Diseño se ve así:

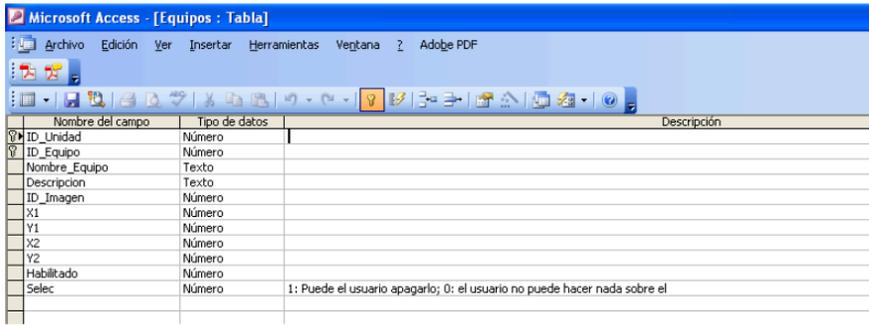


Imagen 4.1.16- Tabla "Equipos" en vista Diseño

En vista Hoja de Datos se ve así:

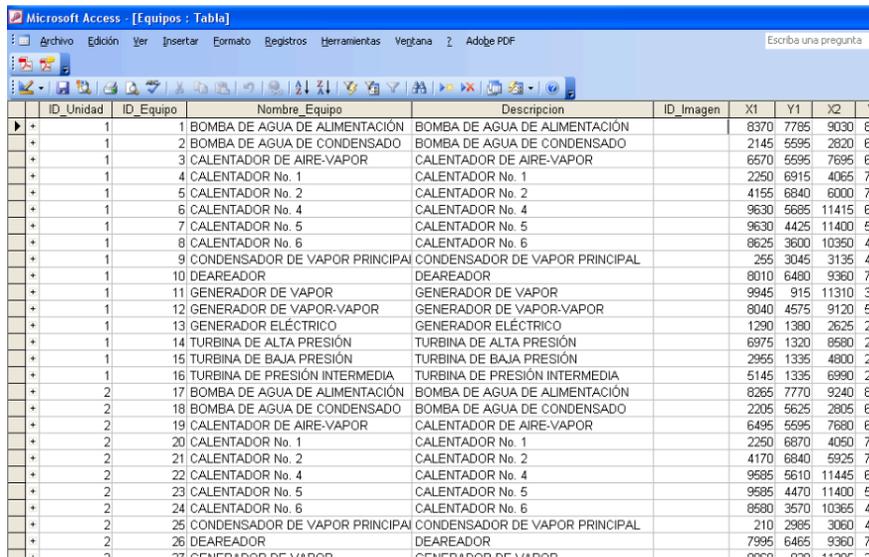


Imagen 4.1.17- Tabla "Equipos" en vista Hoja de Datos

Así como se crearon las 3 tablas anteriores, de la misma forma se crearon las 12 tablas que componen a la base de datos del sistema LAPEM-TERMO®, quedando la base completa de la siguiente forma:

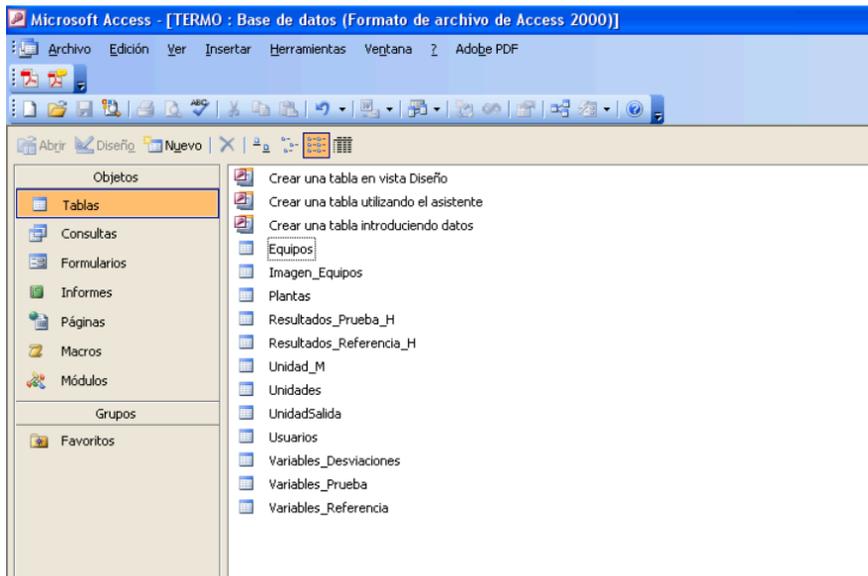


Imagen 4.1.18- Base de Datos del sistema LAPEM-TERMO® con sus 12 tablas.

Una vez creadas las 12 tablas procedemos a relacionarlas, tomando como base el diagrama realizado en Erwin 7.0 en el CAPÍTULO anterior.

La base de datos en Microsoft Access ya con las relaciones entre tablas realizadas, queda de la siguiente forma:

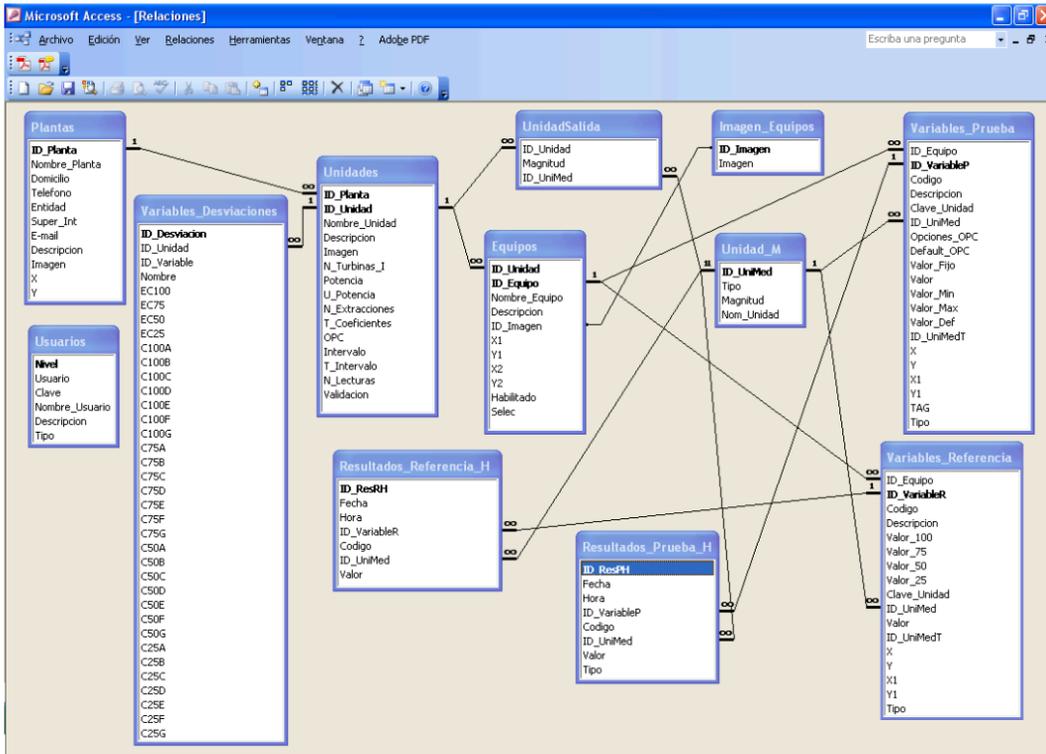


Diagrama 4.1.19- Diagrama de la Base de Datos del sistema LAPEM-TERMO® en Microsoft Access.

Después de analizar este diagrama podemos llegar a una observación muy importante, la cual es que es exactamente el mismo diagrama en comparación al realizado en Erwin 7.0, por lo que podemos decir que la base de datos para el sistema LAPEM-TERMO® fue creada exitosamente en Microsoft Access 2003.

4.2 Desarrollo de la Base de Datos en Microsoft SQL Server

Como segundo paso del presente capítulo desarrollaremos la base de datos para el sistema LAPEM-TERMO® en Microsoft SQL Server 2005. De la misma forma que en el tema anterior, tomaremos como punto de partida el modelo de base de datos y el esquema de las relaciones planteadas en el capítulo tres de ésta tesis.

Una ventaja muy importante con la que cuenta tanto Microsoft SQL Server 2005 como Microsoft Access 2003 es que nos proporciona la capacidad de importar o exportar entre ellos una base de datos, es decir migrar la base de datos de un manejador a otro; por lo que la tarea de crear la base de datos en SQL Server 2005 se simplifica notablemente.

Para lograr esto, exportamos la base de datos creada en Microsoft Access realizando los siguientes pasos:

Primero abrimos nuestra base de datos en Microsoft Access 2003, y en el menú superior de la ventana principal damos clic en "**Herramientas**", al hacer esto se despliega otro pequeño menú, en él elegimos la opción "**Utilidades de la base de datos**"; nuevamente se despliega otro pequeño menú dentro del cual podemos ejecutar el "**Asistente para convertir a SQL Server**".

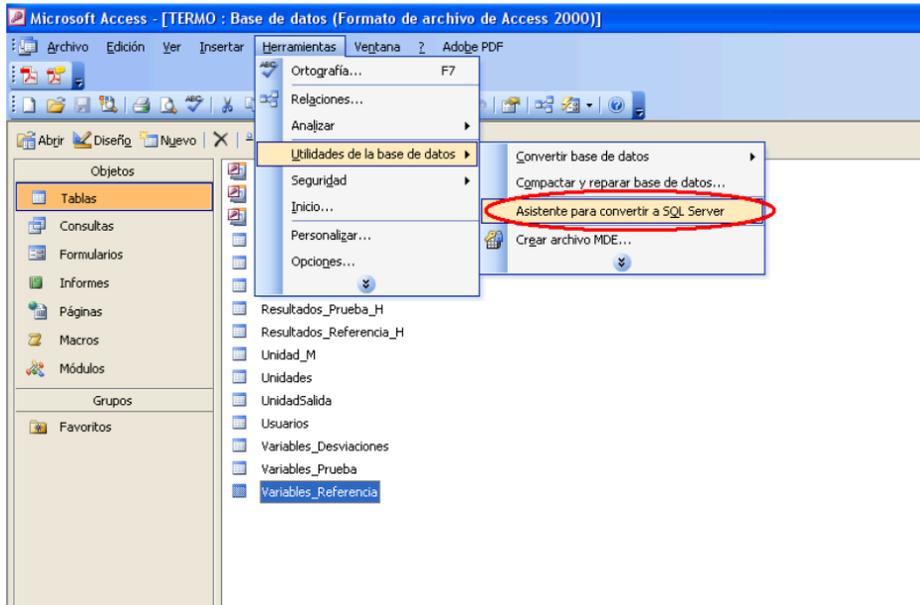


Imagen 4.2.1- Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005.

Al realizar lo anterior se iniciará el asistente para convertir a SQL Server abriéndose una nueva ventana dentro de la cual marcamos la opción **“Crear nueva base de datos”** y damos clic en siguiente.

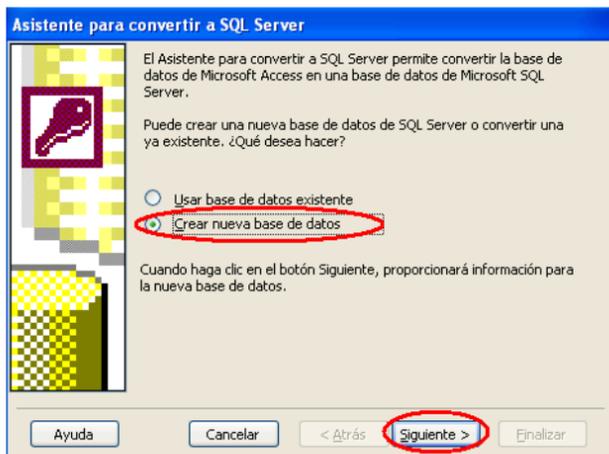


Imagen 4.2.2- Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005.

Posteriormente se nos pide elegir el servidor que deseamos utilizar para esta base de datos, elegimos el predeterminado “*SIS\SQL EXPRESS*”, marcamos la opción “*Usar conexión segura*” e introducimos el nombre que deseamos dar a la nueva base de datos.

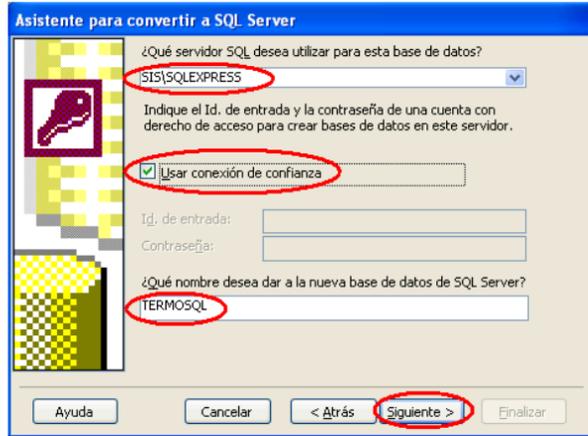


Imagen 4.2.3- Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005.

Después el asistente nos pide las tablas que deseamos agregar a la nueva base de datos, marcamos todas y damos clic en siguiente.

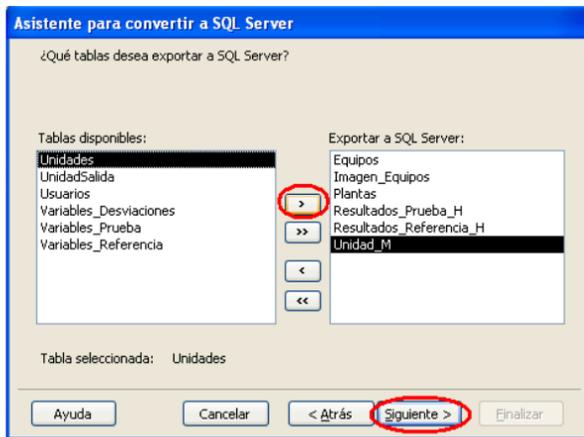


Imagen 4.2.4- Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005.

En seguida el asistente nos hace la aclaración de que se pueden exportar los atributos de la tabla además de los datos, para ello verificamos que estén marcadas las opciones “Índices, Reglas de validación, Valores predeterminados, Relaciones de tablas y Utilizar DRI”. Damos clic en siguiente.

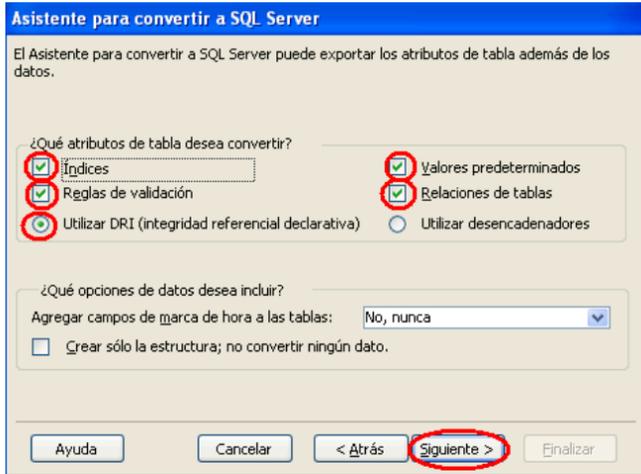


Imagen 4.2.5- Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005.

Por último verificamos que está marcada la opción por default que presenta el asistente la cual es “**Vincular tablas de SQL Server a una aplicación existente**” y damos clic en siguiente.

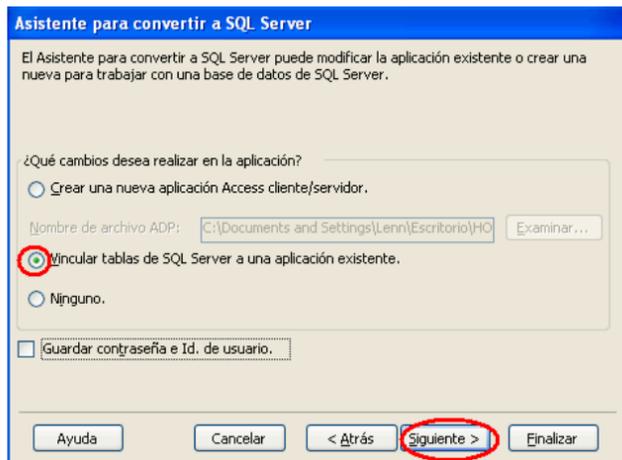


Imagen 4.2.6- Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005.

Damos clic en finalizar.



Imagen 4.2.7- Importando la base de datos de Microsoft Access 2003 a Microsoft SQL Server 2005.

Ahora que ya hemos exportado nuestra base de datos realizada en Microsoft Access podemos empezar a trabajar con ella en Microsoft SQL Server. Para ello primero abrimos el programa y posteriormente abrimos la base de datos teniendo en cuenta el nombre que le dimos al momento de utilizar el asistente para exportarla.

Después de abrirla toca corroborar que las tablas se crearon correctamente, tanto en el diseño de todos y cada uno de los campos así como en la información que contienen cada una de las tablas.

Para ello primero comprobamos que estén creadas todas las 12 tablas de la base de datos del sistema LAPEM-TERMO®, lo cual podemos corroborar en la siguiente imagen:

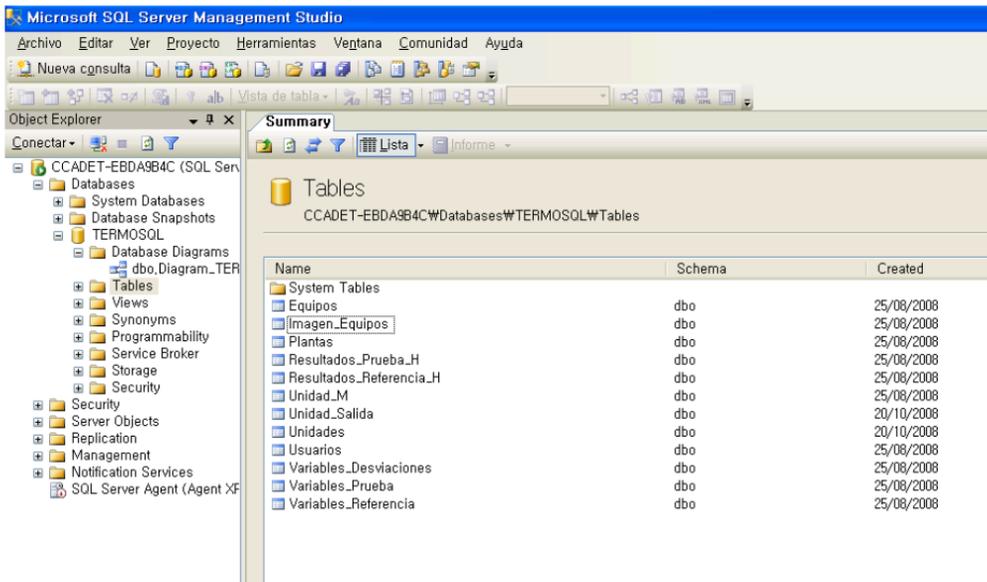


Imagen 4.2.8- Base de Datos del Sistema LAPEM-TERMO® en Microsoft SQL Server 2005.

Ahora toca revisar tabla por tabla para verificar que los campos de cada una de ellas están completos y que tienen asignado el tipo de dato adecuado, así como que cada campo tenga la información correcta, que este completa y que no hayan sufrido cambio alguno en su estructura las llaves primarias y las llaves foráneas.

Empezamos verificando la tabla Plantas, la cual debe contener 11 atributos, donde ID_Planta debe ser la llave primaria y no contiene llaves foráneas. En la siguiente imagen podemos observar que la tabla Plantas cumple correctamente con las características planteadas en el capítulo de diseño:

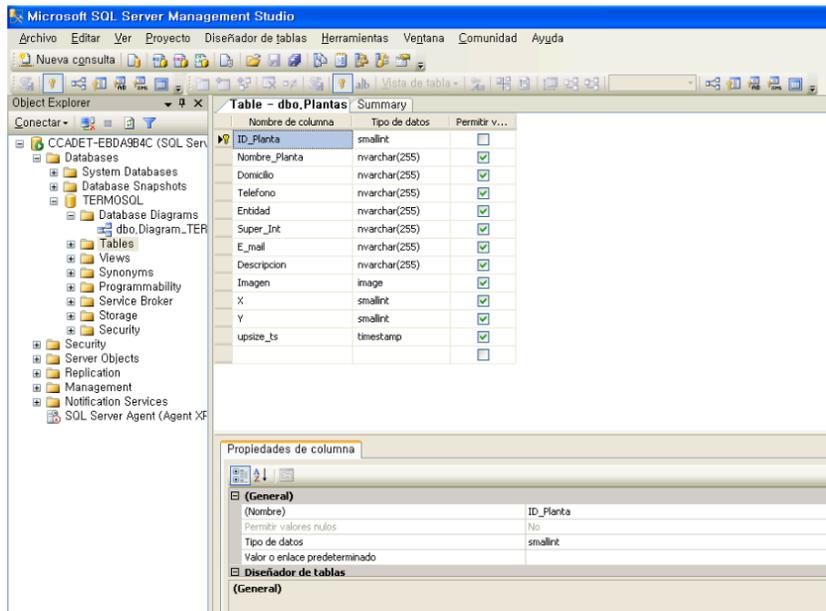


Imagen 4.2.9- Diseño de la tabla Plantas en Microsoft SQL Server 2005.

Y de la misma forma verificamos que toda la información de la tabla Plantas se cargo correctamente.

| ID_Planta | Nombre_Planta | Domicilio | Telefono | Entidad | Super_Int | E_mail | Descripcion |
|-----------|--------------------|---------------------|-------------|------------------|---------------------|------------------|-------------------|
| 1 | C.T. ALTAMIRA | Estación Colonia... | 1111111 | TAMAULIPAS | DESCONOCIDO | 12345@cfe.gob... | ALTAMIRA |
| 2 | C.T. CARBON II | CONOCIDO | 0 | CONOCIDO | CONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 3 | C.T. FRANCISCO... | DELICIAS | 1111111 | CHIHUAHUA | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 4 | C.T. GUADALUP... | VILLA JUAREZ, ... | 1111111 | DURANGO | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 5 | C.T. GUAYMAS II | GUAYMAS | 1111111 | SONORA | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 6 | C.T. LERMA | PLANTA TERMO... | 1111111 | CAMPECHE | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 7 | C.T. MANZANILL... | CONOCIDO | 3333333 | COLIMA | CONOCIDO | 12345@cfe.gob... | MANZANILLO I |
| 8 | C.T. MANZANILL... | CONOCIDO | 1111111 | COLIMA | CONOCIDO | 12345@cfe.gob... | MANZANILLO II |
| 9 | C.T. MAZATLAN II | MAZATLAN | 1111111 | SINALOA | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 10 | C.T. MERIDA II | YUCATAN | 1111111 | MERIDA | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 11 | C.T. PETACALCO | Kim 28 Carr. Láz... | 1111111 | GUERRERO | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 12 | C.T. POZA RICA | TIHUATLAN | 1111111 | VERACRUZ | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 13 | C.T. PUERTO LI... | PUERTO LIBERT... | 1111111 | SONORA | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 14 | C.T. PUNTA PRI... | LA PAZ | 1111111 | BAJA CALIFORNIA | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 15 | C.T. RIO BRAVO | CONOCIDO | 1111111111 | CONOCIDO | CONOCIDO | 12345@cfe.gob... | POTE. EMILIO P... |
| 16 | C.T. RIO ESCON... | RIO ESCONDIDO | 111111 | TAMAULIPAS | DESCONOCIDO | 12345@cfe.gob... | FALTAN DATOS |
| 17 | C.T. ROSARITO II | ROSARITO | 1111111111 | BAJA CALIFORN... | ING. DE ANALIS... | 12345@cfe.gob... | SIN COMENTARIO |
| 18 | C.T. SALAMANCA | SALAMANCA | 4646473500 | GUANAJUATO | Ing. Victor Ramirez | 12345@cfe.gob... | SALAMANCA |
| 19 | C.T. SAMALAYUCA | A 52 km. AL SUR... | 1111111 | CHIHUAHUA | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 20 | C.T. TOPOLOBA... | TOPOLOBAMPO ... | 1111111 | SINALOA | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 21 | C.T. TULA | TULA DE ALLENDE | 01-11111111 | HIDALGO | ING. DE ANALISIS | 12345@cfe.gob... | FRANCISCO PÉ... |
| 22 | C.T. TUXPAN | DESCONOCIDO | 00 | VERACRUZ | Ing. de Análisis | 12345@cfe.gob... | SIN COMENTARIO |
| 23 | C.T. VALLADOLID | VALLADOLID | 1111111 | YUCATAN | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |
| 24 | C.T. VALLE DE M... | ECATEPEC | 00000 | ESTADO DE MEX... | ING. DE ANALISIS | 12345@cfe.gob... | SIN COMENTARIO |
| 25 | C.T. VILLA DE R... | DESCONOCIDO | 1111111 | SAN LUIS POTOSI | DESCONOCIDO | 12345@cfe.gob... | SIN COMENTARIO |

Imagen 4.2.10- Contenido de la tabla Plantas en Microsoft SQL Server 2005.

Enseguida comprobamos la tabla Unidades, la cual debe de contener 15 atributos, en donde ID_Unidad es su llave primaria e ID_Planta es su única llave foránea. En la imagen 4.2.11 comprobamos que ésta tabla cumple con las características planteadas en el diseño.

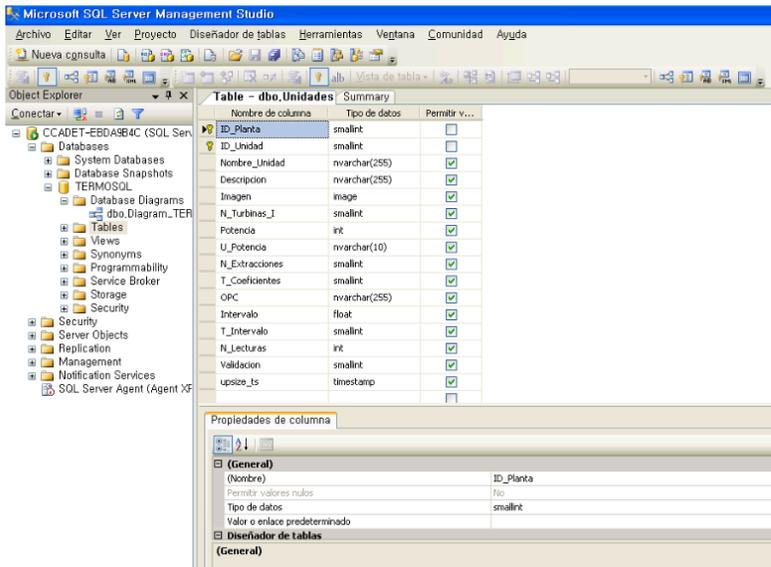


Imagen 4.2.11- Diseño de la tabla Unidades en Microsoft SQL Server 2005.

Y por medio de la imagen 4.2.12 podemos observar que la información se cargó correctamente en la tabla Unidades.

| ID_Planta | ID_Unidad | Nombre_Unidad | Descripción | Imagen | N_Turbinas_I | Potencia | U_Potencia | N_Extracciones |
|-----------|-----------|--------------------|---------------------|---------------|--------------|----------|------------|----------------|
| 1 | 1 | UNIDAD 1 (158 ... | C.T. ALTAMIRA, ... | <Binary data> | 1 | 158000 | Kw | 1 |
| 1 | 2 | UNIDAD 2 (158 ... | C.T. ALTAMIRA, ... | <Binary data> | 1 | 158000 | Kw | 1 |
| 1 | 3 | UNIDAD 3 (300 ... | C.T. ALTAMIRA, ... | <Binary data> | 1 | 300000 | Kw | 1 |
| 1 | 4 | UNIDAD 4 (300 ... | C.T. ALTAMIRA, ... | <Binary data> | 1 | 300000 | Kw | 1 |
| 2 | 5 | UNIDAD 1 (350 ... | C.T. CARBÓN II, ... | <Binary data> | 1 | 350000 | Kw | 1 |
| 2 | 6 | UNIDAD 2 (350 ... | C.T. CARBÓN II, ... | <Binary data> | 1 | 350000 | Kw | 1 |
| 2 | 7 | UNIDAD 3 (350 ... | C.T. CARBÓN II, ... | <Binary data> | 1 | 350000 | Kw | 1 |
| 2 | 8 | UNIDAD 4 (350 ... | C.T. CARBÓN II, ... | <Binary data> | 1 | 350000 | Kw | 1 |
| 3 | 9 | UNIDAD 4 (158 ... | C.T. FRANCISCO, ... | <Binary data> | 1 | 158000 | Kw | 1 |
| 3 | 10 | UNIDAD 5 (158 ... | C.T. FRANCISCO, ... | <Binary data> | 1 | 158000 | Kw | 1 |
| 4 | 11 | UNIDAD 1 (160 ... | C.T. GUADALUP, ... | <Binary data> | 1 | 160000 | Kw | 1 |
| 4 | 12 | UNIDAD 2 (160 ... | C.T. GUADALUP, ... | <Binary data> | 1 | 160000 | Kw | 1 |
| 5 | 13 | UNIDAD 1 (84 MW) | C.T. CARLOS RO... | <Binary data> | 1 | 84000 | Kw | 1 |
| 5 | 14 | UNIDAD 2 (84 MW) | C.T. CARLOS RO... | <Binary data> | 1 | 84000 | Kw | 1 |
| 5 | 15 | UNIDAD 3 (158 ... | CT. GUAYMAS II... | <Binary data> | 1 | 158000 | Kw | 1 |
| 5 | 16 | UNIDAD 4 (158 ... | CT. GUAYMAS II... | <Binary data> | 1 | 158000 | Kw | 1 |
| 6 | 17 | UNIDAD 1 (37.5 ... | C.T. LERMA, CA... | <Binary data> | 1 | 37500 | Kw | 1 |
| 6 | 18 | UNIDAD 2 (37.5 ... | C.T. LERMA, CA... | <Binary data> | 1 | 37500 | Kw | 1 |
| 6 | 19 | UNIDAD 3 (37.5 ... | C.T. LERMA, CA... | <Binary data> | 1 | 37500 | Kw | 1 |
| 6 | 20 | UNIDAD 4 (37.5 ... | C.T. LERMA, CA... | <Binary data> | 1 | 37500 | Kw | 1 |
| 7 | 21 | UNIDAD 1 (300 ... | C.T. MANZANILL... | <Binary data> | 2 | 300000 | Kw | 1 |
| 7 | 22 | UNIDAD 2 (300 ... | C.T. MANZANILL... | <Binary data> | 2 | 300000 | Kw | 1 |
| 7 | 23 | UNIDAD 3 (300 ... | C.T. MANZANILL... | <Binary data> | 2 | 300000 | Kw | 1 |
| 7 | 24 | UNIDAD 4 (300 ... | C.T. MANZANILL... | <Binary data> | 2 | 300000 | Kw | 1 |
| 8 | 25 | UNIDAD 1 (350 ... | C.T. MANZANILL... | <Binary data> | 1 | 350000 | Kw | 2 |

Imagen 4.2.12- Contenido de la tabla Unidades en Microsoft SQL Server 2005.

A continuación verificamos la tabla Equipos, que debe contener 11 atributos, siendo ID_Equipo su llave primaria e ID_Unidad e ID_Imagen sus dos llaves foráneas.

| Nombre de columna | Tipo de datos | Permitir v... |
|-------------------|---------------|-------------------------------------|
| ID_Unidad | smallint | <input type="checkbox"/> |
| ID_Equipo | smallint | <input type="checkbox"/> |
| Nombre_Equipo | nvarchar(255) | <input checked="" type="checkbox"/> |
| Descripción | nvarchar(255) | <input checked="" type="checkbox"/> |
| ID_Imagen | smallint | <input checked="" type="checkbox"/> |
| X1 | float | <input checked="" type="checkbox"/> |
| Y1 | float | <input checked="" type="checkbox"/> |
| X2 | float | <input checked="" type="checkbox"/> |
| Y2 | float | <input checked="" type="checkbox"/> |
| Habilitado | float | <input checked="" type="checkbox"/> |
| Selec | smallint | <input checked="" type="checkbox"/> |
| upose_ts | timestamp | <input checked="" type="checkbox"/> |
| | | <input type="checkbox"/> |

| Propiedades de columna | |
|-------------------------------|-----------|
| (General) | |
| (Nombre) | ID_Unidad |
| Permitir valores nulos | No |
| Tipo de datos | smallint |
| Valor o enlace predeterminado | |
| Diseñador de tablas | |
| (General) | |

Imagen 4.2.13- Diseño de la tabla Equipos en Microsoft SQL Server 2005.

Y por medio de la siguiente imagen podemos observar que la información se cargó correctamente en ésta tabla.

| ID_Unidad | ID_Equipo | Nombre_Equipo | Descripción | ID_Imagen | X1 | Y1 | X2 | Y2 |
|-----------|-----------|------------------|------------------|-----------|------|------|-------|------|
| 1 | 1 | BOMBA DE AGU... | BOMBA DE AGU... | NEGL | 8370 | 7785 | 9030 | 8790 |
| 1 | 2 | BOMBA DE AGU... | BOMBA DE AGU... | NEGL | 2145 | 5595 | 2820 | 6585 |
| 1 | 3 | CALENTADOR D... | CALENTADOR D... | NEGL | 6570 | 5595 | 7695 | 6315 |
| 1 | 4 | CALENTADOR N... | CALENTADOR N... | NEGL | 2250 | 6915 | 4065 | 7560 |
| 1 | 5 | CALENTADOR N... | CALENTADOR N... | NEGL | 4155 | 6840 | 6000 | 7530 |
| 1 | 6 | CALENTADOR N... | CALENTADOR N... | NEGL | 9630 | 5685 | 11415 | 6360 |
| 1 | 7 | CALENTADOR N... | CALENTADOR N... | NEGL | 9630 | 4425 | 11400 | 5160 |
| 1 | 8 | CALENTADOR N... | CALENTADOR N... | NEGL | 8625 | 3600 | 10350 | 4275 |
| 1 | 9 | CONDENSADOR ... | CONDENSADOR ... | NEGL | 255 | 3045 | 3135 | 4800 |
| 1 | 10 | DEAREADOR | DEAREADOR | NEGL | 8010 | 6480 | 9360 | 7440 |
| 1 | 11 | GENERADOR DE... | GENERADOR DE... | NEGL | 9945 | 915 | 11310 | 3390 |
| 1 | 12 | GENERADOR DE... | GENERADOR DE... | NEGL | 8040 | 4575 | 9120 | 5265 |
| 1 | 13 | GENERADOR EL... | GENERADOR EL... | NEGL | 1290 | 1380 | 2625 | 2295 |
| 1 | 14 | TURBINA DE AL... | TURBINA DE AL... | NEGL | 6975 | 1320 | 8580 | 2685 |
| 1 | 15 | TURBINA DE BA... | TURBINA DE BA... | NEGL | 2955 | 1335 | 4800 | 2715 |
| 1 | 16 | TURBINA DE PR... | TURBINA DE PR... | NEGL | 5145 | 1335 | 6990 | 2700 |
| 2 | 17 | BOMBA DE AGU... | BOMBA DE AGU... | NEGL | 8265 | 7770 | 9240 | 8620 |
| 2 | 18 | BOMBA DE AGU... | BOMBA DE AGU... | NEGL | 2205 | 5625 | 2805 | 6585 |
| 2 | 19 | CALENTADOR D... | CALENTADOR D... | NEGL | 6495 | 5595 | 7680 | 6300 |
| 2 | 20 | CALENTADOR N... | CALENTADOR N... | NEGL | 2250 | 6870 | 4050 | 7560 |
| 2 | 21 | CALENTADOR N... | CALENTADOR N... | NEGL | 4170 | 6840 | 5925 | 7515 |
| 2 | 22 | CALENTADOR N... | CALENTADOR N... | NEGL | 9585 | 5610 | 11445 | 6435 |
| 2 | 23 | CALENTADOR N... | CALENTADOR N... | NEGL | 9585 | 4470 | 11400 | 5205 |
| 2 | 24 | CALENTADOR N... | CALENTADOR N... | NEGL | 8580 | 3570 | 10365 | 4305 |
| 2 | 25 | CONDENSADOR ... | CONDENSADOR ... | NEGL | 210 | 2985 | 3060 | 4710 |

Imagen 4.2.14- Contenido de la tabla Equipos en Microsoft SQL Server 2005.

Y de la misma forma que las tres tablas anteriores, se comprobó para las 9 tablas restantes que su diseño fuera el correcto así como que su información estuviera completa y bien almacenada.

Un pequeño inconveniente que surge al exportar la base de datos de Microsoft Access a Microsoft SQL Server, es que el diagrama de la base de datos donde aparecen relacionadas las tablas no se crea en el DBMS destino, por lo que tenemos que realizarlo tal y como lo hicimos en Microsoft Access. Por lo anterior, el siguiente paso fue realizar el diagrama, el cual quedó de la siguiente manera:

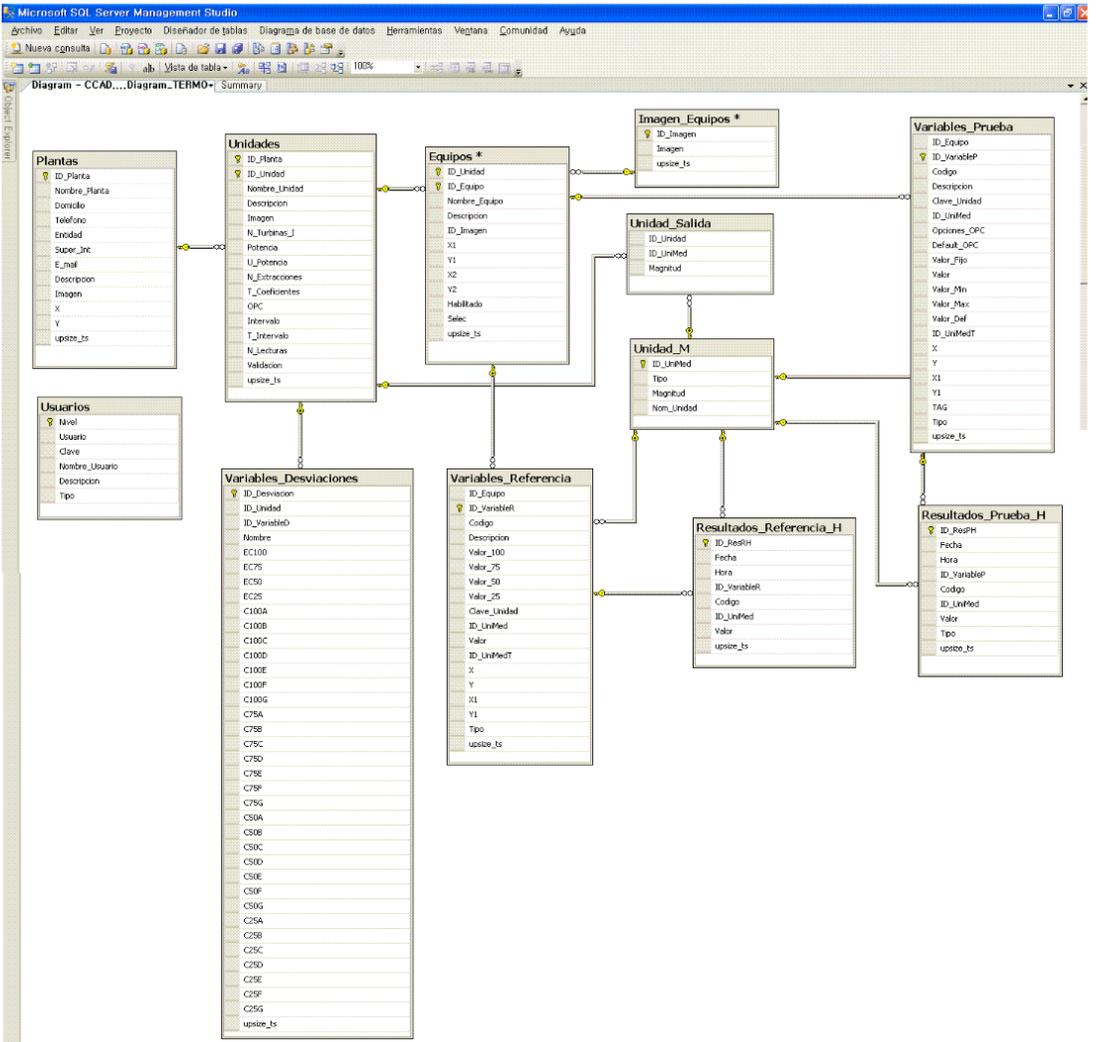


Diagrama 4.2.15 - Diagrama de la Base de Datos del sistema LAPEM-TERMO® en Microsoft SQL Server 2005.

Al observar el diagrama anterior podemos observar que es el mismo que el realizado en Microsoft Access e igual al planteado en el capítulo de diseño, por lo que podemos concluir que la base de datos para el sistema LAPEM-TERMO® se creó correctamente en Microsoft SQL Server 2005.

4.3 Seguridad

Para mantener resguardado el gran valor que tiene la información almacenada en la base de datos del sistema LAPEM-TERMO®, el DBMS debe garantizar la seguridad de los datos frente a usuarios malintencionados y debe proteger la base de datos de ataques de manipulación o destrucción. Para ello el sistema LAPEM-TERMO® cuenta con un sistema de permisos para tres tipos de usuario, el cual fue un requisito que debía contener el sistema de manera obligatoria; los tres tipos de usuarios son los siguientes:

- Usuario tipo “LAPEM-TERMO®”
- Usuario tipo “Directivo”
- Usuario tipo “Ingeniero de Pruebas”

Al arrancar el sistema, se creó un módulo de validación, en el cual se puede elegir el tipo de usuario con el cual se desea entrar al sistema.

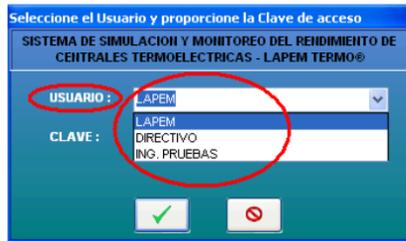


Imagen 4.3.1- Modulo de validación de usuarios del sistema LAPEM-TERMO®.

Y posteriormente se pide una contraseña para poder ingresar y se da clic en el botón Aceptar.

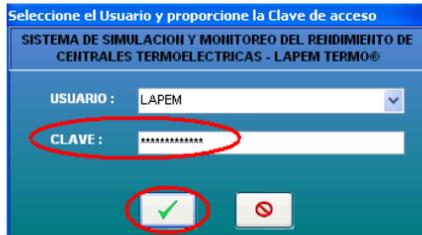


Imagen 4.3.2- Modulo de validación de usuarios del sistema LAPEM-TERMO®.

Para realizar esto, la contraseña ingresada se compara con la contraseña correcta la cual está guardada en la base de datos en la tabla "Usuarios" y en el campo "Clave", si el usuario ingreso correctamente la contraseña dependiendo su tipo, el sistema inicia, pero si no coincide, el sistema inmediatamente se cierra.

Para evitar que usuarios que tengan acceso a la base de datos puedan observar las contraseñas de los demás, la contraseña se guarda de manera cifrada, empleando el método de cifrado SHA1.

4.4 Acceso a la base de datos

Puesto que el usuario no trabajará manualmente con las base de datos sino mediante un gestor (en este caso escrito en C#) para realizar operaciones reales sobre la base de datos (ya sea Access o SQL Server), se requiere de un controlador. Además, también será necesario mantener una conexión con la base de datos, ya que las consultas SQL, así como sus resultados se ejecutarán mediante esa conexión.

En C#, mediante .NET Framework, el acceso a los datos se simplifica a través del conjunto de clases proporcionado por ADO .NET.

ADO .NET es la última versión de la tecnología de acceso a datos de Microsoft, está compuesta por un conjunto de clases insertadas en el .NET Framework, que permite la interacción de manera uniforme con diversas fuentes de datos.

ADO .NET puede usar:

- El proveedor **SQL Server**
- El proveedor **OLEDB**
- El proveedor **ODBC**

Para el sistema de gestión de base de datos LAPEM se utilizará el proveedor OLEDB ya que consideramos que es el más adecuado al ser más eficiente en cuanto a la velocidad de acceso, y soporta mayor número de instrucciones.

OLEDB permite separar los datos de la aplicación que los requiere. Esto se hizo así ya que diferentes aplicaciones requieren acceso a diferentes tipos y almacenes de datos, y no necesariamente desean conocer cómo tener acceso a cierta funcionalidad con métodos de tecnologías específicas.

OleDb está conceptualmente dividido en *consumidores* y *proveedores*; el consumidor es la aplicación que requiere acceso a los datos y el proveedor es el componente de software que expone una interfaz OleDb.

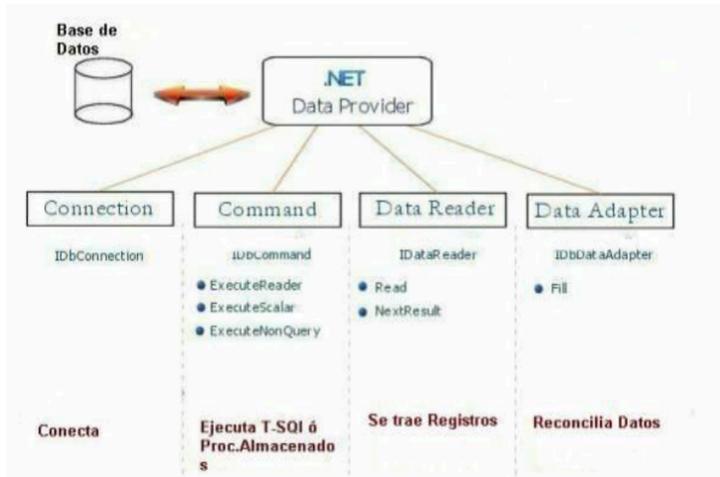


Imagen 4.4.1- Tecnología ADO.NET.

A continuación se describen los pasos para establecer la conexión y acceder a los datos de la base de datos desde la aplicación desarrollada.

1. Establecimiento de la conexión entre la aplicación y la base de datos.

La forma más natural de conectarse al Proveedor OleDb es mediante el uso de `OleDbConnection`.

```
public clsConexion(String pathBase)
{
    //Creamos la cadena de conexion.
    strCnn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
pathBase;
    cnn = new OleDbConnection(strCnn);
    cnn.Open();
} //Fin del constructor
```

El objeto **OleDbConnection** crea una conexión entre la aplicación y el proveedor de base de datos y debe ser inicializado antes de utilizar cualquier otro objeto ADO.NET. Este objeto requiere el proveedor, en este caso *Microsoft.Jet.OLEDB.4.0.* (proveedor para MS Access) Se debe pasar el resto de la cadena de conexión, la cual puede estar contenida en un DSN de ODBC.

Luego se requiere el método `Open()` para abrir la conexión entre la aplicación .NET y el proveedor. Si este método falla, se arrojará una excepción:

'System.Data.OleDb.OleDbException' occurred in system.data.dll

2. Acceso a la base de datos.

Una vez conectados en .NET, hay 3 formas de acceder a los datos de la base de datos.

- `DataReader`
- `DataSet` y `DataTable`
- `ExecuteNonQuery`

2.1 Objeto `DataReader`

La recuperación de datos mediante DataReader implica crear una instancia del objeto OleDbCommand y de un OleDbDataReader, para lo cual se llama a Command.ExecuteReader a fin de recuperar filas desde un origen de datos.

```
    /// Regresa un DataReader que podrá ser leído desde el exterior de
    la clase
    public OleDbDataReader BuscarDR(String strSQL)
    {
        OleDbDataReader resultado;

        OleDbCommand cmdBuscar = new OleDbCommand(strSQL, cnn);

        resultado = cmdBuscar.ExecuteReader();

        return resultado;
    } //Fin del metodo
```

Éste es un objeto que permanece conectado a la base de datos, y es de sólo lectura por lo que no se puede utilizar para actualizar los datos.

2.2 Objetos DataSet y Datatable.

El DataSet de ADO.NET es una representación de datos residente en memoria que proporciona un modelo de programación relacional coherente independientemente del origen de datos que contiene. Un DataSet representa un conjunto completo de datos, incluyendo las tablas que contienen, ordenan y restringen los datos, así como las relaciones entre las tablas.

Hay varias maneras de trabajar con un DataSet, que se pueden aplicar de forma independiente o conjuntamente. Puede:

- Crear mediante programación una DataTable, DataRelation y una Constraint en un DataSet y rellenar las tablas con datos.

- Llenar el DataSet con tablas de datos de un origen de datos relacional existente mediante DataAdapter.
- Cargar y hacer persistente el contenido de DataSet mediante XML.

El DataSet suele ser el resultado de una consulta SQL. Con el DataSet se toman un conjunto de datos y a continuación se desconecta inmediatamente de la Base de Datos para liberar recursos. Esto significa que si se realizan modificaciones en el DataSet será necesario conectarse nuevamente a la base de datos para grabar en ella los cambios. Por lo tanto se debe complementar el DataSet con el DataAdapter.

```
/// Regresa un DataSet que podrá ser leído desde el exterior de la clase
public DataSet BuscarDS(string strSQL)
{
    DataSet Resultado = new DataSet();
    Resultado.Clear();
    OleDbCommand cmdBuscar = new OleDbCommand(strSQL, cnn);
    cmdBuscar.CommandType = CommandType.Text;
    OleDbDataAdapter Adaptador = new OleDbDataAdapter(cmdBuscar);
    Adaptador.Fill(Resultado);
    return Resultado;
} //Fin del metodo
```

Un objeto DataTable, que representa una tabla de datos relacionales en la memoria, se puede crear y usar de manera independiente o lo pueden usar otros objetos de .NET Framework, normalmente como miembro de un objeto DataSet.

```
/// Regresa un DataTable, nos permite por ejemplo asignarlo de forma
automatica a un Grid

    /// Ejemplo: unGrid.DataSource = instanciaAcceso.BuscarDT("Select *
from tblTabla");

    /// </returns>

public DataTable BuscarDT(string strSQL) //OleDbCommand cmdBuscar
{
    DataTable Tabla = new DataTable();

    OleDbCommand cmdBuscar = new OleDbCommand(strSQL, cnn);

    cmdBuscar.CommandType = CommandType.Text;

    OleDbDataAdapter adaptador = new OleDbDataAdapter(cmdBuscar);

    adaptador.Fill(Tabla);

    return Tabla;

} //Fin del método
```

2.3 Método ExecuteNonQuery.

Ejecuta una instrucción SQL en un objeto de conexión.

Se puede utilizar para realizar operaciones de catálogo (por ejemplo, consultar una estructura de una base de datos o crear objetos de base de datos como tablas) o para cambiar la información de una base de datos ejecutando las instrucciones UPDATE, INSERT o DELETE.

Aunque ExecuteNonQuery no devuelva ninguna fila, los parámetros de salida o los valores devueltos asignados a los parámetros se rellenan con datos.

Para las instrucciones UPDATE, INSERT y DELETE, el valor devuelto corresponde al número de filas afectadas por el comando. Para los demás tipos de instrucciones, el valor devuelto es -1.

```
public void Insertar(string strSQL)
{
    OleDbCommand cmdInsertar = new OleDbCommand(strSQL, cnn);
    cmdInsertar.CommandType = System.Data.CommandType.Text;
    cmdInsertar.ExecuteNonQuery();
} //Fin del metodo
```

Este método se encarga de insertar información a la base de datos. El comando debe ser creado desde el exterior de la clase, para evitar generar indeseadas al momento de insertar los datos.

```
public void Eliminar(OleDbCommand cmdEliminar)
{
    cmdEliminar.Connection = cnn;
    cmdEliminar.ExecuteNonQuery();
} //Fin del metodo
```

Este método se encarga de eliminar información desde la base de datos, es necesario el uso de un OleDbCommand para evitar que se generen consultas anidadas después de realizar la operación de eliminar.

```
public void Actualizar(string strSQL)
{
    OleDbCommand cmdUpdate = new OleDbCommand(strSQL, cnn);
    cmdUpdate.CommandType = CommandType.Text;
    cmdUpdate.ExecuteNonQuery();
} //Fin del metodo
```

Este método se encarga de actualizar la base de datos.

3. Operaciones con la base de datos.

Ahora que ya se tiene el código para la conexión y acceso a la base de datos, se procede a programar las interfaces y cuando sea necesario operar con la base de datos, se utilizarán los métodos antes definidos.

Al momento de correr el programa, luego del login, el primer paso es seleccionar la central termoeléctrica. Cuando se hace la selección queda en memoria la planta y unidad de trabajo, y se toma como base esa información para realizar todas las operaciones de búsqueda, lectura y escritura.

A continuación describiremos algunos módulos en donde se pueden observar cómo funciona cada uno de las operaciones.

Módulo de configuración:

Módulos de administración de plantas, unidades, equipos y usuarios:

Si bien todos los módulos del sistema, se comunican con la base de datos y realizan diferentes operaciones sobre ella, estos son los módulos en los que se distinguen perfectamente todas las operaciones.

El Sistema de Simulación y Monitoreo del Rendimiento de Centrales Termoeléctricas cuenta con 25 plantas termoeléctricas de México. Cada una de ellas tiene cierto número de unidades, y éstas cuentan a su vez con un determinado número de equipos.

El módulo de administración de plantas permite realizar cuatro operaciones, cada una de las cuales utiliza un método definido en el módulo de conexión para ejecutar el comando SQL correspondiente:

- Visualizar Plantas: esta opción muestra todas las plantas dadas de alta en el sistema así como sus datos perfil.

```
this.c = new DataTable();
```

```
this.c = TERMO.frmPrincipal.Acceso.BuscarDT("SELECT * FROM Plantas ORDER  
BY Plantas.ID_Planta");
```

- Modificar Planta: esta opción sirve para modificar los datos de perfil de una planta en particular.

```
TERMO.frmPrincipal.Acceso.Insertar("UPDATE Plantas SET " + Criterio +  
this.c.Rows[this.Cont]["ID_Planta"]);
```

- Crear Nueva Planta: con esta opción se da de alta una nueva planta en el sistema.

```
TERMO.frmPrincipal.Acceso.Insertar("INSERT INTO Plantas (ID_Planta,  
Nombre, Domicilio, Entidad, Telefono, Super_int,Email,Descripcion) VALUES  
(" + Criterio);
```

- Eliminar Planta: con esta opción se elimina una planta en particular que exista en el sistema.

```
TERMO.frmPrincipal.Acceso.Eliminar(new OleDbCommand("DELETE FROM Plantas  
WHERE Plantas.Nombre = '" + this.txt1.Text + "'"));
```

La programación para los módulos de usuarios, unidades y equipos se procedió de igual manera.

La manipulación de la base de datos desde el sistema LAPEM se realiza desde el menú de configuración, y se escoge el módulo a modificar.

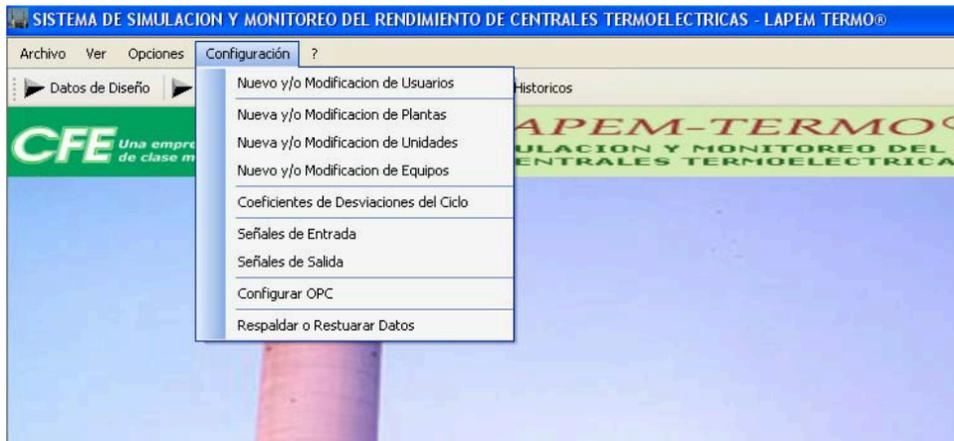


Imagen 4.4.2- Menú de configuración del sistema LAPEM-TERMO®

Si escogemos el módulo de plantas, aparece la siguiente pantalla, desde el cual se realizan las operaciones de visualizar, crear, eliminar y modificar.



Imagen 4.4.3- Menú de configuración de Plantas del sistema LAPEM-TERMO®

Módulo de Reportes:

Si bien todos los módulos del sistema se comunican constantemente con la base de datos, describiremos el módulo de reportes para poder apreciar el uso de las consultas SQL.

Estos reportes despliegan la información acorde a la última simulación realizada. Se crean cuatro reportes:

- Datos Generales
- Eficiencia de equipos
- Reporte General
- Reporte de equipos

Como se trata de reportes, la única operación que se realiza es de consulta:

Consulta para datos de referencia:

```
DTr = frmPrincipal.Acceso.BuscarDT("SELECT CODIGO, VALOR, ID_UNIMEDT, NOMBRE  
FROM Equipos LEFT JOIN Variables_Referencia ON Equipos.ID_EQUIPO =  
Variables_Referencia.ID_EQUIPO WHERE Equipos.ID_UNIDAD=" +  
frmPrincipal.ID_Unidad.ToString());
```

Consulta para datos de prueba:

```
DTP = frmPrincipal.Acceso.BuscarDT("SELECT CODIGO, VALOR, NOMBRE FROM  
Equipos LEFT JOIN Variables_Prueba ON Equipos.ID_EQUIPO =  
Variables_Prueba.ID_EQUIPO WHERE Equipos.ID_UNIDAD=" +  
frmPrincipal.ID_Unidad.ToString());
```

Éstas son las sentencias SQL que se requieren para los cuatro reportes. Los datos extraídos se almacenan en los datatable asignados, y manipulamos esos datos mediante código, para desplegar la información requerida en cada uno de los cuatro reportes. Por ejemplo para el reporte de datos generales de referencia se tiene el siguiente segmento de código:

```
for (i = 0; i < DTr.Rows.Count; i++)
{
    switch (DTr.Rows[i]["CODIGO"].ToString())
    {

#region // DATOS GENERALES

        case "W300":

        case "RTD":

        case "RTI":

        case "CEV":

        case "CEC":

        case "CTU":

            (Doc1.ReportDefinition.ReportObjects["r" +
DTr.Rows[i]["CODIGO"].ToString()] as TextObject).Text =
Convert.ToDouble(DTr.Rows[i]["VALOR"]).ToString("#####0.00");

            (Doc1.ReportDefinition.ReportObjects["u" +
DTr.Rows[i]["CODIGO"].ToString()] as TextObject).Text =
frmPrincipal.Simbologia_US[Convert.ToInt32(DTr.Rows[i]["ID_UNIMEDT"]
) - 1].ToString();

            break;

#endregion

#region // DESVIACIONES

        case "P1":

        case "T1":

        case "T2":

        case "P3P2":

        case "P6":
```

```
case "G122":  
  
case "G123":  
  
case "G40":  
  
case "T47":  
  
case "G112":  
  
case "G113":  
  
case "G114":  
  
case "HP":  
  
case "IP":  
  
case "LP":  
  
    (Doc1.ReportDefinition.ReportObjects["r" +  
    DTr.Rows[i]["CODIGO"].ToString()] as TextObject).Text =  
    Convert.ToDouble(DTr.Rows[i]["VALOR"]).ToString("#####0.00");  
  
    (Doc1.ReportDefinition.ReportObjects["u" +  
    DTr.Rows[i]["CODIGO"].ToString()] as TextObject).Text =  
    frmPrincipal.Simbologia_US[Convert.ToInt32(DTr.Rows[i]["ID_UNIMEDT"  
    ]) - 1].ToString();  
  
    break;  
  
#endregion  
  
default:  
  
    break;  
  
}
```

Y el resultado se muestra utilizando Cristal Reports, quedando de la siguiente manera.

CENTRAL TERMoeLECTRICA MANZANILLO II
 FECHA: Martes, 11 Noviembre, 2011 HORA: 12:56:06 pm
 CONDICIONES DE OPERACION Vapor a los CAVs de Extraccion #5 , GVV de Extraccion #5 y EVA de Vapor Auxiliar
 CONDICIONES DE OPERACION El DREN de los CAVs al Condensador , El GVV al Condensador y La EVA al Deareador

| DATOS GENERALES | | | | | | |
|------------------------|-----------------------------------|-------|----------|------------|-----------|------------|
| No | ELEMENTO | CLAVE | U. M. | REFERENCIA | PRUEBA | DIFERENCIA |
| 1 | POTENCIA ELECTRICA | W300 | kW | 321839.00 | 321839.00 | 0.00 |
| 2 | REGIMEN TERMICO DIRECTO | RTD | kcal/kWh | 531.93 | 2269.52 | -1737.59 |
| 3 | REGIMEN TERMICO INDIRECTO | RTI | kcal/kWh | 521.11 | 2252.34 | -1731.23 |
| 4 | CONSUMO ESPECIFICO DE VAPOR | CEV | kg/kWh | 3.11 | 3.05 | 0.06 |
| 5 | CONSUMO ESPECIFICO DE COMBUSTIBLE | CEC | kg/kWh | 0.22 | 0.23 | -0.01 |
| 6 | CONSUMO TERMICO UNITARIO | CTU | kcal/kWh | 1944.10 | 2009.57 | -65.47 |

Imagen 4.4.4- Reporte de Datos Generales de Referencia del sistema LAPEM-TERMO®

Módulo de respaldos:

Este módulo sirve para respaldar las tablas históricas, que son:

- ResultadosReferenciaH
- ResultadosPruebaH.

El respaldo y restauración se realiza eligiendo la opción Respaldar o restaurar datos del menú de configuración, y se puede hacer de dos formas:

1. Respaldar o restaurar en formato MS ACCESS.



Imagen 4.4.5- Menú para respaldar en Access en el sistema LAPEM-TERMO®

En este caso se respaldará la información de las dos tablas como un nuevo archivo mdb (base de datos de MS ACCESS), por lo que cada vez que se seleccione este formato de respaldo se creará una nueva base de datos así como las dos tablas para luego insertar en ellas los datos de las tablas originales.

El procedimiento es como sigue:

1.1 Crear la base de datos.

```
ADOX.CatalogClass cat = new ADOX.CatalogClass();  
  
cat.Create("Provider=Microsoft.Jet.OLEDB.4.0;Data Source = " +  
Sdialog.FileName + ";Jet OLEDB:Engine Type=5");  
  
cat = null;
```

1.2 Abrir la base y crear las tablas.

```
Con = new OleDbConnection("Provider = Microsoft.Jet.OLEDB.4.0;Data Source = " + Sdialog.FileName);  
  
Con.Open();  
  
Cadena = "CREATE TABLE ResultadosReferenciaH(ID INTEGER, FECHA DATE, " +  
"HORA VARCHAR(5), ID_VARIABLE INTEGER, CODIGO VARCHAR, " +  
"ID_UNIMED INTEGER, VALOR DOUBLE, CONSTRAINT key1 PRIMARY KEY(ID))";  
  
cmd = new OleDbCommand(Cadena, Con);  
  
cmd.ExecuteNonQuery();  
  
Cadena = "CREATE TABLE ResultadosPruebaH(ID INTEGER, FECHA DATE, " +  
"HORA VARCHAR(5), ID_VARIABLE INTEGER, CODIGO VARCHAR, " +  
"ID_UNIMED INTEGER, VALOR DOUBLE, TIPO VARCHAR(1), CONSTRAINT key1 PRIMARY  
KEY(ID))";  
  
cmd = new OleDbCommand(Cadena, Con);  
  
cmd.ExecuteNonQuery();
```

Es importante que los nombres y las propiedades de los campos de la nueva tabla sea exactamente igual a las originales, para evitar conflictos al momento de la restauración.

Los pasos siguientes se realizan para cada una de las tablas.

1.3 Seleccionar los registros de las tablas originales.

Para la tabla de Referencias:

```
Dt = new DataTable();  
Dt = frmPrincipal.Acceso.BuscarDT("SELECT * FROM ResultadosReferenciaH");
```

Para la tabla de Pruebas:

```
Dt = new DataTable();  
Dt = frmPrincipal.Acceso.BuscarDT("SELECT * FROM ResultadosPruebaH");
```

1.4 Insertar los registros a seleccionados a la tabla de respaldo, y eliminar registros respaldados de la tabla original.

Es necesario aclarar que el respaldo no debe incluir todos los registros de la tabla, sino que respalda solamente los registros cuya fecha sea anterior a los 30 días a partir de la fecha de corte:

Para la tabla de ResultadosReferenciaH:

```
for (i = 0; i < Dt.Rows.Count; i++)
{
    FechaFila =
    Convert.ToDateTime(Dt.Rows[i]["FECHA"].ToString());

    if (FechaFila > FechaCorte)

    {
        //En este caso, NO se realiza ninguna operación
    }

    else

    {
        //INSERTAMOS REGISTROS

        Cadena = "INSERT INTO ResultadosReferenciaH(ID,FECHA,HORA," +
        "ID_VARIABLE,CODIGO,ID_UNIMED,VALOR) VALUES (" +
        Dt.Rows[i]["ID"].ToString() + "," +
        Convert.ToDateTime(Dt.Rows[i]["FECHA"]).ToString("dd/MM/yyyy"
        ) + "," +
        Convert.ToDateTime(Dt.Rows[i]["HORA"]).ToString("HH:mm") +
        "," + Dt.Rows[i]["ID_VARIABLE"].ToString() + "," +
        Dt.Rows[i]["CODIGO"].ToString() + "," +
        Dt.Rows[i]["ID_UNIMED"].ToString() + "," +
        Dt.Rows[i]["VALOR"].ToString() + ")";

        cmd = new OleDbCommand(Cadena, Con);

        cmd.ExecuteNonQuery();

        //COMO EL REGISTRO SE RESPALDO, SE BORRA DE LA BASE EN USO

        frmPrincipal.Acceso.Eliminar(new OleDbCommand("DELETE FROM
        ResultadosReferenciaH WHERE ID=" + Dt.Rows[i][0]));

    }
}
```

De igual forma, para la tabla ResultadosPruebaH:

```
for (int j = 0; j < Dt.Rows.Count; j++)
{
    FechaFila =
    Convert.ToDateTime(Dt.Rows[j]["FECHA"].ToString());

    if (FechaFila > FechaCorte)
    {
        // En este caso, , NO se realiza ninguna operación
    }
    else
    {
        //INSERTAMOS REGISTROS

        Cadena = "INSERT INTO ResultadosPruebaH(ID,FECHA,HORA,"
+
        "ID_VARIABLE,CODIGO,ID_UNIMED,VALOR,TIPO) VALUES (" +
        Dt.Rows[j]["ID"].ToString() +
        "," +
        Convert.ToDateTime(Dt.Rows[j]["FECHA"]).ToString("dd/MM
/yyyY") + "," +
        Convert.ToDateTime(Dt.Rows[j]["HORA"]).ToString("HH:mm"
) + "," +
        Dt.Rows[j]["ID_VARIABLE"].ToString() + "," +
        Dt.Rows[j]["CODIGO"].ToString() + "," +
        Dt.Rows[j]["ID_UNIMED"].ToString() + "," +
        Dt.Rows[j]["VALOR"].ToString() + "," +
        Dt.Rows[j]["TIPO"].ToString() + ")";

        cmd = new OleDbCommand(Cadena, Con);

        cmd.ExecuteNonQuery();

        //COMO EL REGISTRO SE RESPALDO, SE BORRA DE LA BASE EN USO

        frmPrincipal.Acceso.Eliminar(new OleDbCommand("DELETE
FROM ResultadosPruebaH WHERE ID=" + Dt.Rows[j][0]));
    }
}
```

2. Respaldo en archivo plano.

En este caso, no es necesario crear nuevas tablas. Solamente se seleccionan los datos de la tabla a respaldar, y se realizan las conversiones de formato necesarios para ser almacenados en un archivo plano.

Los pasos para realizar este respaldo son:

2.1 Seleccionar la tabla a respaldar



Imagen 4.4.6- Menú para respaldar en Archivo Plano en el sistema LAPEM-TERMO®

2.2 Seleccionar los registros a respaldar.

Para la tabla de Referencias:

```
Dt = new DataTable();  
Dt = frmPrincipal.Acceso.BuscarDT("SELECT * FROM ResultadosReferenciaH");
```

Para la tabla de Pruebas:

```
Dt = new DataTable();  
Dt = frmPrincipal.Acceso.BuscarDT("SELECT * FROM ResultadosPruebaH");
```

2.3 Se guardan los datos en un archivo plano y se eliminan los registros respaldados de la tabla original.

Para la tabla de Referencias:

```
StreamWriter sw = new StreamWriter(Sdialog.FileName);  
  
for (i = 0; i < Dt.Rows.Count; i++)  
{  
    FechaFila =  
    Convert.ToDateTime(Dt.Rows[i]["FECHA"].ToString());  
  
    if (FechaFila > FechaCorte)  
        Dt.Rows[i].Delete();  
  
    else  
    {  
        sw.Write(Convert.ToDateTime(Dt.Rows[i]["FECHA"]).ToString("dd/MM/yyyy") + "," +  
        Convert.ToDateTime(Dt.Rows[i]["HORA"]).ToString("HH:mm") + "," +  
        Dt.Rows[i]["ID_VARIABLE"].ToString() + "," +  
        Dt.Rows[i]["CODIGO"].ToString() + "," +  
        Dt.Rows[i]["ID_UNIMED"].ToString() + "," +  
        Dt.Rows[i]["VALOR"].ToString());  
  
        sw.WriteLine();  
  
        frmPrincipal.Acceso.Eliminar(new OleDbCommand("DELETE  
FROM ResultadosReferenciaH WHERE ID=" +  
Dt.Rows[i][0]));  
    }  
}
```

```
    }  
}  
sw.WriteLine();  
sw.Close();
```

Para la tabla de Pruebas:

```
StreamWriter sw = new StreamWriter(Sdialog.FileName);  
for (i = 0; i < Dt.Rows.Count; i++)  
{  
    FechaFila =  
    Convert.ToDateTime(Dt.Rows[i]["FECHA"].ToString());  
    if (FechaFila > FechaCorte)  
        Dt.Rows[i].Delete();  
    else  
    {  
        sw.Write(Convert.ToDateTime(Dt.Rows[i]["FECHA"]).ToString("dd  
/MM/yyyy") + "," +  
Convert.ToDateTime(Dt.Rows[i]["HORA"]).ToString("HH:mm") +  
"," + Dt.Rows[i]["ID_VARIABLE"].ToString() + "," +  
Dt.Rows[i]["CODIGO"].ToString() + "," +  
Dt.Rows[i]["ID_UNIMED"].ToString() + "," +  
Dt.Rows[i]["VALOR"].ToString() + "," +  
Dt.Rows[i]["TIPO"].ToString());  
        sw.WriteLine();  
        frmPrincipal.Acceso.Eliminar(new OleDbCommand("DELETE FROM  
ResultadosPruebaH WHERE ID=" + Dt.Rows[i][0]));  
    }  
}  
sw.WriteLine();  
sw.Close();
```

CAPÍTULO 5

PRUEBA DE SOFTWARE E IMPLEMENTACIÓN DEL SISTEMA

La gestión de la base de datos, que fue nuestro objeto de enfoque en esta tesis, es solamente una parte de un proyecto más grande, no es posible verificar su correcto funcionamiento de manera independiente. Por lo que en este capítulo se describirá el proceso de prueba y referencia de LAPEM-TERMO en modo automático, tal como se implementó en la unidad de Manzanillo. Si los resultados obtenidos con la simulación son correctos, podremos demostrar que el sistema de gestión de bases de datos se desarrolló adecuadamente.

Lo primero que se hizo fue configurar las variables de entrada, de salida y los coeficientes de desviaciones. Para ello, entramos al menú de configuración y se eligió la opción correspondiente:



Imagen 5.1- Menú Configuración del sistema LAPEM-TERMO®

1.1. Señales de entrada.

Con esta opción se configuraron las variables de entrada, para las cuales se definió su origen y las unidades de medida. Para definir las unidades de medida, primero se tuvo que identificar las señales disponibles en la planta.

Por ejemplo para el generador de vapor-vapor, se observa que sus variables de entrada tienen diferentes orígenes.

Configuración de Señales - C.T. MANZANILLO II - UNIDAD 2 (350 MW)

| UNIDAD SELECCIONADA | VARIABLES DE ENTRADA AL MODULO DE PRUEBA | | | | | |
|-------------------------------|--|--|---------------|--------------|--------------|---------------|
| UNIDAD 2 (350 MW) | VARIABLE | DESCRIPCION | ORIGEN | VALOR MINIMO | VALOR MAXIMO | VALOR DEFAULT |
| BOMBA DE AGUA DE ALIMENTACIÓN | G113 | FLUJO DE VAPOR AL GWV | CALCULADO | 0 | 0 | 0 |
| BOMBA DE AGUA DE CONDENSADO | G130 | FLUJO DEL COMBUSTIBLE | OPC | 0 | 0 | 0 |
| CALENTADOR DE AIRE | P113 | PRESION DEL VAPOR AL GWV | OPC | 0 | 0 | 0 |
| CALENTADOR N° 1 | T113 | TEMPERATURA DEL VAPOR AL GWV | OPC | 0 | 0 | 0 |
| CALENTADOR N° 2 | T128 | TEMPERATURA DEL DREN DEL GWV | PARAMETRIZADO | 0 | 0 | 0 |
| CALENTADOR N° 3 | G_Api | GRADOS API | PROPUESTO | 0 | 0 | 0 |
| CALENTADOR N° 4 | T130 | TEMPERATURA DEL COMBUSTIBLE A LA ENTRADA | OPC | 0 | 0 | 0 |
| CALENTADOR N° 6 | T131 | TEMPERATURA DEL COMBUSTIBLE A LA SALIDA | OPC | 0 | 0 | 0 |
| CALENTADOR N° 7 | | | | | | |
| CONDENSADOR DE EYECTORES | | | | | | |
| CONDENSADOR DE SELLOS | | | | | | |
| CONDENSADOR PRINCIPAL | | | | | | |
| DEAREADOR | | | | | | |
| ENFRIADOR DE DRENES | | | | | | |
| EVAPORADORA | | | | | | |
| GENERADOR DE VAPOR | | | | | | |
| GENERADOR DE VAPOR-VAPOR | | | | | | |
| GENERADOR ELÉCTRICO | | | | | | |
| TURBINA DE ALTA PRESIÓN | | | | | | |
| TURBINA DE BAJA PRESIÓN | | | | | | |
| TURBINA DE PRESIÓN | | | | | | |

Imagen 5.2.- Señales de entrada del generador de vapor-vapor de la unidad 2 de Manzanillo II

- Las variables de entrada pueden ser originadas a partir de:
 - **OPC:** es un sistema encargado de obtener las señales directamente de los equipos para después proporcionarlas al sistema LAPEM-TERMO®.
 - **Calculado:** depende del valor de las señales de otros dispositivos.
 - **Parametrizado:** se obtiene a partir de ecuaciones de n (2 a 7) grados.
 - **Propuesto:** a partir de un valor fijo externo, como por ejemplo el valor del combustible.
 - **Diseño:** se obtiene interpolando diferentes valores de diseño.

Un punto interesante al momento de definir las variables de entrada fue que se tuvo que tomar en cuenta las particularidades de los equipos que se utilizan en la planta.

Así, para calcular las temperaturas, para los equipos con dos lados, A y B, como ocurrió con varios calentadores se tuvo que sacar el promedio, mientras que para los equipos con dos valores de flujos se sumaron los valores de cada lado.

1.2. Señales de salida.

Las unidades de medida para las señales de salida no necesariamente tienen que coincidir con las unidades de las señales de entrada, sino que dependerá de las preferencias de cada planta y sus usuarios.

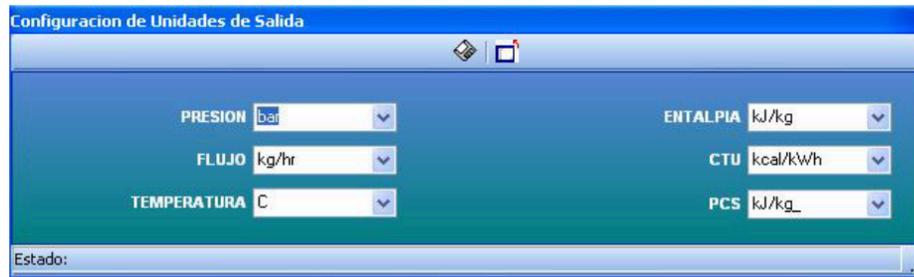


Imagen 5.3- Configuración de señales de salida en el sistema LAPEM-TERMO®

1.3. Coeficientes de desviaciones del ciclo.

Se configuraron los coeficientes de las ecuaciones para el cálculo de las desviaciones del ciclo.

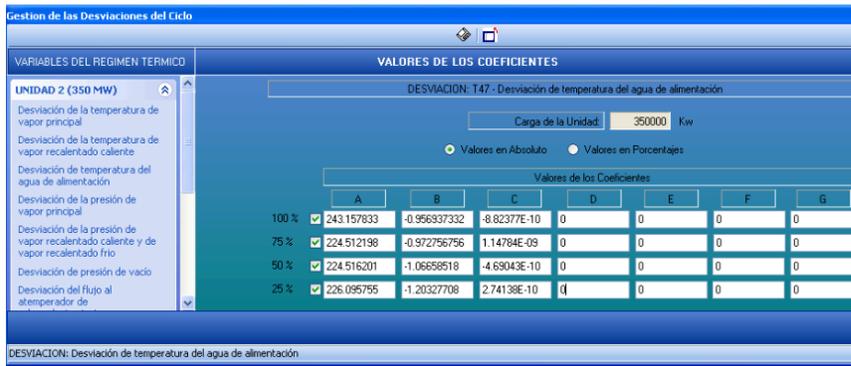


Imagen 5.4- Configuración de los coeficientes de las ecuaciones en el sistema LAPEM-TERMO®

Una vez configurados las variables y coeficientes, se procedió a realizar las configuraciones para la puesta en operación en modo automático. Esta configuración se puede realizar eligiendo la opción Configurar OPC dentro del menú de configuración.

Se definieron los siguientes puntos:

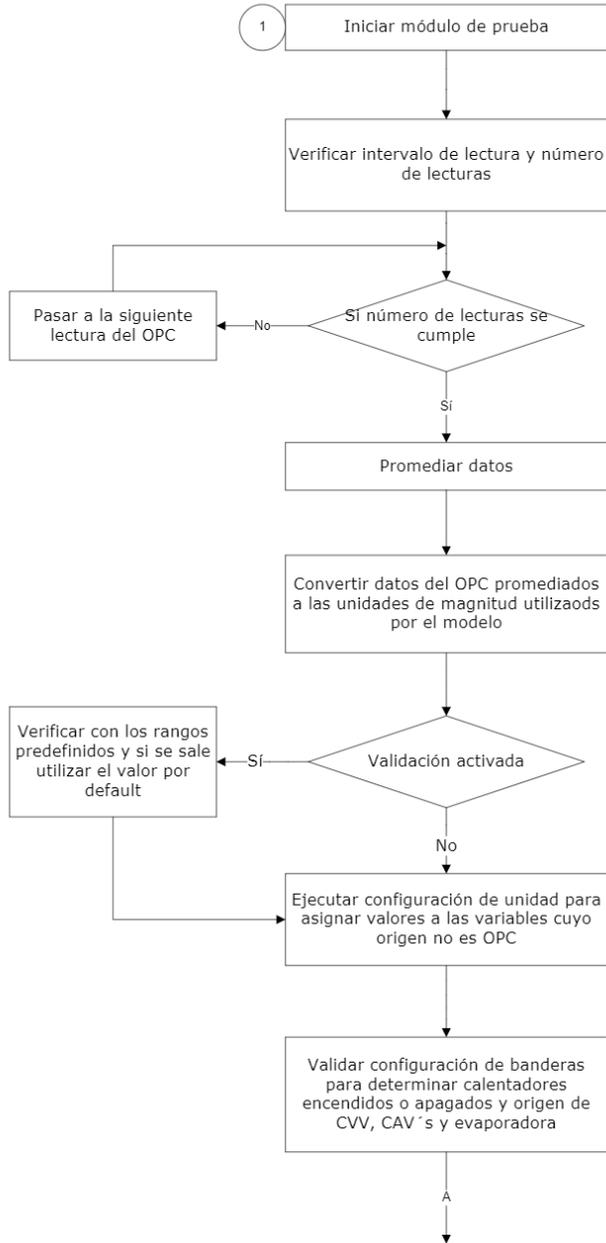
- Ubicación del archivo OPC.
- Intervalo de prueba (horas, minutos).
- Número de lecturas para promediar los datos de OPC.
- Decidir si se utiliza la validación de valores del OPC.



Imagen 5.5- Configuración del modo automático en el sistema LAPEM-TERMO®

Cabe aclarar que los dos puntos descritos hasta el momento, puede variar según la planta en la que se implemente el software, ya que tendrán diferentes equipos, tipo de señales y preferencias particulares propias de cada planta.

El siguiente proceso es iniciar el módulo de prueba. En el siguiente diagrama se muestra el funcionamiento de este módulo.



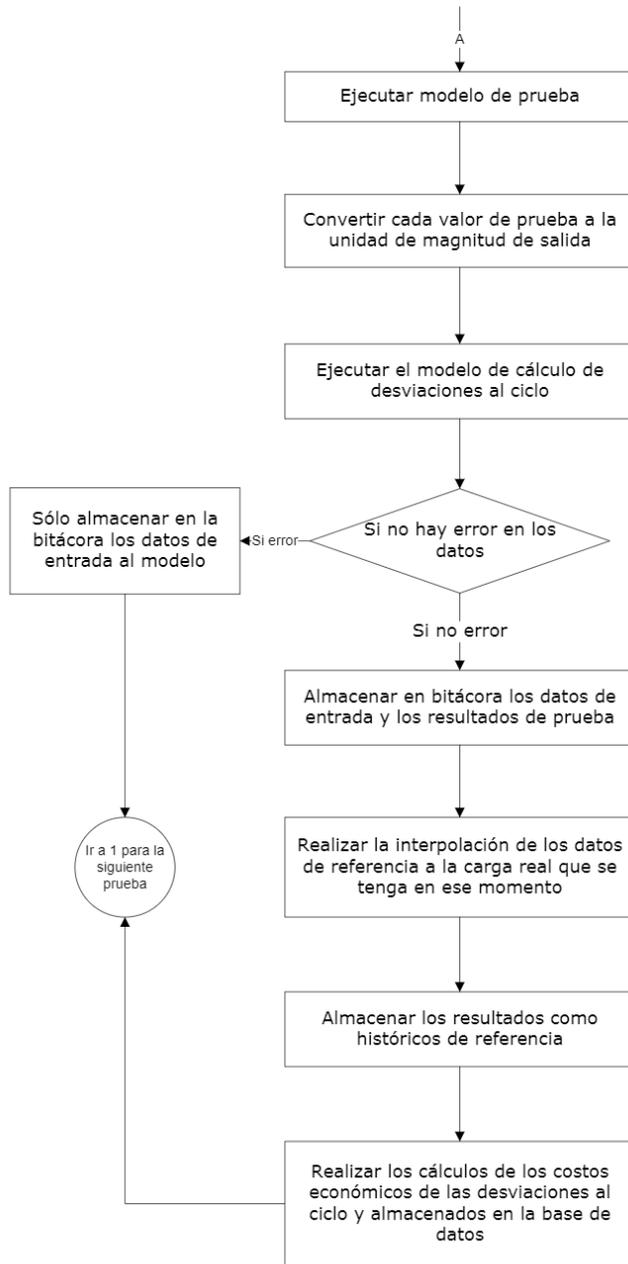


Diagrama 5.6- Modelo de prueba para el sistema LAPEM-TERMO®

CONCLUSIONES

Después de una larga fase de trabajo en la elaboración del Sistema de Gestión de Base de Datos para el Sistema de Simulación y Monitoreo del Rendimiento de Centrales Termoeléctricas "LAPEM-TERMO®", pasando desde la etapa de análisis, diseño, desarrollo, implementación y pruebas del sistema, toca ahora establecer las conclusiones para la presente tesis.

Estas conclusiones se enfocaran en dos puntos distintos, la primera parte se enfocará en el aspecto social, en donde se abordarán los beneficios que deja nuestra tesis para nuestra sociedad, mientras la segunda se enfocará en el aspecto profesional, que describirá las aportaciones que deja nuestra tesis en la rama de la Ingeniería en Computación y las conclusiones en general de nuestro trabajo.

La presente tesis expone una parte del diseño y desarrollo del sistema LAPEM-TERMO®, enfocada principalmente a su modulo de base de datos. Para mencionar los aspectos importantes en los que nuestra tesis aporta algo a nuestra sociedad, tomaremos en cuenta el sistema en su totalidad. El sistema LAPEM-TERMO® pretende ser una herramienta importante dentro de la toma de decisiones en las centrales termoeléctricas de México; su principal objetivo es el de realizar las pruebas de régimen térmico dentro de las plantas.

Por ello el sistema LAPEM-TERMO® aporta como herramienta un apoyo muy importante para los directivos de las centrales termoeléctricas al momento de tomar decisiones dentro de la planta, ya que brinda una gran parte de los datos necesarios para mantener informado al personal acerca de lo que acontece en ella, datos que se obtienen dentro de los monitoreos que realiza el sistema en cada unidad y en cada equipo de la central.

De la misma forma, permite efectuar un control sobre la central, ya que el sistema tiene la posibilidad de realizar cambios en alguna unidad o en algún equipo en específico (ya sea activando o desactivando algún equipo, disminuyendo o aumentando el ritmo de trabajo de alguna unidad, modificando el valor de alguna variable como puede ser la temperatura en una caldera, etc.), tareas que pueden efectuarse debido a una falla o simplemente por una estrategia de trabajo.

Además el sistema brinda la posibilidad de realizar simulaciones que auxilien a los directivos al momento de querer saber los resultados de posibles decisiones futuras, sobre todo al modificar el valor de algunas variables (ya sean de presión, temperatura, entalpía, etc.) que resulta difícil realizar experimentalmente sin tener consecuencias graves o peligrosas.

Por todo esto el sistema LAPEM-TERMO® adquiere una gran importancia dentro de las centrales termoeléctricas del país, pues brinda una manera más práctica de administrarlas y con ello una manera de volverlas más eficientes y productivas, ya que las decisiones que toman los directivos se pueden efectuar en tiempo real, y por ende, tener resultados casi inmediatamente; lo cual beneficia a un alto porcentaje de la sociedad mexicana, ya que el correcto funcionamiento de estas plantas se traduce en un buen servicio y por ello, permite que millones de mexicanos se desempeñen productivamente dentro de su trabajo y hasta en sus mismos hogares, pues la energía eléctrica que se genera en éstas centrales termoeléctricas representa alrededor de un 75% de la que se produce en el país.

Pasando ahora al segundo punto, nos referiremos a las conclusiones de nuestra tesis y de lo que ésta aporta a nuestra rama profesional.

El primer punto a destacar dentro de este entorno, es el resaltar la importancia de manejar una base de datos relacional, ya que es una manera muy sencilla de organizar y almacenar información de manera estructurada mediante tablas, lo cual hace más sencillas las consultas que desee realizar un usuario. Permiten una garantía de acceso total a los datos, pues podemos acceder a ellos siempre mediante una combinación muy simple conformada por el nombre de tabla, llave primaria y nombre de la columna. Otra ventaja muy importante que se obtiene al utilizar una base de datos relacional es que nos permite trabajar con valores nulos en los registros (excepto en los campos que sean parte de la llave primaria de la tabla) y además nos brindan herramientas que garantizan evitar la duplicidad de registros, especialmente en las llaves primarias y foráneas.

A su vez, garantizan la integridad referencial, pues al eliminar, insertar o actualizar un registro, este cambio se realiza en todas las tablas que lo utilicen, ya sea completo o solo en uno o varios campos de él. También favorece la normalización por su estructura y por ser más fácil de comprender, ya que podemos ver directamente cada uno de los registros o la tabla en su totalidad, lo cual también facilita la aplicación de cambios en la estructura y organización de la base de datos.

Otro punto importante es que nos proporcionan una gran facilidad de poder establecer búsquedas eficientes en la información almacenada, ya que solo basta indicar el nombre de la tabla en donde se desea buscar, su llave primaria del registro y el campo o los campos de interés.

Por todo lo anterior, las bases de datos relacionales, son una muy buena alternativa para los sistemas de información que se desarrollan hoy en día, lo cual hace que sean la estructura de diseño de base de datos más utilizada en el mundo actual.

Por otra parte, también queremos destacar las ventajas de la programación orientada a objetos, pues contribuye a poder realizar una mejora continua en cuanto al diseño y programación de nuestras aplicaciones, aspecto que se debe directamente a las mejoras que presenta en comparación a otras formas de programación (por ejemplo la programación estructurada) como lo son la inserción de nuevos módulos y clases, lo cual lo podemos hacer debido a características que presenta, como lo son la modularidad, herencia, polimorfismo, encapsulamiento y la jerarquía.¹

El sistema LAPEM-TERMO® en sí, resulta ser bastante escalable y adaptable a cambios debido a su programación estructurada en módulos bien definidos desde la etapa de diseño. La programación orientada a objetos y una base de datos relacional son dos características con las que cuenta el sistema, lo que lo hace un sistema sólido y apegado a la tendencia de la programación de hoy en día.

¹ **Modularidad:** propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

Herencia: En la programación orientada a objetos, la herencia es un mecanismo que permite derivar una clase de otra, de manera que extienda su funcionalidad

Polimorfismo: En programación orientada a objetos se denomina polimorfismo a la capacidad que tienen los objetos de una clase de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación.

Encapsulamiento: en programación orientada a objetos, se denomina encapsulamiento al ocultamiento del estado, es decir, de los datos miembro, de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto.

Jerarquía: consiste en la clasificación y organización de las abstracciones según su naturaleza. El más claro ejemplo de jerarquía en la programación orientada a objetos es la herencia.

También brindan la posibilidad de efectuar cambios y actualizaciones en el sistema en un futuro (debido a los puntos que ya se comentaron anteriormente), sobre todo porque este sistema puede ser la base para otro que englobe todas las plantas generadoras de electricidad en México, sin importar la forma de generación, y así poder tener sistematizada toda la producción, reduciendo considerablemente las fallas, atendiendo inmediatamente los desperfectos y aumentando la productividad y eficiencia de las centrales.

En cuanto a los resultados del sistema LAPEM-TERMO®, como se comento en el capítulo 5 de la presente tesis, éste se encuentra en fase de pruebas en la Central Termoeléctrica de Manzanillo, dentro de las cuales a obtenido buenos resultados dentro de las primeras pruebas; y de continuar así y una vez siendo aprobado por la Comisión Federal de Electricidad, se instalará en todas y cada una de las 25 centrales que conforman este proyecto, y con ello lograr tener sistematizada una gran parte de las plantas termoeléctricas de México.

Finalmente, dentro de las habilidades que desarrollamos al realizar ésta tesis, se encuentran la habilidad de realizar un análisis de requerimientos, habilidad para realizar un análisis y un diseño para un sistema de información, habilidad para diseñar y desarrollar una base de datos relacional, experiencia programando en un lenguaje de alto nivel (orientado a objetos) y experiencia realizando una implementación física y pruebas a un sistema informático. Si observamos atentamente los puntos anteriores podemos concluir que realizando nuestra tesis adquirimos experiencia trabajando dentro del ciclo de vida de los sistemas de información, hecho que nos prepara aún más para poder enfrentarnos a la vida profesional.

ANEXO # 1

DICCIONARIO DE DATOS

TABLA "Plantas"

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|---------------|---------------------|---|
| ID_Planta PK | NUMBER | Identificador asignado a cada planta para diferenciar una con respecto a las demás. |
| Nombre_Planta | CHAR | Nombre oficial de la planta en cuestión. |
| Domicilio | CHAR | Dirección física de la planta dentro de la República Mexicana. |
| Telefono | CHAR | Teléfono de la oficina central de la planta. |
| Entidad | CHAR | Estado de la República Mexicana en donde está ubicada la planta. |
| Super_Int | CHAR | Nombre del encargado de la planta. |
| E-mail | CHAR | Dirección de correo electrónico de la oficina central de la planta. |
| Descripcion | CHAR | Breve descripción de la planta y de sus características más importantes. |
| Imagen | OLE OBJECT | Imagen de la planta para la interfaz gráfica del sistema. |
| X | NUMBER | Coordenada (abscisa) para imagen de Planta. |
| Y | NUMBER | Coordenada (ordenada) para imagen de Planta. |

TABLA “Unidades”

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|----------------|---------------------|--|
| ID_Unidad PK | NUMBER | Identificador asignado a cada unidad para diferenciar una con respecto a las demás. |
| ID_Planta FK | NUMBER | Identificador asignado a cada planta para diferenciar una con respecto a las demás. Sirve para saber a qué planta pertenece determinada unidad. |
| Nombre_Unidad | CHAR | Nombre oficial de la unidad en cuestión. |
| Descripcion | CHAR | Breve descripción de la unidad y de sus características más importantes. |
| Imagen | OLE OBJECT | Imagen de la unidad para la interfaz gráfica del sistema. |
| N_TurbinasI | NUMBER | Número de turbinas intermedias con las que cuenta la unidad. |
| Potencia | NUMBER | Magnitud de potencia que es capaz de producir la unidad. |
| U_Potencia | CHAR | Unidad de medida de la magnitud de potencia del campo anterior. |
| N_Extracciones | NUMBER | Número de extracciones realizado en la unidad. |
| T_Coeficientes | NUMBER | Tenemos dos opciones → 1: Si es en Absolutas; 0: Si es en Porcentajes para las Desviaciones. |
| OPC | CHAR | Ubicación y nombre del archivo proveniente del OPC. |
| Intervalo | NUMBER | Intervalo de Simulación. |
| T_Intervalo | NUMBER | Tenemos dos opciones → 1: Minutos; 2: Horas |
| N_Lecturas | NUMBER | Numero de lecturas realizadas. |
| Validacion | BOOLEAN | Casilla para verificar si los datos de la unidad se van a validar o no con los valores mínimos y máximos y considerar el valor default de la unidad. |

TABLA “Equipos”

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|---------------|---------------------|---|
| ID_Equipo PK | NUMBER | Identificador asignado a cada equipo para diferenciar uno con respecto a las demás. |
| ID_Unidad FK | NUMBER | Identificador asignado a cada unidad para diferenciar una con respecto a las demás. Sirve para saber a que unidad pertenece determinado equipo. |
| Nombre_Equipo | CHAR | Nombre oficial del equipo. |
| Descripcion | CHAR | Breve descripción del equipo y de sus características más importantes. |
| ID_Imagen FK | NUMBER | Identificador asignado a cada imagen para diferenciar una con respecto a las demás. |
| X1 | NUMBER | Coordenada (abscisa1) para imagen de Equipo. |
| Y1 | NUMBER | Coordenada (ordenada1) para imagen de Equipo. |
| X2 | NUMBER | Coordenada (abscisa2) para imagen de Equipo. |
| Y2 | NUMBER | Coordenada (ordenada2) para imagen de Equipo. |
| Habilitado | NUMBER | Valor que nos indica si el equipo está habilitado o no. |
| Selec | NUMBER | Tenemos dos opciones → 1: Puede el usuario apagarlo; 0: el usuario no puede hacer nada sobre el |

TABLA “Imagen_Equipos”

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|--------------|---------------------|---|
| ID_Imagen PK | NUMBER | Identificador asignado a cada imagen para diferenciar una con respecto a las demás. |
| Imagen | OLE OBJECT | Imagen del equipo para ser utilizada en la interfaz gráfica del sistema. |

TABLA "Usuarios"

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|----------------|---------------------|---|
| Nivel PK | NUMBER | Identificador asignado a cada usuario para diferenciar uno con respecto a los demás. |
| Usuario | CHAR | Describe que tipo de usuario es, ya sea LAPEM, DIRECTIVO ó ING. PRUEBAS. |
| Clave | CHAR | Contraseña cifrada del usuario. |
| Nombre_Usuario | CHAR | Nombre del usuario. |
| Descripcion | CHAR | Breve descripción del usuario y sus privilegios en el sistema. |
| Tipo | NUMBER | Tenemos 3 opciones → 1: Configuracion_Lapem, 2: Administrador_Planta ó Directivo, 3: Operador ó Ing. De Pruebas. |

TABLA "Unidad_M"

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|--------------|---------------------|--|
| ID_UniMed PK | NUMBER | Identificador asignado a cada unidad de medida para diferenciar una con respecto a las demás. |
| Tipo | NUMBER | Hace referencia a un indicador asignado a cada tipo de magnitud, por ejemplo: el numero 1 para Presión, el numero 2 para Flujo, el numero 3 para Temperatura, etc. |
| Magnitud | CHAR | Indica el nombre de la magnitud a la que hace referencia la unidad de medida. |
| Nom_Unidad | CHAR | Nombre de la unidad. |

TABLA "Unidad_Salida"

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|--------------|---------------------|--|
| ID_Unidad FK | NUMBER | Identificador asignado a cada unidad para diferenciar una con respecto a las demás. Sirve para saber a qué unidad pertenece la unidad de salida en cuestión. |
| ID_UniMed FK | NUMBER | Identificador asignado a cada unidad de medida para diferenciar una con respecto a las demás. Sirve para saber con qué unidad de medida trabaja la unidad de salida. |
| Magnitud | CHAR | Indica el nombre de la magnitud a la que hace referencia la unidad de medida, ya sea Presión, Flujo, Temperatura, Potencia, etc. |

TABLA "Variables_Prueba"

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|-----------------|---------------------|--|
| ID_VariableP PK | NUMBER | Identificador asignado a cada variable para diferenciar una con respecto a las demás. |
| ID_Equipo FK | NUMBER | Identificador asignado a cada equipo para diferenciar uno con respecto a las demás. Sirve para saber a qué equipo pertenece la variable de prueba en cuestión. |
| Codigo | CHAR | Código de la variable. |
| Descripcion | CHAR | Descripción de la variable. |
| Clave_Unidad | NUMBER | Clave de la unidad de medida. |
| ID_UniMed FK | NUMBER | Identificador asignado a cada unidad de medida para diferenciar una con respecto a las demás. |
| Opciones OPC | NUMBER | Opciones para la configuración del OPC. |

| | | |
|-------------|--------|--|
| Default OPC | NUMBER | Valor por default del OPC. |
| Valor Fijo | NUMBER | Valores leídos del OPC y Propuestos |
| Valor | NUMBER | Valores calculados y convertidos a la unidad de salida. |
| ValorMin | NUMBER | Valor mínimo. |
| ValorMax | NUMBER | Valor máximo. |
| ValorDef | NUMBER | Valor por default por si el valor no está en el rango. |
| ID_UniMedT | NUMBER | Identificador para la unidad de medida alternativa. |
| X | NUMBER | Coordenada (abscisa) para imagen. |
| Y | NUMBER | Coordenada (ordenada) para imagen. |
| X1 | NUMBER | Coordenadas del informe. |
| Y1 | NUMBER | Coordenadas del informe. |
| TAG | NUMBER | Para ciertas presiones: 1:P Absoluta 2:P Barométrica; 0 no considerar. |
| Tipo | CHAR | 1: Entrada 2: Calculado 3: Nueva Variable. |

TABLA "Variables_Referencia"

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|-----------------|---------------------|--|
| ID_Equipo PK | NUMBER | Identificador asignado a cada equipo para diferenciar uno con respecto a las demás. Sirve para saber a qué equipo pertenece la variable de prueba en cuestión. |
| ID_VariableR FK | NUMBER | Identificador asignado a cada variable para diferenciar una con respecto a las demás. |
| Codigo | CHAR | Código de la variable. |

| | | |
|--------------|--------|---|
| Descripcion | CHAR | Descripción de la variable. |
| Valor_100 | CHAR | Valor de la variable al 100% |
| Valor_75 | CHAR | Valor de la variable al 75% |
| Valor_50 | CHAR | Valor de la variable al 50% |
| Valor_25 | CHAR | Valor de la variable al 25% |
| Clave_Unidad | NUMBER | Clave de la unidad de medida. |
| ID_UniMed FK | NUMBER | Identificador asignado a cada unidad de medida para diferenciar una con respecto a las demás. |
| Valor | NUMBER | Valor actual de la variable. |
| ID_UniMedT | NUMBER | Identificador para la unidad de medida alternativa. |
| X | NUMBER | Coordenada (abscisa) para imagen. |
| Y | NUMBER | Coordenada (ordenada) para imagen. |
| X1 | NUMBER | Coordenadas para el grafico de informe |
| Y1 | NUMBER | Coordenadas para el grafico de informe |
| Tipo | NUMBER | 1:Variable del modelo; 2: Variable Agregada por el usuario |

TABLA "Variables_Desviaciones"

| Campo | Tipo de Dato | Descripción |
|------------------|---------------------|--|
| ID_Desviacion PK | AUTOINCREMENT | Identificador para la desviación. |
| ID_Unidad FK | NUMBER | Identificador asignado a cada unidad para diferenciar una con respecto a las demás. Sirve para saber a qué unidad pertenece la variable de desviación en cuestión. |
| ID_VariableD | CHAR | Identificador de la variable de desviación. |
| Nombre | CHAR | Nombre de la variable de desviación. |

| | | |
|-------|--------|----------------------------|
| EC100 | NUMBER | 1: Si tiene; 0:No tiene |
| EC75 | NUMBER | 1: Si tiene; 0:No tiene |
| EC50 | NUMBER | 1: Si tiene; 0:No tiene |
| EC25 | NUMBER | 1: Si tiene; 0:No tiene |
| C100A | NUMBER | Parámetro al 100% valor A. |
| C100B | NUMBER | Parámetro al 100% valor B. |
| C100C | NUMBER | Parámetro al 100% valor C. |
| C100D | NUMBER | Parámetro al 100% valor D. |
| C100E | NUMBER | Parámetro al 100% valor E. |
| C100F | NUMBER | Parámetro al 100% valor F. |
| C100G | NUMBER | Parámetro al 100% valor G. |
| C75A | NUMBER | Parámetro al 75% valor A. |
| C75B | NUMBER | Parámetro al 75% valor B. |
| C75C | NUMBER | Parámetro al 75% valor C. |
| C75D | NUMBER | Parámetro al 75% valor D. |
| C75E | NUMBER | Parámetro al 75% valor E. |
| C75F | NUMBER | Parámetro al 75% valor F. |
| C75G | NUMBER | Parámetro al 75% valor G. |
| C50A | NUMBER | Parámetro al 50% valor A. |
| C50B | NUMBER | Parámetro al 50% valor B. |
| C50C | NUMBER | Parámetro al 50% valor C. |
| C50D | NUMBER | Parámetro al 50% valor D. |
| C50E | NUMBER | Parámetro al 50% valor E. |
| C50F | NUMBER | Parámetro al 50% valor F. |
| C50G | NUMBER | Parámetro al 50% valor G. |

| | | |
|------|--------|---------------------------|
| C25A | NUMBER | Parámetro al 25% valor A. |
| C25B | NUMBER | Parámetro al 25% valor B. |
| C25C | NUMBER | Parámetro al 25% valor C. |
| C25D | NUMBER | Parámetro al 25% valor D. |
| C25E | NUMBER | Parámetro al 25% valor E. |
| C25F | NUMBER | Parámetro al 25% valor F. |
| C25G | NUMBER | Parámetro al 25% valor G. |

TABLA "Resultados_Prueba_H"

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|-----------------|---------------------|---|
| ID_ResPH PK | AUTOINCREMENT | Identificador de resultado de prueba. |
| Fecha | CHAR | Fecha de la prueba. |
| Hora | CHAR | Hora de la prueba. |
| ID_VariableP FK | NUMBER | Identificador de la variable de prueba. |
| Codigo | CHAR | Codigo de la Variable |
| ID_UniMed FK | NUMBER | Identificador asignado a cada unidad de medida para diferenciar una con respecto a las demás. |
| Valor | NUMBER | Valor Calculado o Leido de la Variable |
| Tipo | CHAR | E:Variable Entrada; S: Variable de Salida |

TABLA "Resultados_Referencia_H"

| <i>Campo</i> | <i>Tipo de Dato</i> | <i>Descripción</i> |
|-----------------|---------------------|---|
| ID_ResRH PK | AUTOINCREMENT | Identificador del resultado de referencia. |
| Fecha | CHAR | Fecha del resultado de referencia. |
| Hora | CHAR | Hora del resultado de referencia. |
| ID_VariableR FK | NUMBER | Identificador de la variable de referencia. |
| Codigo | CHAR | Codigo de la Variable |
| ID_UniMed FK | NUMBER | Identificador asignado a cada unidad de medida para diferenciar una con respecto a las demás. |
| Valor | NUMBER | Valor Calculado o Leido de la Variable |

ANEXO # 2

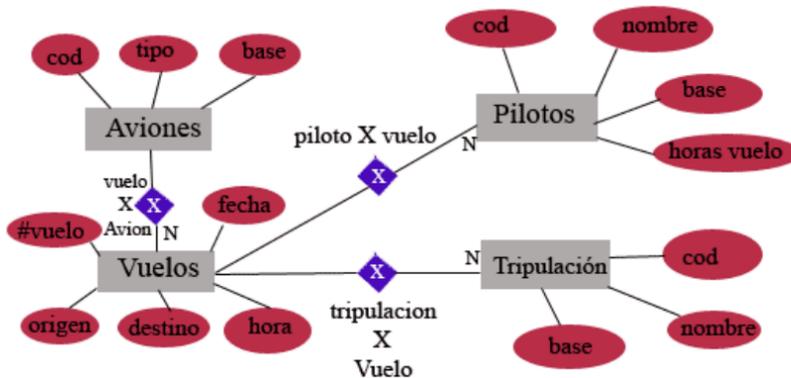
EJEMPLOS DE MODELOS DE DATOS DE ALTO NIVEL

MODELO ENTIDAD-RELACION ¹

Ejemplo:

Se desea almacenar la información de una compañía aérea en una B.D relacional. La compañía aérea tiene tres recursos principales: aviones, pilotos, tripulación. De cada avión se desea conocer su código, nombre y horas de vuelo. De los miembros de la tripulación solo se tendrá el código y el nombre. Pilotos y tripulación tienen una base a la que regresan después de cada jornada, un vuelo va desde un origen a un destino a una hora concreta y tiene numero de vuelo, de cada vuelo que se va a realizar durante los próximos 3 meses, así como de los vuelos que se han realizado se desea saber el avión en el que se va a hacer o en el que se ha hecho, el piloto y la tripulación.

Cada avión tiene un código, es de un tipo (boeing, airbus, entre otros). Y tiene una base donde es sometido a mantenimiento.



MODELO ORIENTADO A OBJETOS ²

Ejemplo:

El modelo SOM, o modelado de Objetos permite ilustrar a la base de datos en términos de Objetos de negocio, y no directamente en tablas. Bajo la metodología SOM, una entidad es cualquier cosa, concreta o abstracta, con un infinito número de propiedades o características que la describen. Por ejemplo, un auto tiene modelo, número de neumáticos, tipo de bencina que utiliza, capacidad del tanque de aceite, potencia del motor, peso molecular, capacidad de reflejo de rayos UV según la pintura del fabricante, y así, podemos tardarnos un buen tiempo describiendo las características del auto. Sin embargo, dependiendo del cliente o usuario al que estemos sirviendo en cuestiones de informática, cada empresa, persona, departamento, institución, compañía, tendrá necesidades distintas y únicas en el conjunto de propiedades o características que desean reunir para una entidad en particular. Al “conjunto nombrado de propiedades que suficientemente describen una entidad en el ambiente de trabajo del usuario”, ese es un Objeto Semántico. Siguiendo con el ejemplo anterior, a un vendedor de autos usados le interesará conocer datos, como:

- Número de Serie
- Marca
- Sub-marca
- Modelo
- Kilometraje
- Estado del auto
- Precio mínimo
- Precio máximo
- Precio final de venta
- Fecha inicial de venta
- Fecha de venta

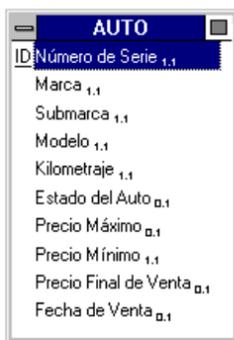
A un técnico de agencia, al que le han llevado ese auto, le podrá interesar:

- Cliente
- Marca
- Sub-marca
- Modelo
- Kilometraje
- Placas actuales del vehículo
- Número de serie
- Reporte de fallas

- Costo en refacciones
- Costo de mano de obra
- Fecha de entrada al taller
- Fecha de salida
- Control de Calidad del Servicio

Como vemos, dependiendo de la realidad en donde operará el sistema que estamos analizando, puede haber diferentes “vistas” o perspectivas de la realidad del mismo Objeto.

Para el vendedor, el Objeto AUTO se modelaría del siguiente modo:

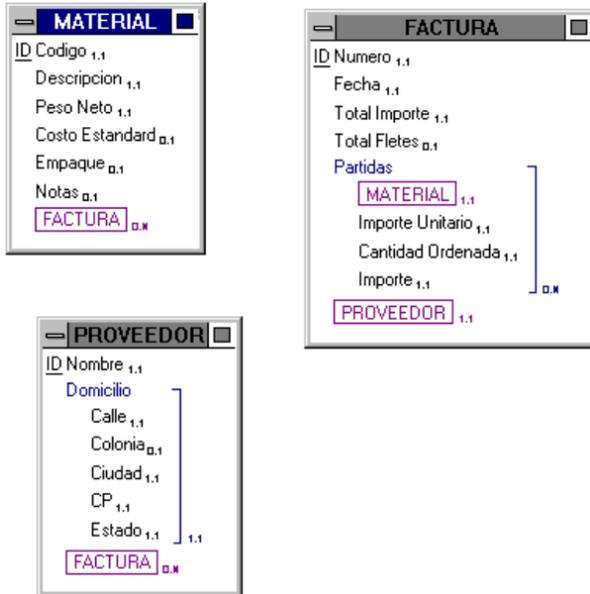


En este diagrama podemos ver varios elementos:

1. La propiedad “Número de Serie” está prefijada con “ID” que significa que esta propiedad identifica a un caso particular del Objeto; en este caso, el “Número de Serie” es la propiedad que identifica a un AUTO en particular.
 - 1.1. Nótese el término formal “**propiedad**”, que se usa bajo SOM para identificar una característica del Objeto que nos interesa considerar.
2. En cada propiedad hay un subíndice tipo “0.1” o “1.1”; esta es la llamada **cardinalidad** que indica el mínimo y máximo número de valores que la propiedad puede tomar.
 - 2.1. Por ejemplo, nótese que “Precio Máximo” tiene una cardinalidad de “0.1”; esto significa la propiedad puede no tener un valor (0), y máximo 1.
 - 2.2. “Número de Serie” tiene una cardinalidad de “1.1”, lo cual indica que debe haber un valor almacenado, pero no más de uno; se trata entonces de una propiedad en que es obligatorio capturar un valor.
 - 2.3. Si tuviéramos una propiedad con cardinalidad “0.N” o “1.N” significa

que pueden almacenarse N valores para esa propiedad. Más adelante veremos un ejemplo con un caso como este.

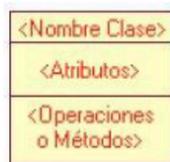
- 2.4. Pueden darse cardinalidades como “1.3, 2.5, 0.30”, dependiendo de las reglas del negocio en torno al Objeto.
3. Podemos observar que las propiedades se nombran de manera clara y explícita en lenguaje natural, describiendo de manera concisa lo que la propiedad contiene y significa.
 - 3.1. No es como en campos o variables, que en ocasiones abreviamos o hacemos algo corto su nombre por restricciones del lenguaje o DBMS, o simplemente por costumbre o estándares.



MODELO LÓGICO ³

Un modelo lógico es una vista estática de los objetos y las clases que cubren el espacio de análisis y diseño. Un ejemplo de modelo lógico es el modelo de clases, que sirve para visualizar las relaciones entre las clases que involucran el sistema.

En UML, una clase es representada por un rectángulo que posee tres divisiones:



- En donde:
 - **Superior:** Contiene el nombre de la Clase
 - **Intermedio:** Contiene los atributos (o variables de instancia) que caracterizan a la Clase.
 - **Inferior:** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno.

Ejemplo:

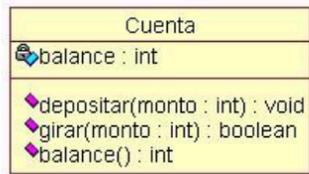
Una Cuenta Corriente que posee como característica:

- Balance

Puede realizar las operaciones de:

- Depositar
- Girar
- y Balance

El diseño asociado es:



MODELO INFOLÓGICO ⁴

Las metodologías del desarrollo sobre el acercamiento infológico y el modelar conceptual (los datos que modelan) tienen su origen en ambientes estadísticos.

A comienzos de los años 70, Langefors y algunos de sus estudiantes, desviaron su atención al desarrollo de una teoría infológica, una teoría de conceptos y contenidos informativos. Sundgren (1973) desarrolló el framework OPR(t), también llamados modelos infológicos, o modelos conceptuales, como pasaron a llamarse más tarde en la literatura internacional. El framework OPR(t) está basado en rigurosos análisis de cuatro conceptos fundamentales: objetos, propiedades, relaciones (entre objetos) y tiempo. Langefors los llamó mensajes elementales, bloques de información.

Podemos considerar al modelo E-R o al modelo E-A-R presentado por Chen en 1976 como un subconjunto de este modelo infológico cuya primera mejora había sido publicada por Sundgren tres años antes (1973).

La teoría infológica, a principios de los 70, se utilizó como proyecto para llevar a cabo estadísticas económicas. Como proyecto paralelo también se desarrolló un catálogo de variables, y que era usado para el proyecto estadístico ya mencionado.

Pero, aún otro proyecto más importante iba a ser desarrollado, como fruto de la inspiración en las ideas infológicas. Se trataba de una base de datos que traía consigo tres grandes partes bien diferenciadas:

- Microbase de datos: datos a nivel atómico.
- Macrobases de datos: Contenia datos estadísticos agregados multidimensionalmente.
- Metabase de datos: Contenia descripciones técnicas y orientadas a contenidos de las microbases y de las macrobases.

Quizá, ya podíamos decir que este iba a ser uno de los primeros “datawarehouse” del mundo.

¹ Tomado de <http://basdatos.tripod.com/ejercicios.html>

² Tomado de http://www.postgresql.cl/articulos/gmo_najar.htm

³ Tomado de <http://www.dcc.uchile.cl/~psalinas/uml/modelo.html>

⁴ Tomado de http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900_FLorenzo.pdf

BIBLIOGRAFIA

1. DATE, C. J.
“An Introduction to Database Systems”, 8a Edición
Ed. Addison Wesley, U.S.A. 2003
2. DE MIGUEL Martínez, Adoración et al.
“Diseño de Bases de Datos Relacionales”
Ed. Alfaomega, México 2001
3. JOHNSON, James
“Bases de Datos, Modelos, Lenguajes, Diseño”
Ed. Oxford, México 2000
4. VIESCAS, John L.
“Microsoft Office Access 2003 Inside Out”
Ed. Microsoft Press
5. SHELDON, Robert
“SQL: A Beginner’s Guide”
Ed. McGraw Hill
6. PONS Capote, Olga et al.

“Introducción a las Bases de Datos. El Modelo relacional”

Ed. Paraninfo

7. PONS Capote, Olga et al.

“Introducción a los sistemas de Bases de Datos”

Ed. Paraninfo

8. CABRERA Sánchez, Gregorio

“Sistemas Gestores de Bases de Datos”

Ed. Paraninfo

REFERENCIAS DE INTERNET

9. <http://www.cfe.gob.mx>

Lunes 11 de agosto de 2008, 10:05 hrs.

10. <http://www.microsoft.com/spain/sql/default.mspx>

Martes 12 de agosto de 2008, 12:35 hrs.

11. <http://www.formaselect.com/curso/experto-en-sql-server-2000/Introduccion-a-SQL-Server%202000.pdf>

Martes 12 de agosto de 2008, 13:20 hrs.

12. http://es.wikipedia.org/wiki/SQL_server

Martes 12 de agosto de 2008, 15:30 hrs.

13. <http://www3.uji.es/~mmarques/f47/apun/node45.html>

Jueves 14 de agosto de 2008, 13:30 hrs.

14. <http://www3.uji.es/~mmarques/f47/apun/node83.html>

Jueves 14 de agosto de 2008, 14:50 hrs.

15. http://es.wikipedia.org/wiki/Diagrama_entidad-relaci%C3%B3n

Jueves 14 de agosto de 2008, 11:20 hrs.

16. http://es.wikipedia.org/wiki/Peter_Chen

Jueves 14 de agosto de 2008, 14:25 hrs.

17. <http://www.cfe.gob.mx/es/LaEmpresa/generacionelectricidad/lisctralesgeneradoras/termoelectricas.htm?Combo=termoelectricas>

Domingo 17 de agosto de 2008, 14:30 hrs.

18. http://es.wikipedia.org/wiki/Central_termoel%C3%A9ctrica

Domingo 17 de agosto de 2008, 16:50 hrs.

19. <http://www.monografias.com/trabajos13/genytran/genytran.shtml>

Domingo 17 de agosto de 2008, 18:40 hrs.

20. http://es.wikipedia.org/wiki/Energ%C3%ADa_geot%C3%A9rmica

Lunes 18 de agosto de 2008, 13:25 hrs.

21. http://es.wikipedia.org/wiki/Energia_eolica

Lunes 18 de agosto de 2008, 13:30 hrs.

22. http://es.wikipedia.org/wiki/Energ%C3%ADa_solar_termoel%C3%A9ctrica

Lunes 18 de agosto de 2008, 16:30 hrs.

23. <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>

Lunes 18 de agosto de 2008, 18:20 hrs.

24. <http://www.cinstrum.unam.mx/secciones/infge/sub1/prese.html>

Lunes 8 de Agosto de 2008, 17:50 hrs.

25. http://www.aulaclie.es/sql/a_8_1_1.htm

Lunes 6 de Octubre de 2008, 17:40 hrs.