



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Sistema de administración de bases de datos sobre
el comportamiento del consumo de energía en
edificios de C.U.

TESIS PROFESIONAL
para obtener el título de
INGENIERA EN COMPUTACIÓN

PRESENTA:

POZAS MONTOYA KARLA JOSSAVET

DIRECTORA DE TESIS

Ing. Carolina Garrido Morelos



Ciudad Universitaria, México, Agosto 2013

CONTENIDO

Índice tablas.....	5
Índice de Figuras.....	8
PRÓLOGO.....	10
CAPÍTULO 1 ESTUDIO PRELIMINAR	11
1.1 Situación actual	12
1.2 Requerimientos	14
1.3 Metodología	16
1.4 Objetivos	25
CAPÍTULO 2 MARCO TEÓRICO	29
2.1 Sistemas Operativos.....	30
2.1.1 Funciones básicas de un Sistema Operativo	31
2.1.2 Características de los Sistemas Operativos.....	35
2.1.3 Clasificación de los Sistemas Operativos.....	35
2.1.4 Linux	42
2.1.4.1 Kernel / Shell	43
2.1.4.2 Características Linux.....	46
2.1.5 Windows.....	49
2.1.5.1 Características del Sistema Operativo Windows.....	54
2.2 Lenguajes de Programación	56
2.3 Recomendaciones del Modelo OSI	63
2.3.1 Servicios de Seguridad.....	66
2.3.2 Autenticación	66
2.3.3 Disponibilidad y Control de Acceso.....	67
2.3.4 Integridad de datos	67
2.3.5 Confidencialidad.....	68
2.3.6 No repudio.....	69
2.4 Sistema Manejador de Bases de Datos (DBMS).....	70
2.4.1 Antecedentes y evolución de las Bases de Datos	71

2.4.2	Definición y características de una Base de Datos	73
2.4.3	Modelo de Bases de Datos	75
2.4.3.1	Modelo Entidad-Relación (MER)	77
2.4.3.2	Modelo Relacional.....	77
2.4.3.3	Normalización	80
2.4.4	Niveles de abstracción de datos.....	82
2.5	Lenguajes de Bases de Datos	84
2.5.1	SQL.....	85
2.5.2	Lenguaje de definición de datos	85
2.5.3	Lenguaje de manipulación de datos.....	86
2.5.4	Lenguaje de control de datos.....	87
2.6	Arquitectura de 3 niveles	88
2.6.1	Modelo cliente - servidor	90
2.6.2	Servidor de Bases de datos	92
2.6.3	Servidor Web.....	94
CAPÍTULO 3	ANÁLISIS.....	95
3.1	Análisis técnico.....	96
3.1.1	Diagnóstico energético.....	96
3.2	Análisis del sistema	97
3.2.1	Herramientas para el desarrollo y diseño	107
CAPÍTULO 4	DISEÑO DEL SISTEMA	110
4.1	Diseño de la base de datos.....	111
4.2	Normalización de la base de datos	126
4.2.1	Modelo Entidad – Relación de cada módulo	139
4.3	Diccionario de datos.....	142
CAPÍTULO 5	IMPLEMENTACIÓN DEL SISTEMA.....	150
5.1	Implementación de la Base de Datos.....	151
5.1.1	Autenticación de usuarios a la Base de Datos.....	153
5.2	Implementación del Sitio Web.....	155

CAPÍTULO 6	PRUEBAS Y MANTENIMIENTO DEL SISTEMA	185
6.1	Pruebas del sistema	185
6.2	Mantenimiento del DBMS.....	187
6.2.1	Políticas de mantenimiento de la Base de Datos.....	188
6.2.2	Políticas de respaldo y recuperación de información	189
	Respaldo y Restauración MySQL de Manera Local.	190
6.2.3	Políticas de roles a usuarios	191
6.3	Manual sistema FI-UNAM Ahorro de Energía.....	193
CAPÍTULO 7	CONCLUSIONES.....	202
7.1	Dificultades en el desarrollo.....	202
7.2	Conocimientos adquiridos.....	203
7.3	Tiempo.....	203
7.4	Costo.....	204
ANEXOS.....		205
	Implementación del sistema	205
1.1.1	Política de permisos en Linux.....	205
1.1.2	Administración de grupos	206
1.1.3	Administración de usuarios.....	207
1.1.4	Administración de bitácoras	209
1.2	Programación de la Base de Datos.....	211
1.3	Programación del sitio Web	212
BIBLIOGRAFÍA.....		214

Índice tablas

Tabla 1.1 Diferencias entre metodologías ágiles y no ágiles	16
Tabla 1.2 Comparación de metodologías de software	16
Tabla 1.3 Tipos de Pruebas	24
Tabla 2.1 Generaciones de los Lenguajes de Programación	57
Tabla 2.2 Términos Relacionales y sus equivalentes.....	78
Tabla 2.3 Ejemplo: Relación Alumno.....	79
Tabla 2.4 Lenguaje de DDL	86
Tabla 2.5 Lenguaje de DML	86
Tabla 3.1 Módulos Generales del sistema	98
Tabla 3.2 Entidades Necesarias para el LEDA.....	99
Tabla 3.3 Privilegios de acceso de cada Entidad	99
Tabla 3.4 Módulo Altas (Administrador).....	100
Tabla 3.5 Módulo Bajas (Administrador)	101
Tabla 3.6 Módulo Actualizar (Administrador).....	102
Tabla 3.7 Módulo Consultas (Administrador)	103
Tabla 3.8 Módulo Altas (Encuestador)	104
Tabla 3.9 Módulo Bajas (Encuestador).....	105
Tabla 3.10 Módulo Actualizar (Encuestador).....	105
Tabla 3.11 Módulo Consultas (Encuestador)	106
Tabla 4.1 Módulos para el diseño	111
Tabla 4.2 Entidades necesarias para el diseño.....	112
Tabla 4.3 Diseño del módulo Altas (Administrador)	112
Tabla 4.4 Diseño del módulo Bajas (Administrador).....	116
Tabla 4.5 Diseño del módulo Actualizar (Administrador)	117

Tabla 4.6 Diseño del módulo Consultas (Administrador)	118
Tabla 4.7 Diseño del módulo Altas (Encuestador)	119
Tabla 4.8 Diseño del módulo Bajas (Encuestador).....	123
Tabla 4.9 Diseño del módulo Actualizar (Encuestador)	123
Tabla 4.10 Diseño del módulo Consultas (Encuestador).....	124
Tabla 4.11 Tablas correspondientes al primer formato de captura	128
Tabla 4.12 Tablas correspondientes al segundo formato de captura.....	128
Tabla 4.13 Tablas correspondientes al tercer formato de captura.....	130
Tabla 4.14 Tablas correspondientes al cuarto formato de captura	131
Tabla 4.15 Tablas correspondientes al quinto formato de captura.....	132
Tabla 4.16 Tablas resultantes del análisis de requerimientos	133
Tabla 4.17 Tabla Encuesta	134
Tabla 4.18 1FN de la Tabla Encuesta.....	134
Tabla 4.19 2FN de la Tabla Encuesta.....	135
Tabla 4.20 Tabla Generalidades	135
Tabla 4.21 1FN de la Tabla Generalidades	136
Tabla 4.22 Tabla Iluminación	136
Tabla 4.23 1FN de la Tabla Iluminación	137
Tabla 4.24 2FN de la Tabla Iluminación	137
Tabla 4.25 Las tablas misceláneas, aire_acondicionado, taller_laboratorio, cal_bombeo Normalizadas.....	138
Tabla 4.26 Diccionario de Datos de la Tabla encuesta_recinto	143
Tabla 4.27 Diccionario de Datos de la Tabla encuesta_iluminación	144
Tabla 4.28 Diccionario de Datos de la Tabla taller_laboratorio.....	145
Tabla 4.29 Diccionario de Datos de la Tabla aire_acondicionado	145
Tabla 4.30 Diccionario de Datos de la Tabla encuestador	146

Tabla 4.31 Diccionario de Datos de la Tabla administrador.....	146
Tabla 4.32 Diccionario de Datos de la Tabla consumo_computo	147
Tabla 4.333 Diccionario de Datos de la Tabla consumo_papeleria	147
Tabla 4.344 Diccionario de Datos de la Tabla consumo_electro	147
Tabla 4.35 Diccionario de Datos de la Tabla dependencia	148
Tabla 4.36 Diccionario de Datos de la Tabla cal_bombeo	148
Tabla 4.377 Diccionario de Datos de la Tabla edificio.....	149
Tabla 4.388 Diccionario de Datos de la Tabla encuesta_ilu_artificial.....	149
Tabla 5.1 Módulos a Implementar	152
Tabla 5.2 Entidades a Implementar	152
Tabla 5.3 Implementación del módulo Altas (Administrador).....	157
Tabla 5.4 Implementación del módulo Bajas (Administrador)	165
Tabla 5.5 Implementación del módulo Actualizar (Administrador).....	167
Tabla 5.6 Implementación del módulo Consultas (Administrador)	170
Tabla 5.7 Implementación del módulo Altas (Encuestador).....	174
Tabla 5.8 Implementación del módulo Bajas (Encuestador)	180
Tabla 5.9 Implementación del módulo Actualizar (Encuestador).....	180
Tabla 5.10 Implementación del módulo Consultas (Encuestador)	182
Tabla 6.1 Pruebas del sistema.....	185

Índice de Figuras

Figura 1.1 Extreme Programing.....	20
Figura 1.2 Modelo Vista Controlador (MVC).....	22
Figura 1.3 Objetivos principales de un sistema operativo	27
Figura 1.4 Actividades generales de un sistema operativo.....	28
Figura 2.1 Perspectiva abstracta de los componentes de un sistema operativo.....	30
Figura 2.2 Estructura Monolítica.....	36
Figura 2.3 Estructura Jerárquica.....	36
Figura 2.4 Estructura de anillos.....	37
Figura 2.5 Sistemas Operativos por servicios.....	37
Figura 2.6 Máquina Virtual.....	41
Figura 2.7 Windows MSDOS.....	50
Figura 2.8 Windows Se7ven	52
Figura 2.9 Modelo de Referencia OSI.....	65
Figura 2.10 Servicios de seguridad.....	66
Figura 2.11 Sistema de Administración de Bases de Datos	70
Figura 2.12 Modelos de Bases de Datos	76
Figura 2.13 Niveles de Abstracción en una Base de Datos	83
Figura 2.14 Arquitectura de 3 Niveles.....	90
Figura 2.15 Modelo Cliente-Servidor	90
Figura 4.1 Diagrama general del SisFIUNAMAhorroDeEnergía	110
Figura 4.2 Primer formato de captura (Datos generales de encuesta).....	127
Figura 4.3 Segundo formato de captura (Misceláneos).....	129
Figura 4.4 Tercer formato de captura (Acondicionamiento Ambiental).....	130
Figura 4.5 Cuarto formato de captura (Bombeo y/o Calefactores)	131

Figura 4.6 Quinto formato de captura (Taller o Laboratorio)	132
Figura 4.7 Tablas obtenidas después de la Normalización	139
Figura 4.8 Tipo de Relaciones en un MER	140
Figura 4.9 Modelo Relacional.....	141
Figura 4.10 Modelo Entidad - Relación del SisFIUNAMAhorroDeEnergía	142
Figura 5.1 Secciones generales del SisFIUNAMAhorroDeEnergía.....	150
Figura 5.2 Proceso general del SisFIUNAMAhorroDeEnergía	156
Figura 5.3 Implementación de módulos (Administrador).....	157
Figura 5.4 Implementación de Entidades para el módulo Altas (Administrador).....	157
Figura 5.5 Implementación de Entidades para el módulo Bajas (Administrador)	165
Figura 5.6 Implementación de Entidades para el módulo Actualizar (Administrador).....	167
Figura 5.7 Implementación de Entidades para el módulo de Consultas (Administrador)	170
Figura 5.8 Implementación de módulos (Encuestador)	173
Figura 5.9 Implementación de Entidades para el módulo de Altas (Encuestador).....	174
Figura 5.10 Implementación de Entidades para el módulo de Altas (Encuestador).....	180
Figura 5.11 Implementación de Entidades para el módulo de Consultas (Encuestador)	182

PRÓLOGO

La tecnología actualmente cuenta con acceso a la información y es muy importante, tanto que, el 90% de la información digital es texto, y en todas sus formas, representa un recurso enorme y valioso de información. Sin embargo, contar con ese recurso e incorporarlo con los datos estructurados ha sido en gran medida una meta no alcanzada y es preciso que se tengan soluciones para atender a todas las necesidades que se presentan.

Mientras que robustas bases de datos basadas en SQL fueron usadas para datos estructurados, máquinas de recuperación fueron usadas para el texto. Estas tecnologías se integran como ambientes costosos y complicados: el texto almacenado en las bases de datos relacionales no es buscado de forma nativa, se le ha almacenado accesado y manipulado de manera específica y precisa, desperdiándose así las bondades de las bases de datos (seguridad, consistencia, concurrencia, integridad, independencia, abstracción de la información, manejo de transacciones y tiempo de respuesta). Esto no sólo es inconveniente para los desarrolladores de una aplicación, sino también para los administradores de la base de datos que se encuentran ante la disyuntiva de administración de datos o de texto. Así, la difícil tarea de optimización de consultas y los beneficios de la transparencia de la base de datos son imposibles de lograr.

En este proyecto se presenta un estudio teórico y recuperación de información sobre los diagnósticos energéticos que se ve aplicado a las bases de datos y así presentar soluciones para el consumo racional de energía. El interés surge porque el uso irracional de la energía en México, está acabando con nuestro medio ambiente, y con el diagnóstico energético nos damos cuenta cuanto es el consumo por día, mes y año, es entonces cuando se deben tomar decisiones para ahorrar energía. Gracias a un sistema direccionado a ese problema, podremos reducir el tiempo de espera en las consultas para la generación del reporte de energía de los edificios que se encuentran en Ciudad Universitaria, incrementar la eficiencia de encuestas de captura donde se recolecta la información de aparatos generadores de electricidad y el riesgo de incrementar los problemas del calentamiento global.

El sistema está basado en el modelo relacional de bases de datos pues ha llegado a su madurez y plenitud, la decisión de la elección de este modelo es por que prevalece en el mundo actual y conociendo la capacidad que ofrecen los sistemas manejadores de bases de datos en cuanto a la manipulación, consulta y administración de datos, es impactante lo mucho que se puede hacer.

La Universidad Nacional Autónoma de México, necesita una solución unificada, basada en estándares y escalable para administrar información de cualquier tipo. Su almacenamiento, acceso, análisis y distribución del sistema, son algunos de los desafíos a los que nos encontramos los involucrados en el área de Ingeniería en Computación y que debemos marcar la pauta para la construcción de software.

Este proyecto que se inicia en la Universidad Nacional Autónoma de México, pero esperando que llegue a todo el país, dará resultados favorables en cuanto al ahorro de energía se refiere.

CAPÍTULO 1 ESTUDIO PRELIMINAR

Este proyecto consiste en una serie de observaciones aleatorias para determinar un estimado en los diferentes elementos del trabajo, es decir, con este muestreo observamos a detalle los elementos que se ven involucrados para la realización de un sistema de administración de bases de datos, que nos pueda definir el comportamiento del consumo de energía generado en todos los edificios existentes en la Universidad Nacional Autónoma de México, con el objetivo de poder llevar a cabo una buena administración de la base de datos.

La base de datos contendrá los formatos de captura necesarios para realizar un análisis y generar un reporte sobre el comportamiento del consumo de energía en cada uno de las dependencias de Ciudad Universitaria, para esto es necesario realizar diagnósticos energéticos.

Un diagnóstico energético es la herramienta fundamental para saber cuánto, cómo, dónde y porqué se consume la energía dentro de una institución o empresa, en este caso para conocer el consumo de energía dentro de cada edificio en Ciudad Universitaria, y así poder establecer el grado de eficiencia de su utilización, para lo cual se requiere una inspección y un diagnóstico energético detallado de los consumos y pérdidas de energía.

Los elementos involucrados para la administración de la base de datos que tenemos que conocer y tener un buen control de los mismos son: el sistema operativo, tanto donde se realizará la administración de la base de datos, como dónde se capturará la información de los formatos que se usaran para la realización de las encuestas, también es necesario conocer el manejador de la base de datos a utilizar, y manejar la información contenida en cada uno de los formatos de encuestas sobre la iluminación, misceláneos, facturas eléctricas, aire acondicionado, equipos de bombeo, equipos de taller y/o laboratorio, las cuales se realizan en cada dependencia.

Para el desarrollo de los formatos de encuesta, primero se ha analizado y definido la estructura de la base de datos, que contendrá la información recabada. Además de contener información previamente obtenida.

Lo que se hace es definir cuáles son las tablas que conformarán a la base de datos, que campos tendrá cada tabla y qué relación hay entre las diferentes tablas. Buscando que toda la información útil se incluya, y que además se evite la redundancia de datos.

Se observa y registra cada elemento, para lograr que haya consistencia en la forma de nombrar lo que se observa, se tiene una reunión con dos o más observadores que están dentro del proyecto a realizar.

Entonces ya con toda esta información recabada y con el amplio conocimiento de cada uno de los elementos que se ven involucrados, podremos entonces, tener un sistema adecuado que nos proporcione la información del consumo de energía, y así poder determinar el uso racional que se le da a la energía en los edificios de Ciudad Universitaria.

1.1 Situación actual

Los sistemas de información existen desde las primeras civilizaciones. Los datos se recopilaban, se estructuraban, se centralizaban y se almacenaban convenientemente. El objetivo inmediato de este proceso era poder recuperar estos mismos datos u otros datos derivados de ellos en cualquier momento, sin necesidad de volverlos a recopilar, paso que solía ser el más costoso o incluso irreplicable. El objetivo de un sistema de información, no obstante, era proporcionar a los usuarios información fidedigna sobre el dominio que representaban, con el objetivo de tomar decisiones y realizar acciones más pertinentes que las que se realizarían sin dicha información.

Hoy en día, sin embargo, solemos asociar a las bases de datos con los ordenadores, y su administración no suele ser manual, sino altamente automatizada. Más concretamente la tecnología actual obliga a la administración de una base de datos a unos tipos de aplicaciones de software específicas denominadas sistemas de administración de bases de datos (SGBD) o, simplemente, sistemas de bases de datos.

La relación que existe entre las bases de datos y las varias aplicaciones que tiene, son demasiadas. Podemos hablar del uso que se les da en la *Banca*, Líneas Aéreas, Universidades, Transacciones de Tarjetas de crédito, Telecomunicaciones, Finanzas, Ventas, Producción, Recursos Humanos.

Las bases de datos forman una parte esencial de casi todas las empresas actuales. A lo largo de las últimas cuatro décadas del siglo veinte, el uso de las bases de datos creció en todas las empresas e incluso en las Universidades de todos los países.

Actualmente en la Universidad Nacional Autónoma de México es fundamental contar con bases de datos aplicadas a distintos campos de investigación, ya que tiene como propósito primordial estar al servicio del país y de la humanidad, formar profesionistas útiles a la sociedad, organizar y realizar investigaciones, principalmente acerca de las condiciones y problemas nacionales, y extender con la mayor amplitud posible, los beneficios de la cultura.

Para cumplir con el propósito que tiene la Universidad Nacional Autónoma de México de servir a la sociedad, se identificaron varios factores que dañan a nuestro medio ambiente. Este proyecto se basa en la necesidad de conocer el comportamiento energético, ya que el mal uso que se le da a la energía es uno de los tantos factores que dañan al país y a la humanidad.

En México, como en muchos otros países, se realiza el diagnóstico energético, ya que permite determinar el grado de eficiencia con que es utilizada la energía.

Es entonces que tenemos que investigar sobre el tema, y así entender el comportamiento que tiene la energía y el uso que le damos.

En la actualidad, las denominadas energías duras o convencionales (carbón, petróleo...) siguen representando nuestras fuentes principales de energía, tanto para el sector residencial como para el productivo. Por lo tanto, dado que no podemos prescindir de estos tipos de portadores energéticos que representan costos millonarios a nuestra economía, es necesario reforzar las

medidas de ahorro y el uso racional de dichos potenciales energéticos, de forma tal que en alguna medida se compensen los gastos que de su utilización se derivan.

El ahorro de cualquier forma de energía y su uso racional inevitablemente presupone la aplicación y control de un programa confeccionado para ese fin, pero dicho programa no se elabora de forma empírica, sino a partir de métodos o procedimientos técnicamente fundamentados, es decir, que debe estar sustentado por los diagnósticos energéticos que permiten identificar en cada lugar que se apliquen (industria, centro de servicio, escuela...) la eficiencia y la responsabilidad con que es utilizada la energía, de cualquier tipo (eléctrica, térmica...). Para este propósito se aporta un conjunto de elementos que permiten realizar y evaluar el diagnóstico energético.

Los objetivos del diagnóstico energético son establecer metas de ahorro de energía, diseñar y aplicar un sistema integral para dicho ahorro de energía, evaluar técnica y económicamente las medidas de conservación y ahorro de energía, y disminuir el consumo de energía sin afectar los niveles de producción [1].

Una parte esencial de un diagnóstico energético es la información histórica del consumo de energía de la Institución; ésta debe ser solicitada por un auditor, es necesaria que la información sea detallada.

Se suele organizar un diagnóstico energético en etapas o fases, sin embargo también es común realizar estudios preliminares de la planta energética a través de un recorrido y una inspección visual de la misma.

En la Universidad Nacional Autónoma de México, en particular la Facultad de Ingeniería, cuya misión es la de formar de manera integral recursos humanos en Ingeniería, realizar investigación acorde con las necesidades de la sociedad, y difundir ampliamente la cultura nacional y universal, que recién inicia en este campo, es insuficiente el diagnóstico de recorrido, ya que difícilmente se puede recolectar la información necesaria.

Es por eso que se realiza un diagnóstico más profundo, éste implica el uso de instrumentación adicional y personal calificado.

Es necesario llevar el diagnóstico a dos niveles [2]:

El nivel 1 proporciona una visión general de los patrones de utilización y los costos de energía.

El nivel 2 proporciona información completa de todos los equipos y elementos involucrados de cada dependencia, su relación con el uso eficiente y racional de energía. Esto proporcionará una guía para ajustar la recolección de información y así implementar un sistema de administración una base de datos que cumpla con los requisitos que se piden.

Ambos niveles se llevan a cabo en formatos de captura que son proporcionados por la Facultad de Ingeniería.

Actualmente, los datos se capturan a mano, por alumnos de la Facultad de Ingeniería. Ya una vez obtenida la información de cada edificio, se procede a vaciar todos los datos en hojas de cálculo, teniendo en cuenta que pueden surgir errores, ya que la recopilación de información puede volverse confusa debido al tipo de captura con la que se está llevando a cabo, es decir, como la captura de cada formato es escrita a mano, existe ambigüedad con respecto a *errores humanos* de quien realizó la encuesta.

En la captura de los datos ya en hojas de cálculo, se requiere de mucho tiempo, ya que se tiene que ir revisando a detalle cada dato escrito en las encuestas, esto hace que el diagnóstico se realice de manera bastante lenta y poco fiable.

La situación actual nos lleva a generar un sistema de administración de base de datos que nos brinde esa eficiencia, rapidez y fiabilidad, que necesitamos para realizar un diagnóstico energético, y así conocer el comportamiento del consumo de energía en cada uno de los edificios de Ciudad Universitaria.

1.2 Requerimientos

Debido a la poca conciencia que existe en cuanto al consumo razonable de energía, la Universidad Nacional Autónoma de México, en conjunto con la Facultad de Ingeniería, han decidido realizar un diagnóstico energético para conocer el comportamiento del uso de energía en cada dependencia que se encuentra en Ciudad Universitaria, para esto, se requiere un sistema que genere los formatos de captura y puedan ser llenados con ayuda de un dispositivo móvil, ya sea laptop u otros, y así optimizar el uso de recursos, ofreciendo confiabilidad y consistencia de datos en un menor tiempo de captura.

La necesidad es disponer de un software que facilite la realización de encuestas de los locales que conforman a los edificios de las dependencias así como de un sistema informático que automatice el proceso de análisis de información para la generación de propuestas de mejoramiento y disminución del consumo eléctrico.

Para la realización del software se consideraron los siguientes elementos: equipo disponible y las restricciones de cada uno de los equipos.

De las restricciones que se presentan, está el hecho de que no se dispone de presupuesto para la adquisición de licencias de software privado, que corresponden a herramientas de desarrollo, como son IDE's, compiladores y generadores de interfaces gráficas.

Se requiere contar con un gestor de bases de datos relacionales (RDBMS), herramienta que permite almacenar y gestionar la información que se genere durante las encuestas y durante los análisis. Con la ventaja de que en su uso, se puede tener esquemas de almacenamiento que no contengan redundancia de datos, y que permita realizar consultas complejas, aún aquellas que no se han considerado; buscando tener una aplicación que sea adaptable a necesidades futuras.

Para el caso de herramientas de desarrollo que se corresponde con software libre, la aplicación resultante no está condicionada ni limita a un uso exclusivo o particular, lo que garantiza que la

UNAM puede hacer lo que crea más conveniente con el software desarrollo con dichas herramienta sin tener que aportar algo a terceros (pago de regalías por ejemplo).

También se requiere de una aplicación Web, para poder verificar los datos de las encuestas sin importar el lugar donde se encuentre la persona que verificará dichos datos, bastará con tan sólo conectarse a la web y obtener la información que se desee, en ésta aplicación también se podrán hacer altas, bajas, cambios o simplemente consultas.

Para el caso del gestor de bases de datos, la ventaja es que se podrá desarrollar la aplicación y conectarse con la base de datos, sin restricciones de número de accesos o de personas que pueden ingresar información simultáneamente.

La base de datos se podrá conectar con la aplicación web y así obtener información no redundante y confiable.

La aplicación web y la base de datos trabajan en conjunto asegurando un mejor rendimiento y mayor eficacia para la captura de las encuestas.

En este caso el objeto de estudio es de la base de datos que se requiere.

La base de datos será utilizada muchas veces a lo largo del programa de ahorro de energía, debiendo ser tan completa y exacta como sea posible, debe tener un parecido exacto a los formatos de captura que se mencionaron con anterioridad. La base de datos debe ser perfeccionada a medida que el programa progrese; además, incluirá los siguientes aspectos:

- a) Nombre de la dependencia, edificio, recinto al cual se le está aplicando la encuesta, indicando que uso se le da al recinto, el número de personas que ocupan el recinto, edad promedio de los ocupantes, dimensiones del local y la altura de la fuente luminosa.
- b) Nombre del responsable de la encuesta, fecha y hora en que realiza la encuesta
- c) Consumos y factura de todos los elementos energéticos que se utilizan en cada uno de los edificios.
- d) Tipo de iluminación, volúmenes de producción
- e) Inventario de equipos consumidores de energía

Se deben reunir datos de todas las formas de energía sobre una base periódica que abarca generalmente un mes o el periodo de la factura de la compañía de servicio público.

Se deben reunir los datos de producción para el mismo periodo que el consumo y costo de la energía, de tal manera que se puedan correlacionar los datos sobre la producción; deben considerarse las unidades normalmente empleadas en la dependencia para medir la producción (por ejemplo, tonelada de producto almacenado, piezas producidas).

No siempre es posible hacer un inventario completo de los equipos que consumen energía en las dependencias. Sin embargo, se deberá hacer lo posible para identificar al menos el equipo que signifique el 85% del consumo.

Los equipos se clasifican de acuerdo con el tipo de energía que consumen. Los regímenes o capacidades de los equipos para las distintas fuentes de energía deben ser sumados y comparados con el consumo actual de energía, obtenido de la base de datos de consumos de energía.

1.3 Metodología

Las metodologías en el desarrollo software nos proporcionan las guías para poder conocer todo el camino a recorrer desde antes de empezar la implementación, con lo cual se asegura la calidad del producto final, así como también el cumplimiento en la entrega del mismo, en un tiempo estipulado. Las metodologías del desarrollo del software se clasifican en tradicionales o convencionales y ágiles [3].

Tabla 1.1 Diferencias entre metodologías ágiles y no ágiles

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparados para cambios durante el proyecto.
Impuestas externamente.	Impuestas internamente (por el equipo).
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio.
Más artefactos.	Pocos artefactos.
Más roles.	Pocos roles.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.

Actualmente existen muchas metodologías para el desarrollo de software, por lo que a continuación se presenta un cuadro comparativo entre las metodologías más conocidas:

Tabla 1.2 Comparación de metodologías de software

Modelo	Características	Ventajas	Desventajas
<i>Rational Unified Process (RUP)</i>	Metodología tradicional o ágil. Es iterativo e incremental. Dirigido por los casos de uso. Centrado en la arquitectura. Desarrollo basado en componentes. UML como lenguaje de Modelado. Proceso integrado. Forma disciplinada de asignar tareas y responsabilidades. Incluye artefactos (productos tangibles del proceso) y roles (papel que desempeña una persona en un determinado momento). Administración de requisitos. El cliente interactúa con el equipo de desarrollo mediante reuniones. Control de cambios. RUP se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición.	Es el proceso de desarrollo más general que existe. Se basa en las mejores prácticas que se han intentado y se han probado en el campo. Da énfasis en los requisitos y el diseño. Asegura la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos. Progreso visible en las primeras etapas. Temprana retroalimentación que se ajuste a las necesidades reales. Administración de la complejidad. Conocimiento adquirido en una iteración puede aplicarse de iteración a iteración.	Un cambio en las últimas etapas incrementaría el costo. La puesta en práctica rápida de características se retrasa hasta que se han identificado y se ha probado una arquitectura. Es más apropiada para proyectos grandes dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios. Alta complejidad.

Modelo	Características	Ventajas	Desventajas
<i>Extreme Programming (XP)</i>	<p>Metodología ágil.</p> <p>Para proyectos de corto plazo con un equipo pequeño y rápida entrega.</p> <p>El usuario final es parte del equipo.</p> <p>Refabricación.- se basa en la utilización repetitiva de código.</p> <p>Programación en pares.</p> <p>Las fases de la metodología XP son: Planeación, Diseño, Codificación y Pruebas.</p> <p>Define 4 variables para proyectos de software: costo, tiempo, calidad y alcance.</p>	<p>Las iteraciones cortas permiten la retroalimentación.</p> <p>Se harán pruebas todo el tiempo, no sólo cada nueva clase (pruebas unitarias) sino que también los clientes comprobarán que el proyecto va satisfaciendo los requisitos (pruebas funcionales).</p> <p>La participación del cliente en el equipo de trabajo.</p> <p>Flexibilidad al cambio.</p>	<p>El desarrollo de software riesgoso y difícil de controlar.</p> <p>Se basa en prácticas inestables.</p> <p>El rol del cliente no refleja los diferentes intereses, habilidades y fuerzas a las que se enfrentan los programadores durante el desarrollo de proyectos.</p> <p>Se trabaja exclusivamente en las necesidades del momento.</p>
<i>Cascada</i>	<p>Metodología tradicional.</p> <p>Orden riguroso de las etapas del ciclo de la vida del software de tal forma que el inicio de una etapa debe esperar a la finalización de la etapa inmediatamente anterior.</p> <p>Cada fase genera documentación para la siguiente. Esta documentación debe ser aprobada.</p> <p>Requiere disponer de unos requisitos completos y precisos al principio del desarrollo.</p> <p>Todas las etapas deben desarrollarse para que el proyecto tenga éxito.</p> <p>Las fases continúan hasta que el objetivo es cumplido.</p> <p>Las fases del modelo en cascada son: Ingeniería y análisis del sistema, Análisis de requerimientos, Diseño, Codificación, Pruebas y Mantenimiento.</p>	<p>Ayuda a minimizar los gastos de la planificación porque permite realizarla sin problemas.</p> <p>Facilita la administración del desarrollo.</p> <p>Evita una fuente común de errores importantes.</p> <p>Se utiliza correctamente para ciclos en los que se tiene una definición estable del producto.</p>	<p>Si se cambia el orden de las fases, el producto será de menor calidad.</p> <p>Se tarda mucho tiempo en pasar por todo el ciclo.</p> <p>Dificultad para especificar claramente los requerimientos al comienzo del proyecto (no permite flexibilidad en los cambios).</p> <p>Los errores de análisis y diseño son costosos de eliminar, y se propagan a las fases siguientes.</p> <p>Se genera mucho mantenimiento inicial debido al periodo de congelación de requisitos y éste recae, en su mayor parte, sobre el código fuente, que en consecuencia se va deteriorando y resultando cada vez más difícil de mantener.</p>
<i>SCRUM</i>	<p>Metodología ágil.</p> <p>El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días.</p> <p>El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.</p> <p>Reuniones a lo largo del proyecto, una reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.</p> <p>Para proyectos con un rápido cambio de requisitos.</p>	<p>Entrega de un producto funcional al finalizar cada Sprint.</p> <p>Posibilidad de ajustar la funcionalidad en base a la necesidad de negocio del cliente.</p> <p>Visualización del proyecto y comprometidos con el proyecto, toda vez que ellos definieron el alcance y se auto-administran.</p>	<p>No genera toda la evidencia o documentación de otras metodologías.</p> <p>No es apto para todos los proyectos.</p> <p>Tal vez sea necesario complementarlo con otros procesos (XP).</p>
<i>Espiral</i>	<p>Metodología tradicional.</p> <p>Las actividades de este modelo se conforman en una espiral, en la que cada iteración representa a un conjunto de actividades.</p> <p>Para cada ciclo del modelo en espiral hay cuatro actividades:</p> <ul style="list-style-type: none"> • Determinar o fijar objetivos. • Análisis del riesgo. • Planificar. • Desarrollar, verificar y validar (probar). <p>Toma en consideración explícitamente el riesgo, esta es una actividad importante en la administración del proyecto.</p>	<p>Reduce riesgos del proyecto.</p> <p>Integra el desarrollo con el mantenimiento.</p> <p>Utiliza las fases de modelo tradicionales.</p> <p>Se centra en la eliminación de errores y alternativas poco atractivas.</p> <p>Mejora al modelo en cascada enfatizando la naturaleza iterativa del proceso de diseño.</p> <p>Permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.</p> <p>Puede adaptarse y aplicarse a lo largo de la vida del software de computadora, no terminal cuando se entrega el software.</p>	<p>Genera mucho tiempo en el desarrollo del sistema.</p> <p>Costoso.</p> <p>Requiere experiencia en la identificación de riesgos y habilidad para saber cuándo detener la evolución.</p> <p>Consumo muchos recursos.</p> <p>Mostrar al cliente que el enfoque evolutivo es controlable.</p>

Modelo	Características	Ventajas	Desventajas
<i>Prototipos</i>	<p>Metodología tradicional.</p> <p>Se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición.</p> <p>Entonces se plantea con rapidez una iteración de</p>	<p>Es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.</p> <p>También ofrece un mejor enfoque cuando el responsable del desarrollo del software</p>	<p>El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final.</p> <p>A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos</p>

	<p>construcción de prototipos y se presenta el modelado (en forma de un diseño rápido). El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. El diseño rápido conduce a la construcción de prototipo, el cual es evaluado por el cliente o el usuario para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollara. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe de hacer y el cliente vea resultados a corto plazo.</p>	<p>está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.</p>	<p>importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte de los casos a reconstruirlo una vez que el prototipo ha cumplido su función. En aras de desarrollar rápidamente el prototipo, el desarrollador suele tomar algunas decisiones de implementación poco convenientes.</p>
<p><i>Adaptive Software Development (ASD)</i></p>	<p>Metodología ágil. Iterativo. Orientado a los componentes software más que a las tareas. Tolerante a cambios. Guiado por los riesgos. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje.</p>	<p>Utiliza información disponible acerca de cambios para mejorar el comportamiento del software. Promulga colaboración, la interacción de personas.</p>	<p>Usa los resultados de cada ciclo de desarrollo para adaptar el siguiente. Ambiente imprevisible que genera incertidumbre.</p>
<p><i>Incremental</i></p>	<p>Metodología tradicional. Es una combinación del Modelo de Cascada y Modelo Evolutivo. Es iterativo por naturaleza. Es un desarrollo inicial de la arquitectura completa del sistema, seguido de incrementos y versiones parciales del mismo. Cada incremento tiene su propio ciclo de vida y agrega funcionalidad adicional o mejorada sobre el sistema. Conforme se completa cada etapa, se verifica e integra la versión con las demás versiones ya completadas del sistema. En una visión genérica. El proceso se divide en cuatro partes: Análisis, Diseño, Código, Prueba. En este método se mantiene al cliente en constante contacto con los resultados obtenidos en cada incremento.</p>	<p>Los clientes no esperan hasta el fin del desarrollo para utilizar el sistema. Pueden empezar a usarlo desde el primer incremento. Los clientes pueden aclarar los requisitos que no tengan claros conforme ven las entregas del sistema. Se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento. Las partes más importantes del sistema sin entregadas primero, por lo cual se realizan más pruebas en estos módulos y se disminuye el riesgo de fallos. Se puede financiar el proyecto por partes.</p>	<p>Cada incremento debe ser pequeño para limitar el riesgo. Cada incremento debe aumentar la funcionalidad. Es difícil establecer las correspondencias de los requisitos contra los incrementos. Es difícil detectar las unidades o servicios genéricos para todo el sistema. Hay costos ocultos en su implementación, ya que se incorporan varias actividades a realizar por el equipo.</p>

Las metodologías existentes nos permiten llevar a cabo el desarrollo de un sistema, depende de los requerimientos del proyecto a realizar así como las necesidades a satisfacer, la elección de una de ellas ya que poseen características, fases, principios, etc., propios, que marcan la diferencia entre una y otra, como se representa en la **Tabla 1.2 Comparación de metodologías de software** [3], donde podemos observar las características principales así como sus ventajas y desventajas de las metodologías más conocidas y aplicadas en este campo.

De acuerdo a lo anterior, se realizó un análisis exhaustivo de dichas metodologías y con base a ese análisis el sistema que se implementará y que será descrito posteriormente, se ha determinado que la metodología que se usará en el desarrollo de este proyecto será la de *Extreme Programming (XP)*, ya que necesitamos que la metodología sea ágil, pues es sin duda uno de los temas recientes en ingeniería de software que están acaparando gran interés, y se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios.

Roles XP [3].

Programador: escribe las pruebas necesarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

Cliente: escribe las historias de usuario y las pruebas funcionales para validar su implementación.

Encargado de pruebas (tester): ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (tracker): proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. Determina cuando es necesario realizar algún cambio para lograr los objetivos de cada iteración.

Entrenador (coach): responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

Gestor (Big boss): es el vínculo entre los clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

Esta metodología proporciona todas las bases y principios para llevar a cabo la elaboración del sistema con éxito. Los cuales se presentan a continuación:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

La metodología XP consiste en cuatro fases que son: Planificación, Diseño, Codificación y Pruebas.

La metodología no ha sido concebida como un mapa de carreteras para conducir un diagnóstico, ya que las actividades hechas en una planta no necesariamente tienen que realizarse en otra. Sin embargo, los lineamientos generales de aproximación mostrados a través de un diagrama de flujo de actividades, pueden servir como una guía para realizar un proyecto que ayude a la optimización de los diagnósticos energéticos.

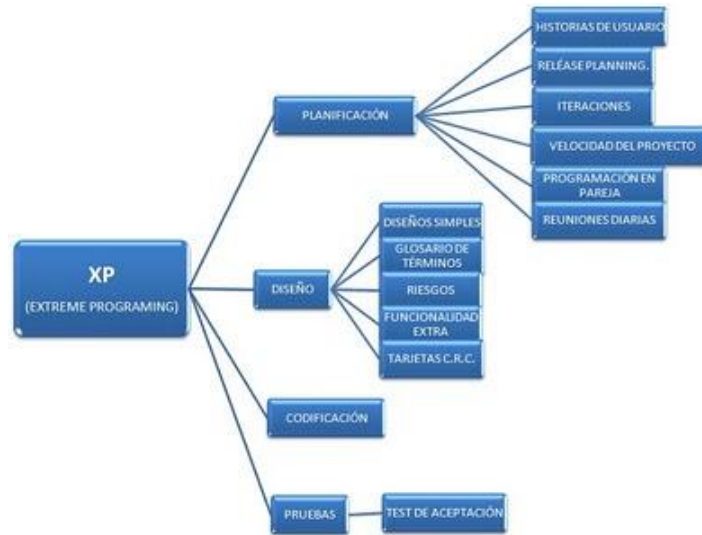


Figura 1.1 Extreme Programing

Fase 1: Planificación

En esta primera etapa, se plantea a grandes rasgos los intereses para la primera entrega del sistema a ser desarrollado. Más adelante, el cliente establece la prioridad de cada historia de usuario, y correspondiente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.

El interés que se tiene en la realización de este proyecto es el de, a grandes rasgos, generar un sistema gestor de bases de datos que brinde eficiencia y mejorar el tiempo de respuesta para la consulta de diagnósticos energéticos realizados en cada edificio de Ciudad Universitaria.

Fase 2: Diseño

En esta fase, se establece una arquitectura del sistema, un modelo de la base de datos y la elección de un manejador para la base de datos, que pueda ser realizada durante el resto del proyecto.

En este sistema gestor se define el modelo entidad-relación para la base de datos que contendrá información de los formatos de captura, que son de vital importancia para la realización de un diagnóstico energético.

La base de datos depende de la buena captura de información. Es importante recopilar y registrar los datos necesarios de manera adecuada para su procesamiento y presentación, mostrar las diferentes técnicas de registro y procesamiento de datos, ya que estas son las que caen dentro de la estructura de la base de datos y así poder tener una buena administración de la misma.

Los datos consisten primordialmente de los registros del edificio encuestado de los cuales obtenemos el consumo histórico de energía. Incluyen registros de consumos de energía de recinto

por recinto, inventarios de equipo, iluminación, misceláneos, y datos de construcción de las instalaciones.

Los datos deben ser ensamblados en una forma conveniente para su óptima operación, el Instituto de Ingeniería desarrolla formatos de captura para la realización del diagnóstico, cada una de estas hojas de captura debe ser muy similar a las tablas que se generarán en la base de datos.

La recopilación de información puede ser lenta, pues en la mayoría de los edificios el control al que más tiempo se dedica no es el de las facturas y de ahí habrá que extraer la información requerida.

Se generan las tablas y diseños para la estructura de la base de datos. Todas las tablas de la base de datos están vinculadas a la tabla encuesta y las relaciones que tiene cada una de las entidades.

Como ya se cuenta con la información requerida, ahora solo se realiza la migración de datos.

Los datos se encuentran en hojas de cálculo, entonces, la migración de datos se ve sencilla.

Hablamos de migración de datos cuando nos referimos al traspaso de información entre bases de datos. Tenemos una aplicación sobre una base de datos hecha en Excel pero ahora necesitamos un sistema gestor de bases de datos potente, ya que la captura de datos para generar un reporte de los diagnósticos energéticos es muy grande.

En este caso, los datos, que estarán en formato "excel" deberán pasar a formato "sqlserver". La migración de los datos consiste en convertir los datos desde un sistema de base de datos a otro. Esta migración conlleva la creación de tablas o modificación de las existentes, cambios en algunos tipos de datos que existen en una base de datos pero no en otras.

La implementación del sistema se realizará en un LAMP.

Linux (Fedora)

Apache

MySQL

PHP

Fase 3: Codificación

La fase de codificación requiere de revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.

En la programación de la aplicación, se seguirá el patrón de diseño Modelo-Vista-Controlador, mismo que se describe a continuación:

MODEL-VIEW-CONTROLLER (MVC)

Modelo Vista Controlador (MVC) [4] es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de administración de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. En la figura 1.1 podemos observar el comportamiento del MVC.

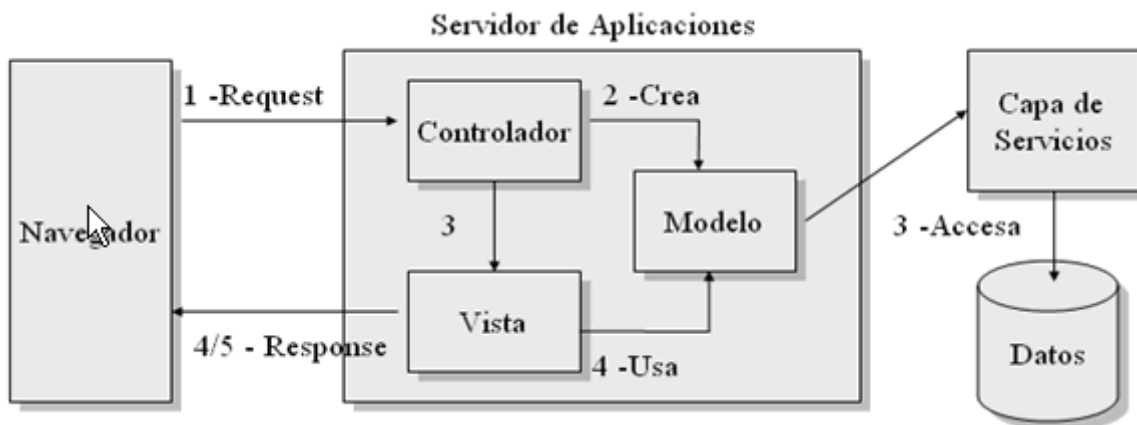


Figura 1.2 Modelo Vista Controlador (MVC)

Descripción del patrón

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la *vista* y su *controlador* facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Muchos de los sistemas informáticos utilizan un Sistema de Administración de Base de Datos para gestionar los datos: en líneas generales del **MVC** corresponde al modelo. La unión entre *capa de presentación* y *capa de negocio* conocido en el paradigma de la Programación por capas representaría la integración entre **Vista** y su correspondiente **Controlador** de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la *capa visual gráfica* de su correspondiente *programación y acceso a datos*, algo que mejora el desarrollo y mantenimiento de la *Vista* y el *Controlador* en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de **MVC**, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz- vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón de observador para proveer cierta dirección errónea entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. *Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.*
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

¿Por qué utilizar MVC?

Porque es un patrón de diseño de software probado y la aplicación se puede desarrollar rápidamente, de forma modular y mantenible. Separar las funciones de la aplicación en modelos, vistas y controladores, hace que la aplicación sea muy ligera. Las características nuevas se añaden y las antiguas toman automáticamente una forma nueva.

Fase 4: Pruebas

Mientras la primer versión del sistema se encuentra en diseño y codificación, el proyecto debe mantener al sistema en funcionamiento al mismo tiempo que se generan nuevas y mejores versiones del sistema. Para realizar esto se requiere de tareas de soporte para el cliente.

En la realización de pruebas al sistema se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

Tipos de pruebas

Tabla 1.3 Tipos de Pruebas

• Pruebas unitarias	• Pruebas de aceptación
• Pruebas funcionales	• Pruebas de regresión
• Pruebas de Integración	• Pruebas de carga
• Pruebas de validación	• Pruebas de prestaciones
• Pruebas de sistema	• Pruebas de recorrido
• Caja blanca (sistemas)	• Pruebas de mutación
• Caja negra (sistemas)	• Pruebas concurrentes

El tipo de prueba que se hace en XP es el de aceptación, también conocido como el **Test case**. Consta de un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio. La prueba de aceptación es realizada por un grupo de usuarios finales o los clientes del sistema, para asegurarse que el sistema desarrollado cumple sus requisitos.

Estructura de los casos de prueba

Formalmente, los casos de prueba escritos consisten principalmente en tres partes con subdivisiones:

Introducción/visión general contiene información general acerca de los Casos de Prueba.

- **Identificador** es un identificador único para futuras referencias, por ejemplo, mientras se describe un defecto encontrado.
- **Caso de prueba dueño/creador** es el nombre del analista o diseñador de pruebas, quien ha desarrollado pruebas o es responsable de su desarrollo.
- **Versión** la actual definición del caso de prueba.
- **Nombre** el caso de prueba debe ser un título entendible por personas, para la fácil comprensión del propósito del caso de prueba y su campo de aplicación.
- Identificador de **requerimientos** el cual está incluido por el caso de prueba. También aquí puede ser identificador de casos de uso o especificación funcional.
- **Propósito** contiene una breve descripción del propósito de la prueba, y la funcionalidad que chequea.
- **Dependencias**

Actividades de los casos de prueba

- **Ambiente de prueba/configuración** contiene información acerca de la configuración del hardware o software en el cuál se ejecutará el caso de prueba.

- **Inicialización** describe acciones, que deben ser ejecutadas antes de que los casos de prueba se hayan inicializado. Por ejemplo, debemos abrir algún archivo.
- **Finalización** describe acciones, que deben ser ejecutadas después de realizado el caso de prueba. Por ejemplo si el caso de prueba estropea la base de datos, el analista debe restaurarla antes de que otro caso de prueba sea ejecutado.
- **Acciones** pasos a realizar para completar la prueba.
- Descripción de los **datos de entrada**

Resultados

- **Resultados esperados** contiene una descripción de lo que el analista debería ver tras haber completado todos los pasos de la prueba
- **Resultados reales** contienen una breve descripción de lo que el analista encuentra después de que los pasos de prueba se hayan completado. Esto se sustituye a menudo con un **Correcto/Fallido**. Si un caso de prueba falla, frecuentemente la referencia al defecto implicado se debe enumerar en esta columna.

1.4 Objetivos

En un proyecto tan ambicioso como es el de gestionar toda una base de datos dedicada al el comportamiento del consumo de energía en los edificios de Ciudad Universitaria, es necesario cumplir con objetivos específicos para su buen uso.

Un diagnóstico energético es un proyecto complejo, pero puede ser ejecutado adecuadamente cuando se realiza de manera secuencial. Cada paso del diagnóstico es un proceso de estudio para las fases subsecuentes, que lleva de lo general a lo específico.

Los diagnósticos tienen como objetivo transmitir los conocimientos necesarios para identificar y evaluar oportunidades de ahorro de energía en empresas y / o instituciones.

El objetivo del sistema de administración de base de datos es el de:

- a) Proporcionar información técnica-económica suficiente sobre el potencial de ahorro y posibles medidas de aprovechamiento.
- b) Requerir de un mínimo de recursos humanos y materiales para que el costo sea bajo.
- c) Reducir la redundancia de los datos, minimizar errores de captura
- d) Aumentar la confiabilidad de la información

Gracias a un sistema manejador de bases de datos se cumplirán los siguientes aspectos [5]:

- **Abstracción de la información.** El SGBD ahorrará a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios *niveles de abstracción*.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de la base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia.** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad que tiene determinadas condiciones, por ejemplo los roles de usuario, los permisos otorgados y privilegios que se dan solo a cierto tipo de usuario. El sistema no debería aceptar datos o consultas de usuarios que no puedan ser autenticados o de aquellos cuyos privilegios son limitados para solo poder consultar la información, mas no modificarla. El SGBD deberá contar con herramientas que faciliten la programación de este tipo de condiciones.
- **Seguridad.** La información almacenada en la base de datos debe llegar a tener un gran valor. El SGBD debe garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Manejo de transacciones.** Una transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. El SGBD deberá proveer mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.
- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD demora en proporcionar la información solicitada y en almacenar los cambios realizados, como lo son la captura de los datos obtenidos en cada dependencia, edificio, recinto que se encuentran en Ciudad Universitaria.

La base de datos contará con un Sistema Manejador de Bases de Datos, ya que se pretende llevar al máximo esta aplicación, no tan solo obtener una base de datos del comportamiento del consumo de energía en ciudad universitaria si no hacerla crecer realizando diagnósticos energéticos en todos los edificios de la Universidad Nacional Autónoma de México, entonces se tendrán facilidades para la manipulación de grandes volúmenes de datos.

Para esto, será necesario disponer de una o más personas que administren la base de datos, de la misma forma en que suele ser necesario en instalaciones de cierto porte disponer de una o más personas que administren los sistemas operativos.

Con la base de datos y un sistema manejador, se disminuirán drásticamente los tiempos de desarrollo y aumentará la calidad del sistema desarrollado si son bien explotados por los desarrolladores, administradores y usuarios del sistema.

Una vez que se tiene diseñada la base de datos procedemos a realizar la implementación del sistema.

El sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o institución.

Elementos de un sistema de información { Equipo computacional
Recurso Humano

El equipo computacional: el hardware necesario para que el sistema de información pueda operar.

El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema de información realiza tres actividades básicas: entrada, almacenamiento/procesamiento y salida de información.

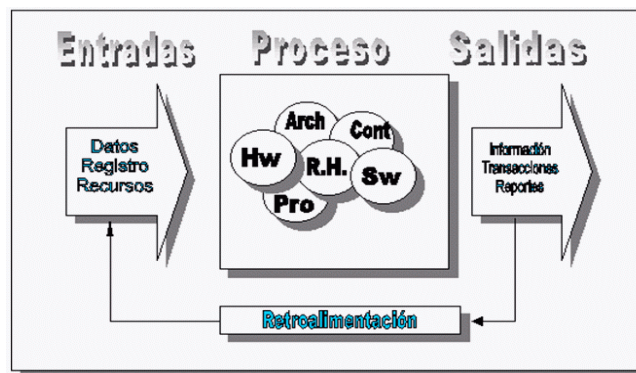


Figura 1.3 Objetivos principales de un sistema operativo

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

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

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.

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

La evaluación de la información obtenida, permite la retroalimentación del sistema.

Durante los próximos años, los Sistemas de Información cumplirán tres objetivos básicos dentro de las organizaciones:

1. Automatización de procesos operativos.
2. Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
3. Lograr ventajas competitivas a través de su implantación y uso.

El sistema a implementar además deberá cumplir con los siguientes objetivos, para así lograr una buena conexión con la base de datos y brindar una interface gráfica agradable para el usuario final:

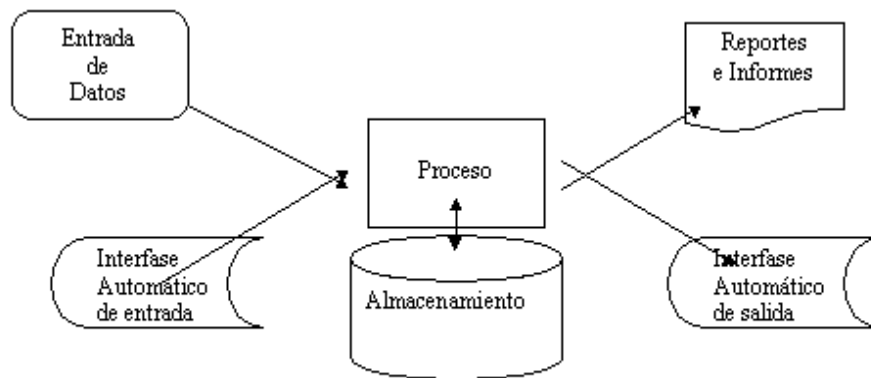


Figura 1.4 Actividades generales de un sistema operativo

- Respalda las operaciones.
- Respalda la toma de decisiones.
- Respalda la ventaja competitiva estratégica.
- Contribuir a la automatización de actividades y procesos.
- Llevar la información de manera oportuna y adecuada de cada edificio en Ciudad Universitaria que así lo requieran.
- Proporcionar un diagnóstico energético de la Universidad Nacional Autónoma de México en un momento dado [6].

CAPÍTULO 2 MARCO TEÓRICO

La complejidad de las organizaciones ha demandado eficiencia en la utilización de datos y generación óptima y oportuna de la información. Gracias al uso de los equipos de cómputo ha sido posible manejar enormes cantidades de datos con mayor rapidez y precisión. A la par, se han dado cambios en las estructuras de datos empleadas, en las técnicas destinadas a la explotación de la información, así como en las interfaces de las aplicaciones y en los sistemas de bases de datos.

Los sistemas de información existen desde las primeras civilizaciones. Los datos se recopilaban, se estructuraban, se centralizaban, y se almacenaban. El objetivo inmediato de este proceso era poder recuperar estos mismos datos u otros datos derivados de ellos en cualquier momento, sin necesidad de volverlos a recopilar, paso que solía ser el más costoso o incluso irreplicable. El objetivo de un sistema de administración, no obstante, era proporcionar información fidedigna sobre el dominio que representaban, con el objetivo de tomar decisiones y realizar acciones más pertinentes que las que se realizaría sin dicha información [6].

Generalmente un sistema de administración de bases de datos, consta de sistemas operativos, una o más bases de datos, aplicaciones de software específicas e interfaces. Para que un sistema automatizado sea una herramienta eficiente que implique la administración de bases de datos, o la implementación de un servidor Web, es necesario contar con los conocimientos fundamentales en áreas especializadas de la ingeniería.

La evolución en el manejo de grandes volúmenes de datos, como por ejemplo investigaciones científicas que se realizaban en la NASA, información de alumnos y asignaturas en las universidades, etc., surge a principios de los años 60's con el comité CODASYL (Conference Of DATA SYSTEMS Language), que representaba a fabricantes de computadoras, agencias gubernamentales, organizaciones de usuarios y universidades; este comité preparó el marco de trabajo de COBOL (COmmon Business Oriented Language) lenguaje que fue diseñado específicamente para el procesamiento de los datos de tipo comercial. La estructura de datos predominante fue el *archivo* [7].

Los archivos estaban por lo general, diseñados para una aplicación determinada o para un grupo de aplicaciones muy similares. Con estos archivos era posible el acceso secuencial y el acceso directo a los registros, pero no una administración de datos, pues solo se necesitaba obtener la información de ciertos datos, ya sea de algún empleado o alumno registrado en la base de datos.

Durante las últimas décadas, las bases de datos han jugado un papel muy importante en el manejo de grandes volúmenes de información; hardware y software (en este caso representado por los DBMS), han tenido que evolucionar para poder manipular las nuevas estructuras de datos y atender las complejas demandas continuas de información.

Para este proyecto de tesis es necesario cumplir con cierto nivel de conocimiento en lo que son los sistemas operativos, lenguajes de programación, las recomendaciones del modelo OSI , los DBMS, lenguajes de bases de datos y realizar análisis técnico sobre la información que vamos a proporcionar con el sistema a implementar.

2.1 Sistemas Operativos

Los sistemas operativos son una parte esencial de cualquier sistema de computación. Los sistemas de computación se pueden dividir a grandes rasgos en cuatro componentes: el hardware, el sistema operativo, los programas de aplicación (o del usuario), y los usuarios, tal como se muestra en la figura 2.1.

El hardware – la unidad central de proceso (CPU), la memoria y los dispositivos de entrada/salida (E/S) – proporcionan los recursos de computación básicos. Los programas de aplicación – como los compiladores, sistemas de base de datos, juegos y programas de negocios – definen las formas en que dichos recursos se utilizan para resolver los problemas de computación de los usuarios. Puede haber muchos tipos de usuarios (personas, máquinas, otras computadoras) tratando de resolver diferentes problemas; por ello, hay muchos tipos de programas de aplicación. El sistema operativo controla y coordina el uso del hardware entre los diversos programas de aplicación de los distintos usuarios [8].

El sistema operativo es un programa que actúa como intermediario entre el usuario de una computadora y el hardware de una computadora. El propósito de un sistema operativo es crear un entorno en el que el usuario pueda ejecutar programas de forma cómoda y eficiente [9].

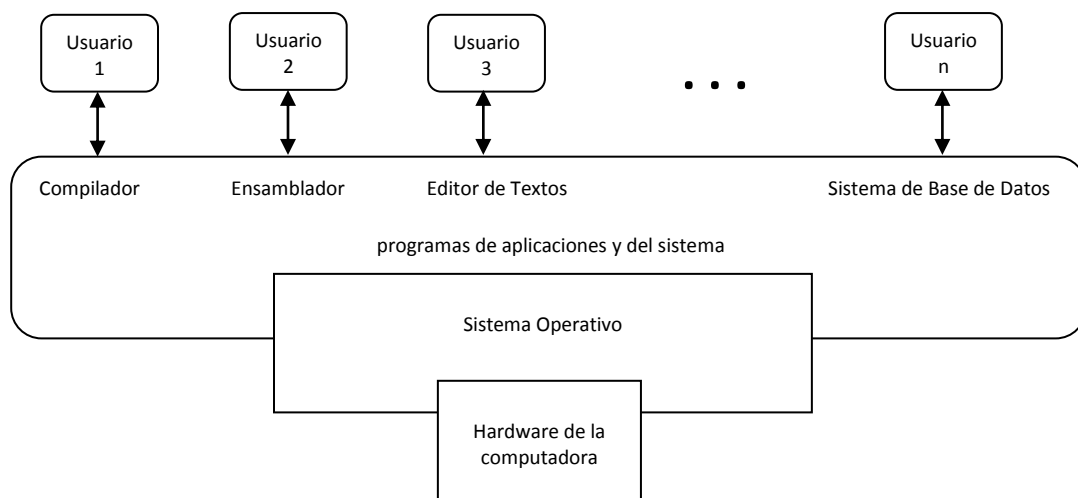


Figura 2.1 Perspectiva abstracta de los componentes de un sistema operativo

El desarrollo de los sistemas operativos abarca desde los primeros sistemas manuales hasta los sistemas multiprogramados y de tiempo compartido actuales.

El sistema operativo debe garantizar el funcionamiento correcto del sistema de la computadora. Para evitar que los programas de usuario interfieran dicho funcionamiento, el hardware debe contar con mecanismos adecuados para asegurar ese buen comportamiento.

Podemos ver al sistema operativo como un *asignador de recursos*. Un sistema de computación tiene muchos recursos (hardware y software) que pueden ser necesarios para resolver un problema: tiempo de CPU, espacio de memoria, espacio para almacenamiento de archivos, dispositivos de E/S, y demás. El sistema operativo actúa como gestor de dichos recursos y los asigna a los programas y usuarios específicos según los necesiten para sus tareas. Dado que puede haber muchas solicitudes de recursos, que tal vez no puedan satisfacerse simultáneamente, el sistema operativo debe decidir cuáles solicitudes atenderá asignando recursos para operar el sistema de computación de manera eficiente y justa [10].

En general no existe una definición perfecta de sistema operativo. Los sistemas operativos existen porque son una forma razonable de resolver el problema de crear un sistema de computación útil. El objetivo fundamental de las computadoras es ejecutar los programas del usuario y facilitar la resolución de problemas de los usuarios. Es con este fin que se construye el hardware de las computadoras. Puesto que el hardware desnudo no es muy fácil de usar, se han desarrollado programas de aplicación de muy diversos tipos, que requieren ciertas operaciones comunes, como las que controlan los dispositivos de E/S. así, las funciones comunes de controlar y asignar recursos se reúnen en un solo programa: *el sistema operativo* [8].

2.1.1 Funciones básicas de un Sistema Operativo

Podemos crear un sistema tan grande y complejo como un sistema operativo sólo si lo dividimos en porciones más pequeñas. Cada una de esas partes deberá ser un componente bien delineado del sistema, con entradas, salidas y funciones cuidadosamente definidas.

Administración de procesos.

Podemos pensar en un proceso como una porción de un programa en ejecución que necesita ciertos recursos, incluidos tiempo de CPU, memoria, archivos y dispositivos de E/S, para llevar a cabo su tarea. Estos recursos se otorgan al proceso al momento en que se crea, o bien se le asignan durante su ejecución.

Un proceso es la unidad de trabajo de un sistema. El sistema consiste en una colección de procesos, algunos de los cuales son procesos del sistema operativo (los que ejecutan código del sistema), siendo el resto procesos de usuario (los que ejecutan código de algún usuario). Potencialmente, todos estos procesos podrían ejecutarse de forma concurrente.

El sistema operativo se encarga de las siguientes actividades relacionadas con la administración de procesos:

- Crear y eliminar procesos tanto de usuario como del sistema
- Suspender y reanudar procesos
- Proveer mecanismos para la sincronización de procesos
- Proveer mecanismos para la comunicación entre procesos
- Proveer mecanismos para manejar bloqueos mutuos (*deadlocks*)

Administración de la memoria principal.

La memoria principal es una matriz grande de palabras o *bytes*, cuyo tamaño va desde cientos de miles hasta cientos de millones. Cada palabra o *byte* tiene su propia dirección. La memoria principal es un depósito de datos a los que se puede acceder rápidamente y que son compartidos por la CPU y los dispositivos de E/S.

La selección de un esquema de administración de memoria para un sistema en particular depende de muchos factores, pero sobre todo del diseño de *hardware* del sistema.

El sistema operativo se encarga de las siguientes actividades relacionadas con la administración de memoria:

- Saber cuáles partes de la memoria se están usando actualmente, y quién las está usando.
- Decidir cuáles procesos se cargarán en la memoria cuando se disponga de espacio.
- Asignar y liberar espacio de memoria según se necesite.

Administración de archivos.

La administración de archivos es uno de los componentes más visibles de un sistema operativo. Las computadoras pueden almacenar información en varios tipos distintos de medios físicos. Los medios más comunes son cinta magnética, disco magnético y disco óptico. Cada uno de éstos tiene sus propias características y organización física.

Un archivo es una colección de información relacionada definida por su creador. Por lo regular, los archivos representan programas y datos. Un archivo consiste en una secuencia de bits, líneas o registros, cuyos significados han sido definidos por su creador.

El sistema operativo establece la correspondencia entre los archivos y los medios físicos, y accede a los archivos a través de los dispositivos de almacenamiento.

El sistema operativo implementa el concepto abstracto de archivo administrando los medios de almacenamiento masivo, como cintas y discos, y los dispositivos que los controlan. Los archivos se organizan normalmente en directorios para facilitar su uso.

El sistema operativo se encarga de las siguientes actividades relacionadas con la administración de archivos:

- Crear y eliminar archivos.
- Crear y eliminar directorios.
- Proveer las primitivas para manipulación de archivos y directorios.
- Establecer la correspondencia de los archivos con el almacenamiento secundario.
- Resguardar los archivos en medios de almacenamiento estables (no volátiles).

Administración del sistema de E/S.

Uno de los objetivos de un sistema operativo es ocultar las peculiaridades de dispositivos de *hardware* específicos de modo que el usuario no las perciba. El *subsistema de E/S* oculta las peculiaridades de los dispositivos de E/S del resto del sistema operativo mismo. Dicho subsistema consiste en:

- Un componente de administración de memoria que incluye el uso de buffers, cachés y spool.
- Una interfaz general con los controladores de dispositivos.
- Drivers (Controladores en software) para dispositivos de hardware específicos.

Sólo el driver del dispositivo conoce las peculiaridades del dispositivo específico al que ya está asignado.

Administración de almacenamiento secundario.

El propósito principal de un sistema de computadora es ejecutar programas. Estos programas, junto con los datos a los que se acceden, deben ser alojados en la memoria principal (*almacenamiento primario*) durante la ejecución. Dado que la memoria principal es demasiado pequeña para dar cabida a todos los datos y programas, y que pierde su información cuando deja de recibir corriente eléctrica, el sistema de computación debe contar con algún *almacenamiento secundario*, para respaldar la memoria principal. Casi todos los programas – incluidos compiladores, ensambladores, rutinas de ordenación editores y formateadores – se guardan en un disco hasta que se cargan en la memoria, y luego utilizan el disco como fuente y como destino de su procesamiento. Por ello, la administración correcta del almacenamiento en disco es extremadamente importante para un sistema computacional.

El sistema operativo se encarga de las siguientes actividades relacionadas con la administración de discos:

- Administración del espacio libre.
- Asignación del almacenamiento.
- Planificación del disco.

El almacenamiento secundario se usa con mucha frecuencia, así que su uso debe ser eficiente. La velocidad de operación global del computador podría depender del subsistema de disco y de los algoritmos que lo manipulan.

Trabajo con redes.

Un sistema *distribuido* es una colección de procesadores que no comparten memoria, dispositivos periféricos ni el reloj. Más bien, cada procesador tiene su propia memoria local y su propio reloj, y se comunica con los otros procesadores a través de distintas líneas de comunicación, como *buses* de alta velocidad o líneas telefónicas. Los procesadores de un sistema distribuido varían en cuanto a tamaño y función; podrían incluir microprocesadores pequeños, estaciones de trabajo, minicomputadoras y sistemas de cómputo de propósito general grandes.

Los procesadores se conectan por medio de una *red de comunicaciones*. La red puede estar total o parcialmente conectada, y su diseño debe considerar estrategias de encaminamiento y conexión, ya también los problemas de contención y seguridad.

El acceso a recursos compartidos permite acelerar los cálculos, ampliar la disponibilidad de los datos y mejorar la confiabilidad. Los sistemas operativos normalmente generalizan el acceso por red como una forma de acceso a archivos; los detalles del trabajo con la red están contenidos en el *driver* de la interfaz con la red.

Sistema de protección.

Si un sistema tiene múltiples usuarios y permite la ejecución concurrente de múltiples procesos, es preciso proteger cada proceso de las actividades de los demás. Con este fin, se incluyen mecanismos para asegurar que sólo aquellos procesos que hayan obtenido la autorización debida del sistema operativo puedan operar con los archivos, segmentos de memoria, procesadores y demás recursos.

El término *protección* se refiere a un mecanismo para controlar el acceso de programas, procesos o usuarios a los recursos definidos por un sistema de computador. Este mecanismo debe incluir alguna forma de especificar los controles que se impondrán, junto con alguna forma de hacerlos cumplir.

La protección mejora la confiabilidad mediante la detección de errores latentes en subsistemas componentes. La detección temprana de errores a menudo puede evitar la contaminación de un subsistema saludable por un subsistema que está fallando. Un sistema orientado hacia la protección permite distinguir entre el uso autorizado y no autorizado.

Sistema de interpretación de órdenes.

Uno de los programas del sistema más importantes de un sistema operativo es el *intérprete de órdenes* o *de comandos*, que es la interfaz entre el usuario y el sistema operativo.

Muchas de las órdenes que se dan al sistema operativo vienen en *enunciados de control*. Cuando se inicia un trabajo nuevo en un sistema por lotes, o cuando un usuario ingresa en un sistema de tiempo compartido, se ejecuta automáticamente un programa que lee e interpreta estos enunciados de control. Dicho programa también se conoce como *intérprete de tarjetas de control*

o *intérprete de línea de comandos* y recibe el nombre común de *Shell*. La función del *Shell* es muy sencilla: obtener la siguiente orden y ejecutarla.

Otros usuarios aprecian los *shells* más potentes, complejos y difíciles de aprender. En algunos de estos *shells* las órdenes se teclean y se exhiben en una pantalla o terminal impresora, y la tecla “*enter*” (o “*return*”) indica que la orden está completa y lista para ejecutarse [9].

Las órdenes mismas sirven para crear y administrar procesos, manejar la E/S, administrar el almacenamiento secundario, gestionar la memoria principal, acceder al sistema de archivos, proteger al sistema y trabajar con redes.

2.1.2 Características de los Sistemas Operativos

Un sistema Operativo cuenta con las siguientes características:

- Proporcionar más comodidad en el uso de una computadora.
- Administrar de manera eficiente los recursos del equipo, ejecutando servicios para los procesos (programas)
- Brindar una interfaz al usuario, ejecutando instrucciones (comandos).
- Permitir que los cambios debidos al desarrollo del propio SO se puedan realizar sin interferir con los servicios que ya se prestaban (evolutividad) [10].

2.1.3 Clasificación de los Sistemas Operativos

Actualmente los sistemas operativos se dividen en tres clasificaciones:

1. visión interna (sistemas operativos por su estructura)
2. sistemas operativos por los servicios que ofrecen
3. visión externa (sistemas operativos por la forma en que ofrecen sus servicios)

1. Visión interna.

Cuando se construye un Sistema Operativo, es necesario contar con dos requisitos, los cuales son:

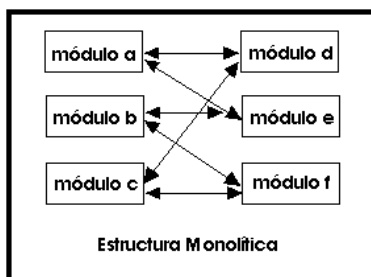
- 1) Requisitos de usuario: Sistema fácil de usar y de aprender, seguro, rápido y adecuado al uso al que se le quiere destinar.
- 2) Requisitos del software: Donde se engloban aspectos como el mantenimiento, forma de operación, restricciones de uso, eficiencia, tolerancia frente a los errores y flexibilidad.

Los sistemas operativos actuales cuentan con distintas estructuras para satisfacer las necesidades que de ellos se quieren obtener.

❖ Estructura Monolítica.

Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra. Las características fundamentales de este tipo de estructura son:

- Construcción del programa final a base de módulos compilados separadamente que se unen a través del *ligador*.
- Buena definición de parámetros de enlace entre las distintas rutinas existentes, que puede provocar mucho acoplamiento.
- Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.



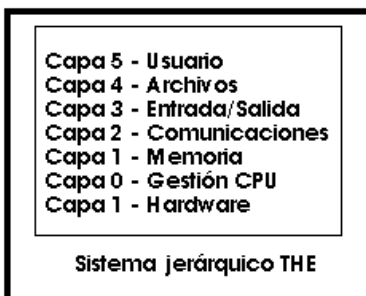
Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y administración, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

Figura 2.2 Estructura Monolítica

❖ Estructura Jerárquica.

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del software, del sistema operativo, donde una parte del sistema contenía *subpartes* y esto organizado en forma de niveles.

Se dividió el sistema operativo en pequeñas partes, de tal forma que cada una de ellas estuviera perfectamente definida y con un claro interface con el resto de elementos.



Se constituyó una estructura jerárquica o de niveles en los sistemas operativos, el primero de los cuales fue denominado THE (Technische Hogeschool, Eindhoven), de Dijkstra, que se utilizó con fines didácticos. Se puede pensar también en estos sistemas como si fueran '*multicapa*'. Multics y Unix caen en esa categoría.

Figura 2.3 Estructura Jerárquica

En esta estructura se basan prácticamente la mayoría de los sistemas operativos actuales.

❖ **Estructura de anillos.**

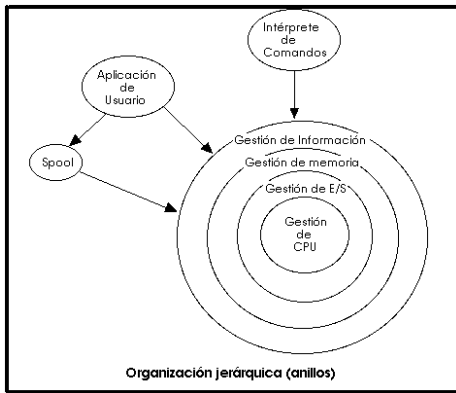


Figura 2.4 Estructura de anillos

Denominada de anillos concéntricos o "rings".

En el sistema de anillos, cada uno tiene una apertura, conocida como puerta o trampa (trap), por donde pueden entrar las llamadas de las capas inferiores. De esta forma, las zonas más internas del sistema operativo o núcleo del sistema estarán más protegidas de accesos indeseados desde las capas más externas. Las capas más internas serán, por tanto, más privilegiadas que las externas.

2. Sistemas operativos por los servicios que ofrecen.

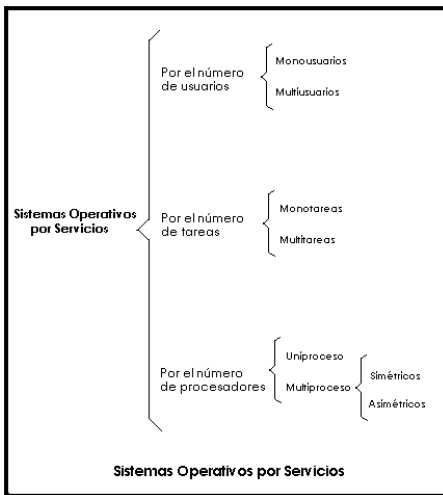


Figura 2.5 Sistemas Operativos por servicios

Esta clasificación es la más comúnmente usada y conocida desde el punto de vista del usuario final. Esta clasificación se comprende fácilmente con el cuadro sinóptico que a continuación se muestra en la figura.

❖ **Por Número de Usuarios.**

- Sistema Operativo Monousuario.

Los sistemas operativos monousuarios son aquéllos que soportan a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales típicamente se han clasificado en este renglón.

En otras palabras los sistemas monousuarios son aquellos que nada más puede atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se esté ejecutando.

- Sistema Operativo Multiusuario.

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas

en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

En esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos. Este tipo de sistemas se emplean especialmente en redes. En otras palabras consiste en el fraccionamiento del tiempo (timesharing).

❖ **Por el Número de Tareas.**

- Sistema Operativo Monotarea.

Los sistemas monotarea son aquellos que sólo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo pero cada uno de ellos puede estar haciendo solo una tarea a la vez.

Los sistemas operativos monotareas son más primitivos y, solo pueden manejar un proceso en cada momento o que solo puede ejecutar las tareas de una en una.

- Sistema Operativo Multitarea.

Un sistema operativo multitarea es aquél que le permite al usuario estar realizando varias labores al mismo tiempo.

Es el modo de funcionamiento disponible en algunos sistemas operativos, mediante el cual una computadora procesa varias tareas al mismo tiempo. Existen varios tipos de multitareas. La conmutación de contextos (context Switching) es un tipo muy simple de multitarea en el que dos o más aplicaciones se cargan al mismo tiempo, pero en el que solo se está procesando la aplicación que se encuentra en primer plano (la que ve el usuario. En la multitarea cooperativa, la que se utiliza en el sistema operativo Macintosh, las tareas en segundo plano reciben tiempo de procesado durante los tiempos muertos de la tarea que se encuentra en primer plano (por ejemplo, cuando esta aplicación está esperando información del usuario), y siempre que esta aplicación lo permita.

Un sistema operativo multitarea se distingue por su capacidad para soportar la ejecución concurrente de dos o más procesos activos. La multitarea se implementa generalmente manteniendo el código y los datos de varios procesos simultáneamente en memoria y multiplexando el procesador y los dispositivos de E/S entre ellos.

La multitarea suele asociarse con soporte hardware y software para protección de memoria con el fin de evitar que procesos corrompan el espacio de direcciones y el comportamiento de otros procesos residentes.

❖ **Por el Número de Procesadores**

- Sistema Operativo de Uniproseso.

Un sistema operativo uniproseso es aquél que es capaz de manejar solamente un procesador de la computadora, de manera que si la computadora tuviese más de uno le sería inútil. El ejemplo más típico de este tipo de sistemas es el DOS y MacOS.

- Sistema Operativo de Multiproseso.

Un sistema operativo multiproseso se refiere al número de procesadores del sistema, que es más de uno y éste es capaz de usarlos todos para distribuir su carga de trabajo. Generalmente estos sistemas trabajan de dos formas: simétrica o asimétricamente.

- Asimétrica.

Cuando se trabaja de manera asimétrica, el sistema operativo selecciona a uno de los procesadores el cual jugará el papel de procesador maestro y servirá como pivote para distribuir la carga a los demás procesadores, que reciben el nombre de esclavos.

- Simétrica.

Cuando se trabaja de manera simétrica, los procesos o partes de ellos (threads) son enviados indistintamente a cual quiera de los procesadores disponibles, teniendo, teóricamente, una mejor distribución y equilibrio en la carga de trabajo bajo este esquema.

Se dice que un thread es la parte activa en memoria y corriendo de un proceso, lo cual puede consistir de un área de memoria, un conjunto de registros con valores específicos, la pila y otros valores de contexto.

3. Visión externa.

Esta clasificación se refiere a la visión del usuario, el cómo accesa a los servicios. Bajo esta clasificación se pueden detectar dos tipos principales:

- 1) Sistemas operativos de red
- 2) Sistemas operativos distribuidos.

❖ **Sistema Operativo de Red**

Los sistemas operativos de red se definen como aquellos que tiene la capacidad de interactuar con sistemas operativos en otras computadoras por medio de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades. El punto crucial de estos sistemas es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que desee acceder.

El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell Netware.

Los Sistemas Operativos de red más ampliamente usados son: Novell Netware, Personal Netware, LAN Manager, Windows NT Server, UNIX, LANtastic.

❖ **Sistemas Operativos Distribuidos**

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso) en una sola máquina virtual que el usuario accesa en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo habitual.

La complejidad que estos sistemas operativos tienen que soportar es:

- Distribuir los procesos en las varias unidades de procesamiento
- Reintegrar sub-resultados
- Resolver problemas de concurrencia y paralelismo
- Recuperarse de fallas de algunos recursos distribuidos
- Consolidar la protección y seguridad entre los diferentes componentes del sistema y los usuarios.

Los avances tecnológicos en las redes de área local y la creación de microprocesadores de 32 y 64 bits lograron que computadoras más o menos baratas tuvieran el suficiente poder en forma autónoma para desafiar en cierto grado a los mainframes, y a la vez se dio la posibilidad de intercomunicarlas, sugiriendo la oportunidad de partir procesos muy pesados en cálculo en unidades más pequeñas y distribuirlas en los varios microprocesadores para luego reunir los sub-resultados, creando así una máquina virtual en la red que exceda en poder a un mainframe.

Las razones para crear o adoptar sistemas distribuidos se dan por dos razones principales:

- Por necesidad (debido a que los problemas a resolver son inherentemente distribuidos)
- Porque se desea tener más confiabilidad y disponibilidad de recursos

Los sistemas distribuidos deben de ser muy confiables, ya que si un componente del sistema se compone otro componente debe de ser capaz de reemplazarlo [8].

Máquina Virtual.

La Máquina Virtual fue creada con el propósito de brindar apoyo a los sistemas operativos, y poder aprovechar todos los recursos de una sola máquina física, incrementando así la potencia de cada computadora.

Entonces podemos describir a una máquina virtual como un tipo de sistema operativo que presenta una interface a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la multiprogramación y la máquina extendida. El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.

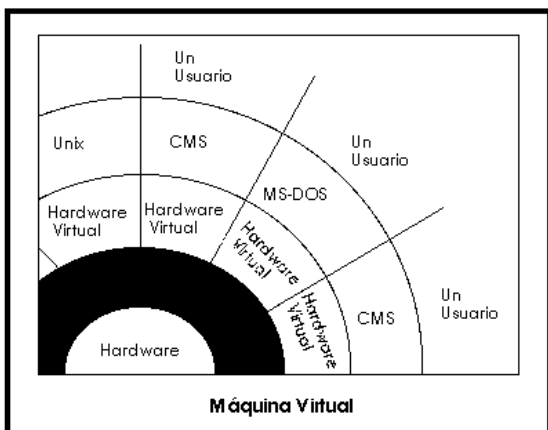


Figura 2.6 Máquina Virtual

El núcleo de estos sistemas operativos se denomina monitor virtual y tiene como misión llevar a cabo la multiprogramación, presentando a los niveles superiores tantas máquinas virtuales como se soliciten. Estas máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario [11].

Las características más notables de este sistema operativo se detallan a continuación:

- Compatibilidad

Al igual que un ordenador físico, una máquina virtual aloja su propio sistema operativo y aplicaciones guest, y dispone de los mismos componentes (placa base, tarjeta VGA, controlador de tarjeta de red, etc.). El resultado de ello es que las máquinas virtuales son totalmente compatibles con la totalidad de sistemas operativos x86, aplicaciones y controladores de dispositivos estándar, de modo de se puede utilizar una máquina virtual para ejecutar el mismo software que se puede ejecutar en un ordenador x86 físico.

- Aislamiento

Aunque las máquinas virtuales pueden compartir los recursos físicos de un único ordenador, permanecen completamente aisladas unas de otras, como si se tratara de máquinas independientes. Si, por ejemplo, hay cuatro máquinas virtuales en un único servidor físico y falla una de ellas, las otras tres siguen estando disponibles. El aislamiento es un factor importante que explica por qué la disponibilidad y protección de las aplicaciones que se ejecutan en un entorno virtual es muy superior a las aplicaciones que se ejecutan en un sistema tradicional no virtualizado.

- Encapsulamiento

Una máquina virtual es básicamente un contenedor de software que ata o “encapsula” un conjunto completo de recursos de hardware virtuales, así como un sistema operativo y todas sus aplicaciones, dentro de un paquete de software. El encapsulamiento hace a las máquinas virtuales extraordinariamente portables y fáciles de gestionar. Por ejemplo, puede mover y copiar una máquina virtual de un lugar a otro como lo haría con cualquier otro archivo de software, o guardar una máquina virtual en cualquier medio de almacenamiento de datos estándar, desde una memoria USB de tamaño de bolsillo hasta las redes de área de almacenamiento (SAN) de una empresa.

- Independencia del hardware

Las máquinas virtuales son completamente independientes de su hardware físico subyacente. Por ejemplo, se puede configurar una máquina virtual con componentes virtuales (CPU, tarjeta de red, controlador SCSI, pongamos por caso) que difieren totalmente de los componentes físicos presentes en el hardware subyacente. Las máquinas virtuales del mismo servidor físico pueden incluso ejecutar distintos tipos de sistema operativo (Windows, Linux, etc.).

Si se combina con las propiedades de encapsulamiento y compatibilidad, la independencia del hardware proporciona la libertad para mover una máquina virtual de un tipo de ordenador x86 a otro sin necesidad de efectuar ningún cambio en los controladores de dispositivo, en el sistema operativo o en las aplicaciones. La independencia del hardware también significa que se puede ejecutar una mezcla heterogénea de sistemas operativos y aplicaciones en un único ordenador físico [8].

2.1.4 Linux

Linux es un sistema operativo tipo Unix (también conocido como GNU/Linux) que se distribuye bajo la Licencia Pública General de GNU o GPL, es decir que es software libre. Su nombre proviene del Núcleo de Linux, desarrollado en 1991 por Linus Torvalds.

En abril de 1991, Linus Torvalds, de 21 años, empezó a trabajar en unas simples ideas para un núcleo de sistema operativo. Decidió crear un sistema basado en Unix para máquinas i386, es entonces que le nació la idea de la creación de Linux. La historia de Linux está fuertemente vinculada a la del proyecto GNU, iniciado en 1983, tiene como objetivo el desarrollo de un sistema Unix completo compuesto completamente de software libre. Para 1991, cuando la versión 1 del núcleo Linux fue liberada, el proyecto GNU había producido varios de los componentes del sistema operativo, incluyendo un intérprete de comandos, una biblioteca C y un compilador, pero aún no contaba con el núcleo que permitiría completar al sistema operativo [12].

El éxito de Linux se debe principalmente a su distribución en Internet, pues varios miles de programadores voluntarios de todo el mundo han participado en el proyecto mejorándolo continuamente. Torvalds y demás desarrolladores de los inicios de Linux, adaptaron los componentes de GNU y de BSD, así como de otros muchos proyectos como son: Perl, Apache y Python, para trabajar con el núcleo Linux, creando un sistema operativo completamente funcional procedente de fuentes distintas, donde la mayoría de estas fuentes son libres.

El Sistema Operativo Linux se define como "un sistema operativo para computadoras que facilita su uso y operación" y es usado ampliamente en servidores y súper-computadores y cuenta con el respaldo de corporaciones como Dell, Hewlett-Packard, IBM, Novell, Oracle, Red Hat y Sun Microsystems [12].

Puede ser instalado en gran variedad de hardware, incluyendo computadores de escritorio y portátiles (PCs x86 y x86-64 así como Macintosh y PowerPC), computadores de bolsillo, teléfonos celulares, dispositivos empotrados, videoconsolas (Xbox, PlayStation 3, PlayStation Portable, Dreamcast, GP2X...) y otros (como juegos arcade, enrutadores y algunos modelos de iPod).

Las variantes de estos sistemas se denominan "distribuciones" y su objetivo es ofrecer una edición que cumpla con las necesidades de determinado grupo de usuarios. De esta forma existen distribuciones para hogares, empresas y servidores. Algunas son gratuitas y otras de pago, algunas insertan software no libre y otras contienen solo software libre.

Hoy en día Linux cuenta con los principales gestores de ventanas, utilidades para internet, compiladores, editores, entre otros más componentes. Linux se utiliza con éxito como servidor de la mayoría de las empresas y universidades alrededor del mundo, y cada vez son más los usuarios particulares que se deciden por este Sistema Operativo.

2.1.4.1 Kernel / Shell

Los términos **Kernel** y **Shell** son constantemente utilizados cuando se habla de informática y de sistemas operativos, pero muchos usuarios desconocen el significado de estos mismos, relacionándolos equivocadamente tan sólo con los sistemas operativos Linux.

❖ Kernel

El Kernel, o núcleo, se refiere al Software que relaciona las aplicaciones con el Hardware de nuestro ordenador. Las órdenes del Kernel son las únicas que interactúan directamente con los componentes físicos del ordenador, como el procesador, la RAM o los discos duros, ya que están expresamente diseñadas para no dañar estos elementos. Las instrucciones del Kernel también son las encargadas de asignar los recursos a cada proceso y decidir que prioridades y recursos del sistema se le asignan.

Las funciones más importantes del mismo, son:

- Administración de la memoria para todos los programas y procesos en ejecución.
- Administración del tiempo de procesador que los programas y procesos en ejecución utilizan.
- Es el encargado de que podamos acceder a los periféricos/elementos de nuestro ordenador de una manera cómoda [13].

Estas funciones son de las más importantes, sin embargo el núcleo de Linux es tan completo que no las hace funciones únicas.

Desarrollo de la serie 2.6

Hasta que empezó el desarrollo de esta serie, existieron dos tipos de versiones del núcleo:

- Versión de producción: La versión de producción, era la versión estable hasta el momento. Esta versión era el resultado final de las versiones de desarrollo o experimentales.

Cuando el equipo de desarrollo del núcleo experimental, decidía que tenía un núcleo estable y con la suficiente calidad, se lanzaba una nueva versión de producción o estable. Esta versión era la que se debía utilizar para un uso normal del sistema, ya que eran las versiones consideradas más estables y libres de fallos en el momento de su lanzamiento.

- Versión de desarrollo: Esta versión era experimental y era la que utilizaban los desarrolladores para programar, comprobar y verificar nuevas características, correcciones, etc. Estos núcleos solían ser inestables y no se debían usar sin saber lo que se hacía.

Como interpretar los números de las versiones de las series por debajo de la 2.6:

- Las versiones del núcleo se numeraban con 3 números, de la siguiente forma: AA.BB.CC

AA: Indicaba la serie/versión principal del núcleo. Solo han existido la 1 y 2. Este número cambiaba cuando la manera de funcionamiento del kernel había sufrido un cambio muy importante.

BB: Indicaba si la versión era de desarrollo o de producción. Un número impar, significaba que era de desarrollo, uno par, que era de producción.

CC: Indicaba nuevas revisiones dentro de una versión, en las que lo único que se había modificado eran fallos de programación.

Con la serie 2.6 del núcleo, el sistema de numeración así como el modelo de desarrollo han cambiado. Las versiones han pasado a numerarse con 4 dígitos y no existen versiones de producción y desarrollo.

- Las versiones del núcleo se numeran hoy en día con 4 dígitos, de la siguiente forma: AA.BB.CC.DD.

AA: Indica la serie/versión principal del núcleo. BB: Indica la revisión principal del núcleo. Números pares e impares no tienen ningún significado hoy en día. CC: Indica nuevas

revisiones menores del núcleo. Cambia cuando nuevas características y drivers son soportados. DD: Este digito cambia cuando se corrigen fallos de programación o fallos de seguridad dentro de una revisión.

Hoy en día se suele usar el núcleo distribuido con la distribución que el usuario utiliza. Son las distribuciones las encargadas de distribuir núcleos estables a sus usuarios y estos núcleos se basan en el núcleo ("vanilla") distribuido por Linus Torvalds y el equipo de programadores del núcleo.

Es muy recomendable el utilizar los núcleos distribuidos por la distribución que utilizamos. Estos núcleos tienen muy buena calidad y son muy completos y con esto nos evitamos el trabajo de configurar, compilar y mantener nuevas versiones del núcleo.

❖ Shell

El Shell, (que significa "caparazón") o intérprete de órdenes, son aplicaciones capaces de interpretar las órdenes del usuario a través de comandos escritos, como por ejemplo el sistema MS-DOS o los terminales de consola de los sistemas operativos Linux. Estas aplicaciones permiten al usuario interactuar con el ordenador, normalmente a través de una sencilla interfaz de texto plano, y suponen la forma más básica de interacción de un usuario con su ordenador, escribiendo las órdenes en este Shell a través de comandos y recogiendo las respuestas de la máquina.

La shell es un archivo ejecutable que debe interpretar los comandos, transmitirlos al sistema y arrojar el resultado. Existen varios shells. La más común es **sh** (llamada "Bourne shell"), **bash** ("Bourne again shell"), **csh** ("C Shell"), **Tcsh** ("Tenex C shell"), **ksh** ("Korn shell") y **zsh** ("Zero shell"). Generalmente, sus nombres coinciden con el nombre del ejecutable [13].

Indicador del sistema

La shell se inicia al leer su configuración completa (en un archivo del directorio */etc/*) y después al leer la configuración propia del usuario (en un archivo oculto cuyo nombre comienza con un punto y que se ubica en el directorio básico del usuario, es decir */home/user_name/.configuration_file*).

De manera predeterminada, para la mayoría de las shells, el indicador consiste en el nombre del equipo, seguido de dos puntos (:), el directorio actual y después un carácter que indica el tipo de usuario conectado:

- "\$" especifica un usuario normal
- "#" especifica el administrador, llamado "root"

El concepto de la línea de comandos

Una línea de comandos es una cadena de caracteres formada por un comando que corresponde a un archivo ejecutable del sistema o, más bien, un comando de shell como así también argumentos opcionales (parámetros).

Entrada – Salida estándar

Una vez que se ejecuta un comando, se crea un proceso. Este proceso abre tres flujos:

- *stdin*, denominado **entrada estándar**, en cuyo caso el proceso lee los datos de entrada. De manera predeterminada, *stdin* se refiere al teclado. STDIN se identifica con el número 0.
- *stdout*, denominado **salida estándar**, en cuyo caso el proceso escribe los datos de salida. De manera predeterminada, *stdout* se refiere a la pantalla. STDOUT se identifica con el número 1.
- *stderr*, denominado **error estándar**, en cuyo caso el proceso escribe los mensajes del error. De manera predeterminada, *stderr* se refiere a la pantalla. STDERR se identifica con el número 2.

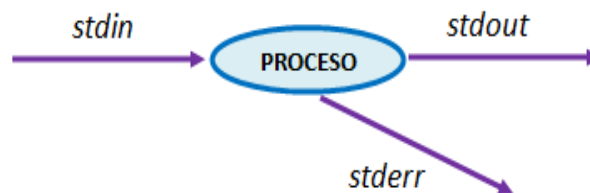


Figura 2.7 Flujo del Shell

Por lo tanto, de manera predeterminada, cada vez que se ejecuta un programa, los datos se leen desde el teclado y el programa envía su salida y sus errores a la pantalla. Sin embargo, también es posible leer datos desde cualquier dispositivo de entrada, incluso desde un archivo, y enviar la salida a un dispositivo de visualización, un archivo, etc.

Tuberías de comunicación

Las **tuberías** (en inglés "*pipes*") (literalmente "tuberías") son mecanismos de comunicación específicos para todos los sistemas UNIX. Una tubería, simbolizada por una barra vertical (carácter "|"), permite asignar la salida estándar de un comando a la entrada estándar de otro, de la misma forma en que una tubería permite la comunicación entre la entrada estándar de un comando y la salida estándar de otro [13].

De manera sencilla podemos resumir que las funciones centrales de un SO son controladas por el núcleo (kernel) mientras que la interfaz del usuario es controlada por el entorno (shell).

2.1.4.2 Características Linux

Dos características muy peculiares lo diferencian del resto de sistemas que podemos encontrar en el mercado, la primera, es que es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del código fuente.

Los beneficios derivados del uso del sistema operativo UNIX, y por lo tanto de Linux, provienen de su potencia y flexibilidad. Estos son resultado de numerosas características integradas al sistema, las que están disponibles tan pronto como se inicia.

- **Multitareas**

Multitareas describe la habilidad de ejecutar aparentemente, al mismo tiempo, numerosos programas sin obstaculizar la ejecución de cada aplicación. Esto se conoce como multitareas preferente, porque cada programa tiene garantizada la posibilidad de correr, esto es, cada programa no se ejecuta sino que hasta que el sistema operativo lo aparta para permitir que otros programas corran. Este tipo de multitareas es exactamente lo que Linux hace.

Para comprender mejor la capacidad de multitareas de Linux, se examinará desde otra perspectiva. El microprocesador de una computadora puede hacer una sola cosa a la vez, pero es capaz de completar esas tareas individuales en periodos tan breves que son difíciles de captar. La mente humana es incapaz de detectar la diferencia entre una demora tan breve y algo que ocurra simultáneamente. De tal forma que pareciera que las tareas se realizan al mismo tiempo.

Es fácil apreciar los beneficios de tener capacidades de multitareas preferente. Además de reducir los tiempos muertos (lapso en el que no puede proseguir su trabajo en una aplicación porque el proceso no termina aún), la flexibilidad de no tener que cerrar las ventanas de aplicaciones antes de abrir y trabajar en otras es mucho más conveniente.

- **Multiusuario**

La capacidad de Linux para asignar tiempo del microprocesador a numerosas aplicaciones simultáneas se prestó como consecuencia a servir a numerosas personas al mismo tiempo, cada una ejecutando una o más aplicaciones. La particularidad en realidad destacada de Linux y sus características de multiusuario y multitareas, es que más de una persona puede trabajar en la misma versión de la misma aplicación de manera simultánea, desde las mismas terminales o desde terminales separadas. Esto no debe confundirse con numerosos usuarios que actualizan un archivo a un tiempo, particularidad que es potencialmente desconcertante y peligrosa a la vez que indeseable.

- **Shells programables**

Esta es otra característica que hace de UNIX y en consecuencia de Linux, lo que es: el sistema operativo más flexible. Aunque en los últimos años se han agregado interfaces gráficas al sistema UNIX, casi todas las utilerías para emplear y administrar Linux, se ejecutan mediante la escritura de comandos. En Linux, al intérprete de la línea de comando de le conoce como shell, que no es otra cosa más que un programa diseñado para aceptar comandos y ejecutarlos. Varios tipos de programas pueden emplearse como shells, pero en casi todas las versiones de Linux existen diversos shells estándares disponibles.

Los shells de Linux, son equivalentes al COMMAND.COM que emplea MS-DOS. Ambos aceptan y ejecutan comandos, y corren archivos de procesamiento por lotes y programas.

- **Independencia de dispositivos bajo Linux**

Al igual que UNIX, Linux comparte muchos de los mismos beneficios de la independencia de dispositivos. Por desgracia, una de las mejores características de Linux es al mismo tiempo una de sus mayores desventajas: su independencia del mundo comercial. Linux ha sido desarrollado durante los últimos años por una variedad de programadores de todo el mundo, quienes no tienen acceso a todos los equipos creados para las PCs de IBM y las compatibles. De hecho, Linux no corre en algunas PCs de IBM, en especial las que utilizan el bus de microcanal.

Como último recurso, puesto que se tiene el código fuente para el núcleo, se puede modificar el núcleo para trabajar con los nuevos dispositivos.

- **Comunicaciones y capacidades de red**

La superioridad de UNIX y de Linux sobre otros sistemas operativos es igual de evidente en sus utilerías para comunicaciones y red. El envío de mensajes internos o la transferencia de archivos pueden hacerse mediante algunos comandos de Linux, entre los que están write, cal, mail, mailx así como cu y uucp. El intercambio de información entre usuarios del mismo sistema se lleva a cabo por medio de comunicación terminal a terminal, correo electrónico y un calendario automático que sirve como administrador y programador de información. Un sistema Unix/Linux permite no solamente trabajar con él en una máquina, sino también conectar varios ordenadores centrales Unix/Linux entre sí de tal forma que cada usuario tenga acceso a la información contenida en todos ellos. La conexión se realiza a través del conjunto de protocolos y servicios que ofrece TCP/IP. Gracias a él puedes ejecutar programas en máquinas Unix/Linux que estén a varios kilómetros de distancia entre sí; enviar correo electrónico de unos equipos a otros; realizar conversación directa entre dos usuarios, etcétera.

- **Portabilidad de sistemas abiertos**

En la interminable búsqueda de la estandarización, muchas organizaciones han asumido un interés renovado en la dirección en la que están evolucionando los sistemas operativos.

La portabilidad es la capacidad de transportar un sistema operativo de una plataforma a otra para que siga funcionando del mismo modo en que lo hacía. La portabilidad que proporciona los medios para que diferentes plataformas de cómputo que corren UNIX se comuniquen adecuada y efectivamente con cualquiera de las otras sin necesidad de agregar interfaces de comunicaciones especiales, costosas y de última hora. Ningún otro sistema operativo puede satisfacer esta demanda.

- **Algoritmo por prioridades o multinivel.**

Es uno de los más complejos y eficaces. Asigna los tiempos de ejecución de la UCP según una lista de prioridades. En cada una de estas listas, el sistema operativo incluirá aquellos procesos a los que se les haya asignado esa prioridad. El tiempo de ejecución del procesador se irá destinando, en primer lugar, de forma secuencial a los procesos de mayor nivel. Terminados éstos, se ejecutarán los procesos del nivel inferior, y así sucesivamente hasta los procesos del nivel más bajo.

- **Memoria virtual.**

Esta técnica permite a los usuarios del sistema ejecutar programas, de tal forma que dé la sensación de que toda la memoria RAM es para ellos. Concretamente en Unix/Linux se utiliza la paginación de la memoria. Esta técnica, que ya viste en la Unidad 2, es la que utilizan la mayoría de los sistemas operativos multiusuario. Dividen la memoria en páginas al igual que los programas y de esta forma se realiza el intercambio entre disco y RAM para ejecutar los mismos.

- **Sistema de archivos jerárquico.**

Utiliza, de forma similar al DOS, un sistema de archivos en forma de árbol invertido. La diferencia esencial frente al DOS es que el Unix/Linux no gestiona dispositivos (como una disquetera) de forma directa. El Unix/Linux gestiona los dispositivos como si fueran directorios, de tal forma que cuando estemos accediendo al directorio asociado a una disquetera, en realidad lo que estarás haciendo es acceder a la información contenida en el disquete.

- **Sistemas de seguridad.**

Es una de las características más importantes, ya que la información a la que un usuario puede tener acceso puede limitarse de forma sencilla. De este modo, el administrador del sistema operativo, a través de palabras clave (para archivos empaquetados o comprimidos) o mediante la asignación de derechos a los usuarios, hace que la información contenida en un servidor Unix/Linux esté totalmente protegida de piratas o usuarios no deseados.

- **Interfaz texto/gráfica.**

Lo normal es que la interfaz utilizada por Unix/Linux sea de tipo texto. Pero poco a poco se han ido incorporando mejoras que permiten gestionar el sistema operativo en modo gráfico. Fundamentalmente, la aparición de las interfaces X **WINDOWS** ha permitido agilizar y mejorar procesos, especialmente para el administrador [12].

2.1.5 Windows

Es el sistema operativo de mayor difusión entre computadoras personales y servidores pequeños y medianos. Fue desarrollado por Microsoft, aunque muchas de sus ideas básicas provinieron del sistema operativo Mac OS, de los equipos Macintosh.

Las primeras versiones del Sistema Operativo de Microsoft recibieron la denominación de **DOS** (Disk Operating System – Sistema Operativo de Disco), software que trabajaba en ambiente de texto, es decir, que las instrucciones o comandos debían ser escritas desde teclado [14].

La primera versión de Windows fue la 1.0, lanzada en noviembre de 1985, carecía de funcionalidades y consiguió un poco de popularidad. No era un sistema operativo en sí mismo, sino que era un programa ejecutándose en el sistema **MS-DOS** [15].

En noviembre de 1987 fue lanzada la versión 2.0 de Windows, y fue un poco más popular que su predecesora. La versión 2.03, lanzada en enero de 1988, cambió su interfaz de ventanas, haciéndose muy parecido al sistema operativo de las Apple, pues la información se presentaba dentro de espacios rectangulares de la pantalla, con bordes, a modo de ventanas (le trajo a Microsoft problemas legales).

Microsoft Windows 3.0 fue lanzada en 1990, fue la primera versión en alcanzar éxito comercial; vendió dos millones de copias en seis meses. Tuvo mejoras en la interfaz de usuario y las capacidades de multitarea.

El 1 de marzo de 1992 salió la primera versión gráfica exitosa del Sistema Operativo de Microsoft, **Windows 3.1** para equipos *stand alone* (aislados), que en realidad era un ambiente gráfico que trabajaba sobre DOS (era una interfaz gráfica para DOS).

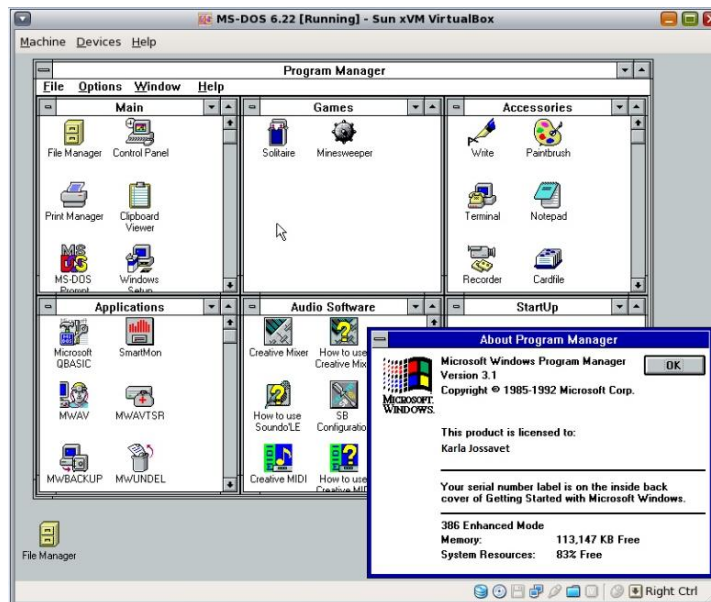


Figura 2.7 Windows MSDOS

Posteriormente apareció una versión mejorada de Windows, identificada como **3.11**, que básicamente tenía la misma funcionalidad que Windows 3.1, pero permitía trabajar al equipo en esquemas básicos de red, por lo que la identificación formal utilizada para describirlo fue **Windows para Trabajo en Grupo**.

En julio de 1993, Microsoft lanzó **Windows NT** basado en un nuevo kernel. NT es considerado de la línea profesional de los sistemas operativos Windows. La línea de hogar y la línea profesional fueron fusionadas años más tarde con la llegada de Windows XP.

Para agosto de 1995, Microsoft lanza **Windows 95**, que es considerado el primer Sistema Operativo Gráfico de Microsoft para estaciones de trabajo (clientes dentro de una red). De todas

maneras seguía dependiendo del MS-DOS, por lo que muchos expertos no lo consideran todavía un sistema operativo. Windows 95 cambió completamente su interfaz y se hizo más fácil de usar. Además fue el primer sistema operativo de 32 bits para clientes tipo PC.

Luego, en junio de 1998, se lanzó Microsoft **Windows 98**, también con gran popularidad, pero con varios problemas de seguridad que necesitaron una **Second Edition** en 1999 para resolverlos.

En febrero de 2000, se lanzó **Windows 2000**, perteneciente a la línea profesional sistemas de Microsoft.

En tanto en la línea de hogar a **Windows 98** le siguió **Windows ME (Millennium Edition)**, lanzado en septiembre de 2000. Fue una de las versiones más criticadas del Windows por sus debilidades en la estabilidad y la compatibilidad.

En octubre de 2001, Microsoft lanza **Windows XP**, la cual está basada en el kernel de Windows NT, pero que también incorpora características de la línea de hogar. Esta versión fue sumamente elogiada en revistas de computación, aunque de todas maneras necesitó dos **Service Pack** de actualización para lograr una seguridad robusta.

En abril de 2003, fue lanzado **Windows Server 2003** en reemplazo de la línea de productos para servidores Windows 2000. Esta versión poseía muchas mejoras y una fuerte seguridad. Fue seguido de **Windows 2003 R2** en diciembre de 2005.

La siguiente versión del sistema operativo fue **Windows Vista**, lanzada el 30 de noviembre de 2006 para clientes de negocios. En cambio la versión para consumidores fue lanzada el 30 de enero de 2007. Windows Vista trajo mejoras en la seguridad, características más avanzadas en sus gráficos con una interfaz opcional llamada Windows Aero, además de múltiples nuevas aplicaciones.

Windows 7 es la versión más reciente de Microsoft Windows, línea de sistemas operativos producida por Microsoft Corporation. Esta versión está diseñada para uso en PC, incluyendo equipos de escritorio en hogares y oficinas, equipos portátiles, tablet PC, netbooks y equipos media center. El desarrollo de Windows 7 se completó el 22 de julio de 2009, siendo entonces confirmada su fecha de venta oficial para el 22 de octubre de 2009 junto a su equivalente para servidores **Windows Server 2008 R2.3**

A diferencia del gran salto arquitectónico y de características que sufrió su antecesor Windows Vista con respecto a Windows XP, Windows 7 fue concebido como una actualización incremental y focalizada de Vista y su núcleo NT 6.0, lo que permitió mantener cierto grado de compatibilidad con aplicaciones y hardware en los que éste ya era compatible. Sin embargo, entre las metas de desarrollo para Windows 7 se dio importancia a mejorar su interfaz para volverla más accesible al usuario e incluir nuevas características que permitieran hacer tareas de una manera más fácil y rápida, al mismo tiempo que se realizarían esfuerzos para lograr un sistema más ligero, estable y rápido.

Diversas presentaciones ofrecidas por la compañía en 2008 se enfocaron en demostrar capacidades multitáctiles, una interfaz rediseñada junto con una nueva barra de tareas y un sistema de redes domésticas simplificado y fácil de usar denominado «Grupo en el hogar», además de importantes mejoras en el rendimiento general del sistema operativo

Windows 8, llegará este 2012. Está construido en base a Windows 7 y se adaptará a las redes sociales y al emergente mercado de las tablets.

Tendrá dos “escritorios”, uno tradicional y otro adaptado para **tablets**. La nueva interfaz específica para tablets (**Metro UI**) es muy sencilla y cómoda. Windows 8 incorporará en la parte superior del explorador de archivos la interfaz *Ribbon*, que es una banda (cinta) donde se exponen todas las funciones que puede realizar un programa en un solo lugar, es una barra de herramientas como la que incluye Office desde la versión 2007. Esta interfaz nos permitirá acceder fácilmente a las principales opciones de administración de archivos.

- **Transferencia de archivos**

Hasta Windows 7 cada operación de transferencia de archivos se visualizaba en una ventana separada. En Windows 8 se mostrará todo en una sola ventana con información detallada y con la posibilidad de pausar la transferencia.

Éste será el primer sistema operativo de Microsoft que soportará procesadores basados en ARM. Estos procesadores tienen una gran capacidad de procesamiento y consumen muy poca energía, es por lo que dominan el mercado de los smartphones y tablets.

Windows App Store

Microsoft incluirá por primera vez en Windows una tienda de aplicaciones. Una captura filtrada hace unos meses nos puede dar una idea de cómo será esta “App Store”:

Actualmente se encuentran vigentes **Windows Vista** y **Windows Se7en**.

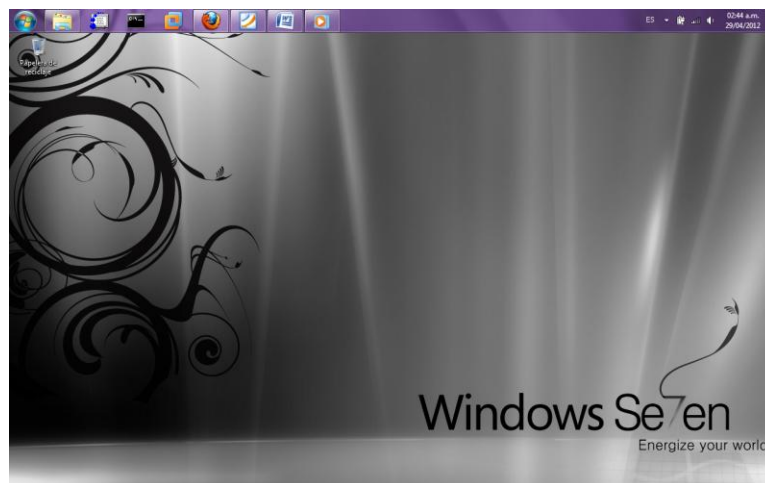


Figura 2.8 Windows Se7ven

Las versiones correspondientes de sistema operativo para servidores son:

Windows NT Server

Sistema operativo para servidores que internamente posee una arquitectura de 32 bits, ampliable e independiente de la plataforma. Puede ejecutarse en sistemas basados en procesadores Intel x86, RISC y DEC Alpha, ofreciendo al usuario mayor libertad a la hora de elegir sus sistemas informáticos. Es ampliable a sistemas de multiproceso simétrico, lo que permite incorporar procesadores adicionales cuando se desee aumentar el rendimiento, posee múltiples threads (subprocesos) de ejecución, lo que permite utilizar aplicaciones más potentes.

Windows 2000 Server

Este sistema operativo está construido sobre la fortaleza de la tecnología Windows NT, integrando servicios de directorio, de Web, de aplicaciones, de comunicaciones, de administración de archivos e impresión basados en estándares, con alta confiabilidad, administración eficiente y soporte para los últimos avances del hardware de redes, para proporcionar la mejor base para integrar su negocio con Internet. Se incrementó su capacidad para manejar hasta 32 procesadores y soportar hasta 64 GB de memoria.

Windows XP Server / Windows 2003 Server

Windows Server 2003 está basado en tecnología NT y su versión del núcleo NT es la 5.2. En términos generales, Windows Server 2003 se podría considerar como un Windows XP modificado para labores empresariales, no con menos funciones, sino que estas están deshabilitadas por defecto para obtener un mejor rendimiento y para centrar el uso de procesador en las características de servidor; por ejemplo, la interfaz gráfica denominada Luna de Windows XP viene desactivada por lo que sólo se utiliza la interfaz clásica de Windows.

Windows Server 2003 R2

Este servidor facilita la extensión de conectividad y control a identidades, ubicaciones, datos y aplicaciones. Brinda una plataforma Web escalable y más segura, proporciona una forma más eficiente de administrar y controlar el acceso a recursos locales y remotos, e integrándose, a la vez, en su entorno Windows Server 2003 existente con suma facilidad.

Windows Server 2008

Windows Server 2008 está diseñado para ofrecer a las organizaciones la plataforma más productiva para virtualización de cargas de trabajo, creación de aplicaciones eficaces y protección de redes. Ofrece una plataforma segura y de fácil administración, para el desarrollo y alojamiento confiable de aplicaciones y servicios web. La nueva consola del Administrador del servidor proporciona un único origen para administrar la configuración del servidor y la información del sistema. El sistema de protección de servicios de Windows ayuda a mantener más seguros los sistemas al evitar que los servicios críticos de servidor estén en riesgo por actividades anormales

en el sistema de archivos, registro, o red. La seguridad también se mejora en el sistema operativo Windows Server 2008 por medio de protección de acceso a redes (NAP), controlador de dominio de sólo lectura (RODC), mejoras en la infraestructura de clave pública (PKI), un nuevo firewall de Windows bidireccional y compatibilidad con criptografía de última generación.

Windows Server 2008 R2

Este servidor está diseñado para ayudar a incrementar el control, la disponibilidad y la flexibilidad de su infraestructura del centro de datos y del escritorio al mismo tiempo que le ayuda a reducir costos. Brinda nuevas características de virtualización incluyendo Live Migration y Cluster Shared Volumes, un reducido consumo de energía, un nuevo conjunto de herramientas de administración, nuevas características Active Directory como una "papelera de reciclaje" para objetos AD borrados, una nueva versión de IIS (7.5) que incluye un renovado servidor FTP, soporte para DNSSEC y el aumento del número de núcleos de procesamiento de 64 a 256. Los procesadores de 32-bits ya no están soportados.

La versión para dispositivos móviles es **Windows Phone 7** [15].

Cada nueva versión del Sistema Operativo presenta más funcionalidades que la versión previa, y generalmente es compatible con la versión anterior (las aplicaciones que funcionaban en la versión antigua también funcionan en la versión nueva).

2.1.5.1 Características del Sistema Operativo Windows

Lanzado en 1985. Tomó un total de 55 programadores para desarrollarlo y no permitía ventanas en cascada.

Microsoft comenzó el desarrollo del "ADMINISTRADOR DE INTERFAZ", que posteriormente derivó en Microsoft Windows en Septiembre de 1981. La interfaz inicial tenía menús ubicados en la parte inferior de la ventana y la interfaz sufrió un cambio en 1982 cuando se diseñaron los ahora comunes menús desplegables.

- **Interfaz gráfica**

Ambiente agradable para el usuario.

Interfaz gráfica con menús desplegables, ventanas en cascada y soporte para mouse.

Gráficos de pantalla e impresora independientes del dispositivo.

Ventanas traslapadas.

- **Multitarea**

Multitarea cooperativa entre las aplicaciones Windows.

- **Almacenamiento**

Soporte para FAT16, FAT32 y NTFS.
Encriptación de ficheros (EFS).
Servicio de indexación.
Sistema de archivos distribuido (DFS).
Nuevo sistema de backup (ASR).
Sistema de tolerancia a fallos (RAID) con discos dinámicos (software).

- **Comunicación**

Servicios de acceso remoto (RAS, VPN, RADIUS y Enrutamiento).
Nueva versión de IIS con soporte para HTTP/1.1.
Directorio Activo.
Balanceo de carga (clustering)
Servicios de instalación desatendida por red (RIS).
Servicios nativos de Terminal Server.

- **Marketing**

La base para que cualquier proyecto tenga éxito es haciendo un buen marketing, y estoy confiado de que Linux puede disfrutar de la comercialización que goza Windows, aun así, Microsoft posee un alto nivel económico como para realizar mejores campañas publicitarias, en conjunción con otros gigantes tecnológicos.

- **Soporte de hardware**

Se puede decir que Linux ya posee un gran soporte de hardware, pero todavía hay áreas demasiado pobres, como por ejemplo la inalámbrica. Aunque la lista de chips inalámbricos compatibles es cada vez más grande, todavía existe un largo camino para batir este objetivo. Podría haber nuevos usuarios en Linux si cuando han instalado una distribución de Linux les hubiera detectado todo dispositivo hardware de su máquina. Sí, algunos de ellos se molestan en buscar un cierto tiempo por Internet hasta que consiguen una solución, pero otros llegan a desistir y vuelven a Windows.

- **Sincronización con el móvil**

Independientemente del teléfono móvil, uno de los mayores beneficios de uso es que se puede sincronizar con su PC. Al menos en Windows. Muchos teléfonos móviles se convierten rápidamente paralizados cuando se conectan a un ordenador con Linux.

- **Presencia de empresa**

En muchos niveles, Linux es una combinación perfecta para SMB y el uso de la empresa, ya sea en escritorio o en servidor, pero eso no tiene y no puede ocurrir sin un cambio real. No estoy seguro qué cambio es, pero creo que la mayor parte es en el extremo de la empresa, y todos sabemos

que no va a suceder. Pero si Linux puede disfrutar de la presencia que Windows tiene en la empresa, todo el paisaje de TI cambiaría.

- **Configuración de grupo de trabajo**

Se puede conseguir con Samba con cierta rapidez, pero eso la mayoría de veces supone tener una buena experiencia en Linux, mientras que con Windows es una tarea más fácil.

- **Pantalla táctil**

Una de las grandes cosas que trae Windows 7 es el soporte mejorado con las pantallas táctiles. Con Linux, el número de pantallas táctiles son muy limitadas, pero incluso las que se encuentran ese número limitado, requiere una configuración delicada (xorg.conf). Por el momento, el uso de las pantallas táctiles no son muy frecuentes, pero quién sabe si en el futuro la cosa cambia.

- **Soporte**

Si tienes un problema con Windows, puedes llamar al soporte técnico de tecnología de Microsoft.

- **Direct X**

Los juegos son la razón por la que mucha gente no migra a Linux. Hay un montón de jugadores en todo el mundo, y hasta que Direct X exista en Linux, los jugadores no encontrarán un camino viable para el cambio de otro sistema operativo que no sea compatible con Direct X [15].

2.2 Lenguajes de Programación

Los lenguajes de programación son herramientas que nos permiten crear software. Pueden ser de alto o bajo nivel. En los lenguajes de programación de bajo nivel, las instrucciones son simples y cercanas al funcionamiento de la máquina, como por ejemplo el código máquina y el ensamblador. En los lenguajes de alto nivel hay un alto grado de abstracción y el lenguaje es más próximo a los humanos, como por ejemplo Léxico, PASCAL, Cobol o Java [16].



❖ Lenguaje de Bajo Nivel

Se llaman de bajo nivel porque están muy cercanos al hardware del ordenador. Es necesario conocer a fondo la arquitectura de la máquina para la que se va a programar.

❖ Lenguaje de Nivel Medio

Suelen ser clasificados muchas veces de alto nivel, pero permiten ciertos manejos de bajo nivel. Son precisos para ciertas aplicaciones como la creación de sistemas operativos, ya que permiten un manejo abstracto (independiente de la máquina, a diferencia del ensamblador), pero sin perder mucho del poder y eficiencia que tienen los lenguajes de bajo nivel.

❖ Lenguaje de Alto Nivel

Su función principal radica en que a partir de su desarrollo, existe la posibilidad de que se pueda utilizar el mismo programa en distintas máquinas, es decir que es independiente de un hardware determinado. Es necesario contar con un traductor o compilador, que lo traduce al lenguaje específico de cada máquina.

La evolución de los lenguajes de programación se puede dividir en 5 generaciones:

Tabla 2.1 Generaciones de los Lenguajes de Programación

GENERACIÓN	NOMBRE	CARACTERÍSTICAS
Primera	De Máquina	Específico para cada microprocesador, uso de código binario.
Segunda	Ensamblador	Uso de nemotécnicos que abstraen del lenguaje máquina.
Tercera	De Procedimientos	Lenguajes estructurados con comandos cercanos al lenguaje común.
Cuarta	Orientados a Procesos	Programas orientados a problemas específicos.
Quinta	Natural	Incluye inteligencia artificial y sistemas expertos.

Se han desarrollado innumerables lenguajes de programación que buscan la mayor abstracción posible y facilitar la vida del programador, aumentando la productividad, encargándose los compiladores o intérpretes de traducir el lenguaje de alto nivel, al lenguaje de la computadora. Por este motivo resulta interesante y muy importante conocer los distintos lenguajes de programación que nos ayudan en el desarrollo, diseño y manipulación de un sistema de software.

Entre los lenguajes de programación más destacados podemos enlistar los siguientes:

❖ C / C++

C es un lenguaje de programación diseñado por Dennis Ritchie, de los Laboratorios Bell, y se instaló en un PDP-11 (fue la primera minicomputadora en interconectar todos los elementos del

sistema — procesador, memoria y periférico — a un único bus de comunicación, bidireccional, asíncrono) en 1972; se diseñó para ser el lenguaje de los Sistemas Operativos UNIX. Este lenguaje de programación combina características de un ensamblador (alto nivel y bajo nivel), es decir, contiene estructuras de programación de alto nivel, y la facilidad de usar librerías que también son características de alto nivel; sin embargo, fue diseñado con muy pocas instrucciones, las cuales son sumamente sencillas, fáciles de traducir al lenguaje de la máquina; y requiere de un entendimiento apropiado de cómo funciona la máquina, el uso de la memoria, etcétera.

❖ Pascal

Pascal es un lenguaje de programación de alto nivel de propósito general; esto es, se puede utilizar para escribir programas para fines científicos y comerciales.

El lenguaje de programación Pascal fue desarrollado por el profesor Niklaus (Nicolás) Wirth en Zurich, Suiza, al final de los años 1960s y principios de los 70s. Wirth diseñó este lenguaje para que fuese un buen primer lenguaje de programación para personas comenzando a aprender a programar. Pascal tiene un número relativamente pequeño de conceptos para aprender y dominar. Su diseño facilita escribir programas usando un estilo que está generalmente aceptado como práctica estándar de programación buena. Otra de las metas del diseño de Wirth era la implementación fácil. Él diseñó un lenguaje para el cual fuese fácil escribir un compilador para un nuevo tipo de computadora.

❖ Java

Fue diseñado por la compañía Sun Microsystems Inc, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas (redes de computadoras formadas por más de un tipo de computadora, ya sean PC, MAC's, estaciones de trabajo, etc.), y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma. El lenguaje fue diseñado con las siguientes características en mente:

- Simple. Elimina la complejidad de los lenguajes como "C" y da paso al contexto de los lenguajes modernos orientados a objetos. Orientado a Objetos. La filosofía de programación orientada a objetos es diferente a la programación convencional.
- Familiar. Como la mayoría de los programadores están acostumbrados a programar en C o en C++, la sintaxis de Java es muy similar al de estos.
- Robusto. El sistema de Java maneja la memoria de la computadora por ti. No te tienes que preocupar por apuntadores, memoria que no se esté utilizando, etc. Java realiza todo esto sin necesidad de que uno se lo indique.
- Seguro. El sistema de Java tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no puede hacer con los recursos críticos de una computadora.
- Portable. Como el código compilado de Java (conocido como byte code) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java.
- Independiente a la arquitectura. Al compilar un programa en Java, el código resultante un tipo de código binario conocido como byte code. Este código es interpretado por

diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida.

- Multithreaded. Un lenguaje que soporta múltiples threads es un lenguaje que puede ejecutar diferentes líneas de código al mismo tiempo.
- Interpretado. Java corre en máquina virtual, por lo tanto es interpretado.
- Dinámico. Java no requiere que compile todas las clases de un programa para que este funcione. Si realizas una modificación a una clase Java se encarga de realizar un Dynamic Binding o un Dynamic Loading para encontrar las clases.

❖ Perl

Es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado bourne shell (sh) y, en un grado inferior, de muchos otros lenguajes de programación. Es un lenguaje imperativo, con variables, expresiones, asignaciones, bloques de código delimitados por llaves, estructuras de control y subrutinas. Estructuralmente, Perl está basado en un estilo de bloques como los del C, y fue ampliamente adoptado por su destreza en el procesador de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas, entre los cuales podemos destacar:

❖ JavaScript

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript.

Ventajas

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.

Desventajas

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

❖ **Active Server Pages (ASP)**

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitio web dinámicos. Fue liberado por Microsoft en 1996. Las páginas web desarrolladas bajo este lenguaje es necesario tener instalado Internet Information Server (IIS).

ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl and Jscript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

Ventajas

- Usa Visual Basic Script, siendo fácil para los usuarios.
- Comunicación óptima con SQL Server.
- Soporta el lenguaje JScript (JavaScript de Microsoft).

Desventajas

- Código desorganizado.
- Se necesita escribir mucho código para realizar funciones sencillas.
- Tecnología propietaria.
- Hospedaje de sitios web costosos.

❖ **Java Server Pages (JSP)**

Es un lenguaje para la creación de sitios web dinámicos. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor. JSP fue desarrollado por Sun Microsystems, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Ventajas

- Ejecución rápida del servlets.
- Crear páginas del lado del servidor.
- Multiplataforma.
- Código bien estructurado.
- Integridad con los módulos de Java.
- La parte dinámica está escrita en Java.
- Permite la utilización se servlets.

Desventajas

- Complejidad de aprendizaje.

❖ Hyper Text Markup Language (HTML)

Básicamente, el HTML consta de una serie de órdenes o directivas, que indican al visor que estemos utilizando, la forma de representar los elementos (texto, gráficos, etc...) que contenga el documento. A esta forma le llamamos *directivas*.

Las directivas de HTML pueden ser de dos tipos, cerradas o abiertas.

Las **directivas cerradas** son aquellas que tienen una palabra clave que indica el principio de la directiva y otra que indica el final. Entre la directiva inicial y la final se pueden encontrar otras directivas. Las **directivas abiertas** constan de una sola palabra clave. Para diferenciar las directivas del resto del texto del documento se encierran entre los símbolos < y >. Las directivas cerradas incluyen el carácter / antes de la palabra clave para indicar el final de la misma. Una directiva puede contener "*parámetros*".

Directiva cerrada <title> UNAM Ahorro de Energía </title>

Directiva abierta <HR>

Directiva con parámetros <BODY background-image: url (unam_mosaico.png) ;"> </BODY>

Criterios para escribir la sintaxis de las directivas de HTML:

- 1) Las directivas se indican en letra mayúscula y en negrilla.
- 2) Los parámetros de las directivas se indican en letra minúscula y negrilla.
- 3) El resto de elementos se indican en letra normal.
- 4) Las palabras a resaltar en el texto se indican en cursiva y negrilla.

Ventajas

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

Desventajas

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en "basura" y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

❖ Hyper Text Preprocessor (PHP)

PHP es un lenguaje de programación interpretado embebido en las páginas HTML. El principal objetivo del lenguaje es permitir a los desarrolladores de aplicaciones basadas en el web escribir

páginas que se generan de forma dinámica de una forma sencilla y rápida. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida, lo cual aumenta la funcionalidad del servidor, convirtiéndolo en un sistema de desarrollo de aplicaciones cliente/servidor mucho más completo. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. A pesar de que es usado principalmente en interpretación del lado del servidor (server-side scripting), actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Con PHP se puede hacer cualquier procedimiento con un script, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Un sitio con páginas dinámicas es el que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para requisitos particulares. Las aplicaciones dinámicas para la Web son frecuentes en los sitios comerciales, donde el contenido visualizando se genera de la información alcanzada en una base de datos u otra fuente externa.

Una de sus características más importantes es su soporte para gran cantidad de bases de datos, entre las cuales pueden mencionarse MySQL, PostgreSQL, Oracle, ODBC, DB", Microsoft SQL Server, Firebird, SQLite, entre otras.

PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier aplicación, desde generar documentos en pdf hasta analizar código XML.

El código PHP va incrustado dentro del código HTML. Sus tags van incluidos dentro de `<? y ¿>`. Este lenguaje ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Como producto de código abierto PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos del funcionamiento se encuentren y se reparen rápidamente.

PHP es utilizado en aplicaciones Web-relacionadas por algunas de las organizaciones más prominentes tales como Mitsubichi, Redhat, Der Spiegel, MP3-Lycos, Ericsson, NASA, etc.

PHP es la opción natural para los programadores en máquinas con Linux que ejecutan servidores web con Apache, pero funciona bien en cualquier otra plataforma de UNIX o de Windows, Netscape o del web server de Microsoft. PHP también utiliza las sesiones de HTTP, conectividad de Java, expresiones regulares, LDAP, SNMO, IMAP, protocolos de COM (bajo Windows).

Ventajas

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.

- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

2.3 Recomendaciones del Modelo OSI

La Organización Internacional de Estandarización (ISO: International Estándar Organization) desarrolló un modelo de referencia para las arquitecturas de sistemas. Le llamó OSI: Open System Interconnection (Interconexión de Sistemas Abiertos).

En el concepto de OSI, un sistema es un conjunto de una o más computadoras, el software asociado, los periféricos, las terminales, los operadores humanos, los procesos físicos, los medios de transferencia de información, etc., que forman un conjunto independiente con capacidad de realizar el procesamiento de la información [5].

OSI pone atención al intercambio de información entre sistemas y no al funcionamiento interno de cada sistema en particular. En otras palabras, el Modelo de Referencia OSI constituye el marco de trabajo para el desarrollo de protocolos estándares para la comunicación entre dos capas homónimas ubicadas en equipos separados.

El objetivo, a largo plazo, de ISO es desarrollar una compatibilidad total inter-sistemas, entre los muchos productos y servicios ofrecidos por los proveedores y las redes transportadoras alrededor del mundo.

Este modelo es estratificado y se estructura en 7 capas:

1: Capa de control de Interconexión Física

Provee las características mecánicas, eléctricas, funcionales y de procedimiento, necesarias para establecer, mantener y liberar conexiones físicas entre el dispositivo terminal (DTE) y el punto de conexión de red (DCE), o entre dos DTE's.

2: Capa de control de Enlace de Datos (DLC)

La capa DLC provee la conexión lógica a través de la línea, el direccionamiento, el secuenciamiento y la recuperación de errores. Existe una dirección de enlace que identifica una conexión de enlace en la capa DLC.

Por lo tanto, juntando las funciones de las capas 1 y 2, ya tenemos la forma de conectar físicamente dos nodos adyacentes y de transferir un mensaje de datos entre ello, manejando direccionamiento, control de errores, etc.

3: Capa de control de Red

Esta capa provee el control entre dos nodos adyacentes. Dos conexiones se proveen: punto a punto o en red. Una o más conexiones de red pueden ser ubicadas en la misma conexión de enlace y se distinguen por sus direcciones. Las funciones proporcionadas por este estrato incluyen el ruteo de los mensajes, las notificaciones de errores y opcionalmente la segmentación y el bloqueo. Más que proveedora esta capa ayuda a la transferencia de datos entre estos puntos.

4: Capa de control de Transporte

Proporciona en control entre nodos de usuarios a través de la red. Cada nodo de la red debe enviar el mensaje hacia un punto perteneciente a la ruta más conveniente para llegar al destino final. Los criterios de selección de una ruta dependen de diversos factores (existencia, ocupación, costo, etc.).

Las capas 1 a 4 de OSI, conforman el Subsistema de Transporte. La 4 releva las sesiones, de cualquier consideración de detalle referente a la forma en la cual se realiza la transferencia de datos.

5: Capa de control de Sesión

Provee el soporte para interacciones entre entidades que cooperan en la capa de Presentación. Las funciones de la capa de sesión se dividen en dos categorías:

- Determinación y cancelación de contrato entre dos entidades de la capa de Presentación (Servicio de Administración de Sesión)

- Control del intercambio de datos, entre esas dos entidades, comprendiendo sincronización delimitación y recuperación de operaciones con los datos (Servicios de Diálogo de Sesión)

6: Capa de servicios de Presentación

Proporciona un conjunto de servicios de conversión y descifrado que la capa de Aplicación puede seleccionar, para poder interpretar el significado de los datos intercambiados.

7: Capa de Aplicación

Todas las otras capas existen en función de brindar soporte a ésta. Una aplicación se compone de procesos cooperantes que se intercomunican mediante el uso de los protocolos definidos en esta capa. Estos procesos de aplicación son la fuente y el destino último de los datos intercambiados.

Podemos resumir brevemente los principios que se aplicaron para llegar a las 7 capas del modelo OSI en la siguiente figura:

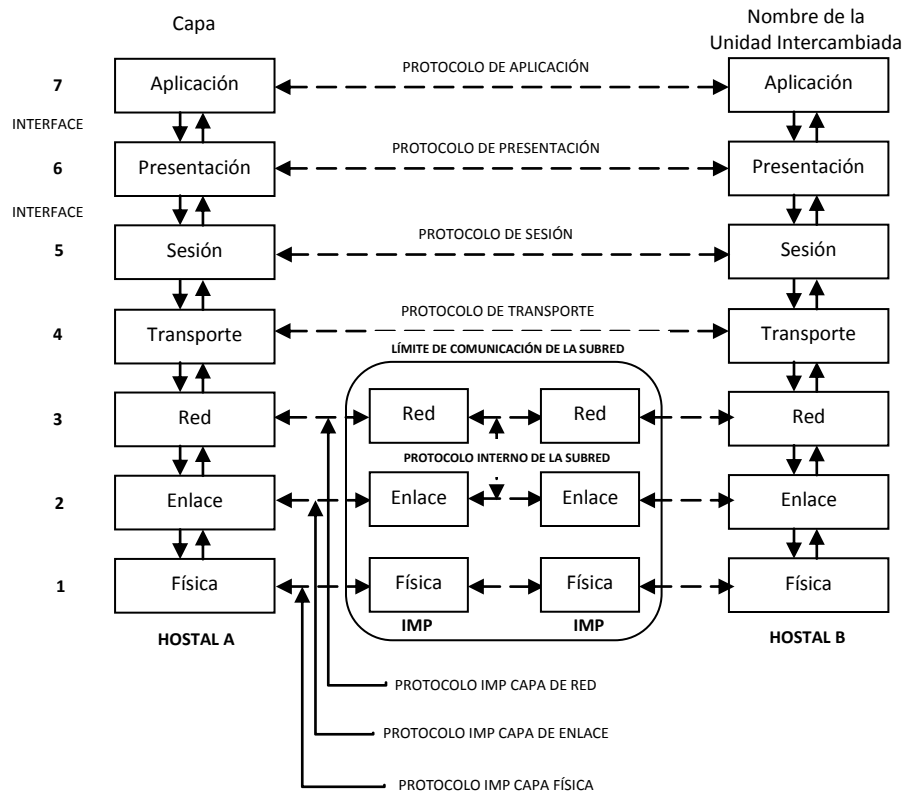


Figura 2.9 Modelo de Referencia OSI

2.3.1 Servicios de Seguridad

La tecnología de seguridad es importante para la imposición efectiva de políticas de seguridad en cualquier sistema informático.

Tal tecnología es especialmente importante en un entorno de red altamente distribuido, donde existen límites para los controles físicos y administrativos.

El estándar Internacional 7498-2 NIST identifica 5 servicios principales de seguridad.

El propósito de este estándar consiste en especificar los aspectos de seguridad del modelo de redes de computadoras de Interconexión de Sistemas Abiertos (OSI) [5]. A continuación mencionamos los 5 servicios de seguridad:

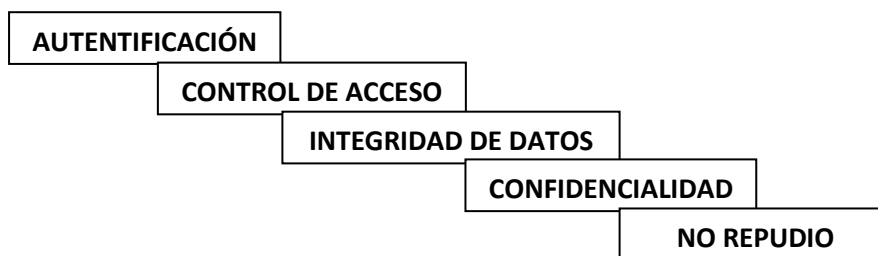


Figura 2.10 Servicios de seguridad

Un gran número de servicios auxiliares (auditorias, aseguración de disponibilidad) y de servicios de apoyo (Administración de llaves, mantenimiento de la seguridad, Administración de red) deberían aumentar esos servicios principales de seguridad.

Un sistema de seguridad integrado de un emplazamiento de Web debe ofrecer todos esos servicios con un gran número de mecanismos de seguridad implementados en una variedad de productos de seguridad.

La tecnología avanzará y proveerá de productos más nuevos, económicos, y de mejor calidad.

2.3.2 Autenticación

Es la verificación de la identidad indicada por el usuario de una computadora o red de computadoras.

Las operaciones de autenticación de usuarios permiten controlar el conjunto de páginas que puede ver un visitante del sitio en función de sus *privilegios* o *nivel de acceso*. El nombre de usuario (*User*), la contraseña (*password*) y los privilegios permiten determinar que páginas puede o no puede ver un usuario.

Como las contraseñas reutilizables son el eslabón más débil de internet del presente, los diseñadores deben diseñar mejores y más avanzadas técnicas de autenticación. Existe un espectro de soluciones, desde las contraseñas de una vez a sistemas de identificación biométricos de alta tecnología.

La autenticación basada en los *token* y sistemas de control de acceso parecen ser un compromiso razonable entre las metas de alta seguridad a bajo costo y de simplicidad de emplazamiento de red [5].

2.3.3 Disponibilidad y Control de Acceso

El control de acceso es la verificación e imposición del acceso autorizado del usuario a una red de computadoras subsecuentes a la autenticación de ese usuario.

La disponibilidad de información es el aseguramiento de que los usuarios autorizados tendrán acceso cuando requieran la información y sus activos asociados.

En el caso de los sistemas informáticos utilizados para almacenar y procesar la información, los controles de seguridad se utilizan para protegerlo, y los canales de comunicación protegidos que se utilizan para acceder a ella deben estar funcionando correctamente. La alta disponibilidad en sistemas tiene como objetivo seguir estando disponible en todo momento, evitando interrupciones del servicio debido a cortes de energía, fallos de hardware, y actualizaciones del sistema.

Garantizar la disponibilidad implica también la prevención de ataque de negación de servicio. Para poder manejar con mayor facilidad la seguridad de la información, las empresas o negocios se pueden ayudar con un sistema de administración que permita conocer y minimizar los posibles riesgos que atenten contra la seguridad informativa del negocio.

La disponibilidad además de ser importante en el proceso de seguridad de la información, es variada en el sentido de que existen varios mecanismos para cumplir con los niveles de servicio que se requiera, tales mecanismos se implementan en infraestructura tecnológica, servidores de correo electrónico, de bases de datos, de web, etc., mediante el uso de clusters o arreglos de discos, equipos en alta disponibilidad a nivel de red, servidores espejo, replicación de datos, redes de almacenamiento (SAN), enlaces redundantes, etc. La gama de posibilidades dependerá de lo que queremos proteger y el nivel de servicio que se quiera proporcionar [5].

2.3.4 Integridad de datos

Es la verificación de que los contenidos de un elemento de datos (mensaje, fichero, programa) no han sido cambiados intencionalmente o accidentalmente de manera no autorizada.

La profesión de la seguridad reconoce la intrusión como el riesgo más significativo para las redes. La tendencia corporativa hacia sistemas de red y arquitecturas abiertas ha excedido la capacidad de la mayoría de las salvaguardas y mecanismos de prueba de proteger adecuadamente los recursos de la información.

Asegurar la integridad de la información en los entornos de sistemas cooperativos actuales es bastante desafiante. La diversidad de plataformas, redes, protocolos y salvaguardas disponibles es abrumante. No existe una salvaguarda o solución universal para autenticar usuarios u transmisiones, ni tampoco existe una para proteger la información contra la destrucción, borrado, omisión, corrupción, compromiso deliberado o accidental. Es por eso que se hace el mayor esfuerzo para brindar al usuario la seguridad de que sus datos serán íntegros a la hora de acceder al sistema.

Para eso se realizan las pruebas de penetración a la información. Las pruebas de penetración son la estrategia más efectiva para evaluar tanto la competencia como las necesidades de una red computacional o una aplicación web. Una prueba de penetración simula un ataque desde afuera y proporciona un riguroso examen de vulnerabilidades actuales y potenciales, incluyendo fallas de software y hardware, configuraciones defectuosas del sistema, medidas de protección insuficientes, y más. El “ataque” se lleva a cabo con sumo cuidado, y numerosas salvaguardas son desplegadas para garantizar que la red o aplicación no sufra ningún daño durante las pruebas. El propósito de dichas pruebas se puede dividir en dos sub-servicios:

- a) Efectuar un análisis de riesgos a la aplicación (modelo de amenazas y vulnerabilidades)
- b) Efectuar pruebas de penetración a la aplicación. Planes de mitigación de riesgos [5].

2.3.5 Confidencialidad

Es la protección contra la revelación no autorizada de la información contenida en los datos. La propiedad de prevenir la divulgación de información a personas o sistemas no autorizados. A groso modo, la confidencialidad es el acceso a la información únicamente por personas que cuenten con la debida autorización.

Una transacción en Internet requiere de un identificador de usuario para poder ser transmitida a una red de procesamiento de transacciones. El sistema intenta hacer valer la confidencialidad mediante el cifrado del número del identificador y los datos que contiene dicha información durante la transmisión de los mismos. Si una parte no autorizada obtiene el número del identificador en modo alguno, se ha producido una violación de la confidencialidad.

La pérdida de la confidencialidad de la información puede adoptar muchas formas. Cuando alguien mira por encima de su hombro, mientras usted tiene información confidencial en la pantalla, cuando se publica información privada, cuando un laptop con información sensible sobre una

empresa es robado, cuando se divulga información confidencial a través del teléfono, etc. Todos estos casos pueden constituir una violación de la confidencialidad.

En un sistema que garantice la confidencialidad, un tercero que entra en posesión de la información intercambiada entre el remitente y el destinatario no es capaz de extraer cualquier contenido inteligible.

Para garantizarla se utilizan mecanismos de cifrado y de ocultación de la comunicación. Digitalmente se puede mantener la confidencialidad de un documento con el uso de llaves asimétricas. Los mecanismos de cifrado garantizan la confidencialidad durante el tiempo necesario para descifrar el mensaje. Por esta razón, es necesario determinar durante cuánto tiempo el mensaje debe seguir siendo confidencial. No existe ningún mecanismo de seguridad absolutamente seguro [5].

2.3.6 No repudio

Proporciona protección contra la interrupción, por parte de alguna de las entidades implicadas en la comunicación, de haber participado en toda o parte de la comunicación. El servicio de Seguridad de No repudio o irrenunciabilidad está estandarizado en la ISO-7498-2.

No Repudio de origen: El emisor no puede negar que envió porque el destinatario tiene pruebas del envío, el receptor recibe una prueba infalsificable del origen del envío, lo cual evita que el emisor, de negar tal envío, tenga éxito ante el juicio de terceros. En este caso la prueba la crea el propio emisor y la recibe el destinatario.

Prueba que el mensaje fue enviado por la parte específica.

No Repudio de destino: El receptor no puede negar que recibió el mensaje porque el emisor tiene pruebas de la recepción. Este servicio proporciona al emisor la prueba de que el destinatario legítimo de un envío, realmente lo recibió, evitando que el receptor lo niegue posteriormente. En este caso la prueba irrefutable la crea el receptor y la recibe el emisor.

Prueba que el mensaje fue recibido por la parte específica.

Si la autenticidad prueba quién es el autor de un documento y cuál es su destinatario, el “no repudio” prueba que el autor envió la comunicación (no repudio en origen) y que el destinatario la recibió (no repudio en destino).

El no repudio evita que el emisor o el receptor nieguen la transmisión de un mensaje. Así, cuando se envía un mensaje, el receptor puede comprobar que, efectivamente, el supuesto emisor envió el mensaje. De forma similar, cuando se recibe un mensaje, el emisor puede verificar que, de hecho, el supuesto receptor recibió el mensaje. Definición según la recomendación X.509 de la

UIT-T Servicio que suministra la prueba de la integridad y del origen de los datos- ambos en una relación infalsificable que pueden ser verificados por un tercero en cualquier momento [5].

2.4 Sistema Manejador de Bases de Datos (DBMS)

En esencia en DBMS es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos.

El objetivo principal del sistema manejador de base de datos es proporcionar una forma de almacenamiento y recuperación de la información en una base de datos de manera que sea tanto práctica como eficiente. Se compone de un lenguaje de definición de datos (DDL: Data Definition Language), de un lenguaje de manipulación de datos (DML: Data Manipulation Language) y de un lenguaje de consulta (SQL: Structured Query Language). Los cuales se describirán a detalle más adelante [17].

En la siguiente figura se muestra una representación muy simplificada de un sistema de bases de datos. En ella se pretende indicar que un sistema de bases de datos consta de cuatro componentes principales: datos, hardware, software y usuarios.

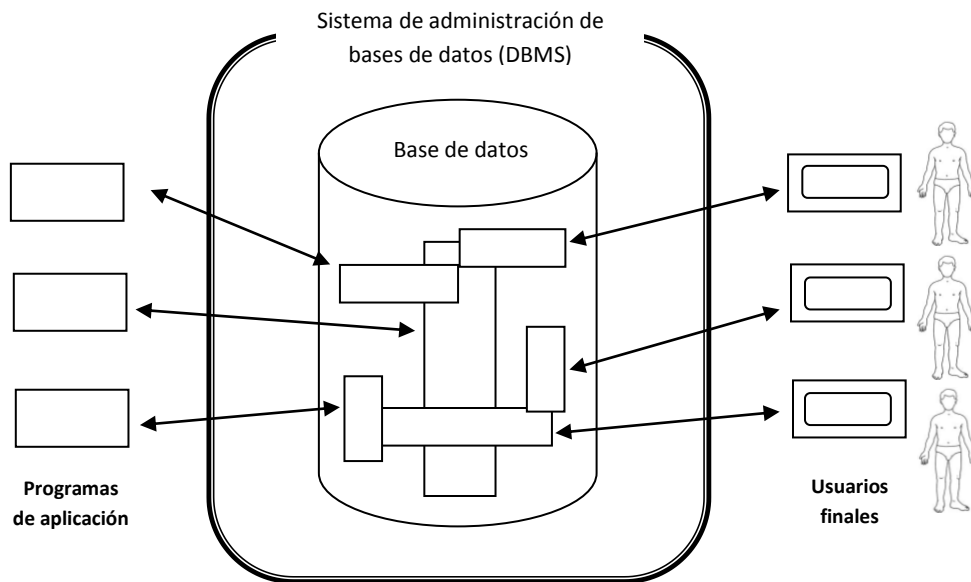


Figura 2.11 Sistema de Administración de Bases de Datos

Una base de datos en un repositorio de datos almacenados, y, en general, es tanto integrada como compartida.

Se entiende por una base de datos *integrada* porque existe una unificación de varios archivos de datos independientes, donde se elimina parcial o totalmente cualquier redundancia entre los mismos.

Y una base de datos *compartida* es porque ésta tiene un comportamiento *concurrente*, es decir, la oportunidad de que diversos usuarios accesen en realidad a la base de datos, incluso puede ser la misma parte de la base de datos, al mismo tiempo.

Un sistema de bases de datos que admite esta forma de comportamiento en ocasiones se llama *sistema de usuarios múltiples*. [17]

2.4.1 Antecedentes y evolución de las Bases de Datos

El campo de la programación de sistemas es producto principalmente del esfuerzo de muchos programadores y administradores que han dedicado su capacidad creadora a producir programas de sistemas prácticos y útiles para atender las demandas planteadas por el auge de la industria de la computación. Al principio, la práctica de la programación tuvo calidad de arte, pues cada programador tenía que inventar sus propias soluciones a diversos datos.

El nacimiento de las bases de datos se dio gracias al Ingeniero Herman Hollerit (1860-1929). Fue denominado el primer ingeniero estadístico de la historia, ya que inventó una computadora llamada "*Máquina Automática Perforadora de Tarjetas*". Para hacer el censo de Estados Unidos en 1880 se tardaron 7 años para obtener resultados [18].

En la década de 1950 se da origen a las cintas magnéticas. Por medio de este mecanismo se empezó a automatizar la información de las nóminas. Consistía en leer una cinta o más y pasar los datos a otra, y también se podían pasar desde las tarjetas perforadas. Simulando un sistema de respaldo de información. Para la década de 1960 el uso de los discos en ese momento fue un adelanto muy efectivo, ya que por medio de este soporte se podía consultar la información directamente, esto ayudó a ahorrar tiempo. No era necesario saber exactamente donde estaban los datos en los discos, ya que en milisegundos era recuperable la información. Los discos dieron inicio a las Bases de Datos, de red y jerárquicas, pues los programadores con su habilidad de manipulación de estructuras junto con las ventajas de los discos era posible guardar estructuras de datos organizadas y mantener un conjunto de datos conectados entre sí [18].

En 1968, Ascher Opler, que en ese tiempo pertenecía al personal de IBM, advirtió que era necesario unificar el conocimiento sobre programación de manera que estuviese al alcance de todos los programadores de sistemas. En respuesta a la recomendación de Opler, IBM decidió patrocinar la *Serie sobre programación de sistemas*, un ambicioso proyecto cuya finalidad era reunir, organizar y publicar los principios y las técnicas que tuvieran utilidad permanente en la industria en general [19].

Fue en 1970 cuando E.F. Codd de IBM introdujo la idea de un modelo relacional de Bases de datos en un documento titulado “*A Relational Model of data for Large Shared Banks*” (Un modelo relacional de datos para grandes bancos de datos compartidos). El objetivo de ese documento es minimizar la redundancia de datos, dividiéndolos en distintos grupos no duplicados. Incrementar la consistencia de datos, por ejemplo si se cambia el nombre de un cliente, este cambiará en todos los reportes. Posteriormente, fue creado un lenguaje llamado SQL (Lenguaje estructurado de consultas) también desarrollado por IBM, para generar reportes y actualizar datos en este nuevo modelo relacional. El Sistema R de IBM nació de este trabajo, pero fue ignorado por IBM, y poco después Oracle saco su versión comercial de BD basada en la teoría relacional de Codd [3].

Las bases de datos han evolucionado bastante gracias a su amplia demanda en distintos campos organizacionales. Existen tres generaciones que nos explican la evolución de las bases de datos [20].

- 1a Generación: Modelo jerárquico
(1960, Medios de 1970) Sistema de bases de datos Jerárquico. IMS (Information Management System).
- 2a Generación: Modelo relacional
(Medios 1970) con Edgar Frank Codd, surgen las bases de datos relacionales. Sistemas Comerciales: Sistema R, SQL, SQL/DS, DB2, ORACLE, IDM, RTI. INGRES, IDM (Intelligent Database Maching). UNIFY, INFORMIX, SUPRA.
- 3a Generación: Modelo orientado a objetos
Modelos Semánticos. Ciertos tipos de inferencia, integran a los sistemas de almacenamiento de datos. Modelos Orientados a Objetos. Objeto y versiones de objetos, consulta de ciertos datos. Bases de datos Deductivas. Bases de Datos no convencionales: CAD/CAM. CASE. DIMS (Sistema de manejo de Imágenes de Documentos). GIS (Sistemas de Información Geográficos).

Como campo comercial, el Sistema Manejador de Bases de Datos aún es relativamente nuevo; los fabricantes y vendedores no empezaron a ofrecer estos sistemas hasta mediados de la década de 1960 (aunque es verdad que ciertos paquetes de software antiguos incluían algunas de las funciones que ahora asocian con tales sistemas). Pese a su calidad de innovación, sin embargo, el campo rápidamente ha cobrado importancia práctica y teórica. La cantidad total de datos encomendados a las bases de datos se mide, sin exagerar, en varios miles de millones de bytes; la inversión financiera al respecto alcanza una cifra igualmente enorme; y no es exagerado afirmar que muchos miles de organizaciones dependen de la operación continuada y eficaz de un sistema de bases de datos.

Las Bases de Datos evolucionan y seguirán evolucionando para satisfacer las nuevas necesidades sobre el tratamiento de datos que vayan surgiendo. No obstante, su futuro dependerán en gran

medida, en la capacidad de conseguir que con esa base de datos, seamos capaces de utilizar estándares reconocidos, de conectarnos, manipular y sincronizarnos con el inmenso mundo de los dispositivos: ya sean móviles con acceso a Internet, e incluso pasando por aquellas bases de datos tan importantes como las médicas o militares, las cuales, además, deberán implementarse en el área de seguridad. Se automatizarán mucho más aspectos, se podrá trabajar en varias bases de datos a la vez y en tiempo real de forma más sencilla, eficaz y segura. En fin, se trata de una tecnología que siempre irá evolucionando, y que se especializará constantemente, debido a que la información es algo que siempre hace falta y que se consulta.

2.4.2 Definición y características de una Base de Datos

Podemos definir a una Base de Datos, como un almacén de datos con diferentes modos de organización, es decir es un conjunto de información que está almacenada en forma sistemática, de manera tal que los datos que la conforman puedan ser utilizados en forma fragmentada cuando sea necesario. Una base de datos representa algunos aspectos del mundo real, aquellos que le interesan al diseñador. Se diseña y se almacenan datos con un propósito específico. Con la palabra "datos" se hace referencia a hechos conocidos que pueden registrarse, como pueden ser números telefónicos, direcciones, nombres, etc. Todo depende de la finalidad para la que sea diseñada la base, permitiendo manipularlos fácilmente y mostrarlos de diversas formas [21].

Un sistema de bases de datos proporciona a la empresa un *control centralizado* de sus datos de operación que, constituyen uno de sus activos más valiosos, es decir, cuando una empresa u organización decide utilizar un sistema de bases de datos, debe existir una persona específica cuya responsabilidad central sea controlar los datos de operación. Esta persona es el administrador de bases de datos (DBA). El DBA debe tener la libertad de modificar la estructura almacenada o la estrategia de acceso (o ambas) en respuesta al cambio de necesidades sin tener que alterar las aplicaciones existentes. La función del DBA requiere un elevado nivel de destreza técnica y capacidad de entender e interpretar los requerimientos administrativos a nivel gerencial.

Consideremos ahora algunas de las características y ventajas de tener un control centralizado de los datos [21]:

- *Reducción de Redundancia.*

En sistemas que no usan bases de datos, cada aplicación tiene sus propios archivos privados. Esto a menudo origina enorme redundancia en los datos almacenados, así como desperdicio resultante del espacio de almacenamiento. Para eliminar la redundancia, es necesario integrar todos los archivos e información generada en la organización. Para integrar la información, el DBA debe tener el control global de la misma, pues está consciente de los requerimientos de información.

En un sistema de bases de datos, la redundancia debe controlarse, es decir, el sistema debe estar al tanto de la redundancia y asumir la responsabilidad de propagar las actualizaciones.

- *Evitar la Inconsistencia de datos.*

Cuando la información es representada por dos entradas distintas en la base de datos, y que el sistema no está al tanto de esta duplicación (es decir, la redundancia no se ha controlado), se dice entonces que la base de datos es *inconsistente*. Desde luego, una base de datos que se halle en estado de inconsistencia puede suministrar información incorrecta o contradictoria.

Si la redundancia no se suprime, pero se controla (dándole conocer al sistema), entonces éste puede garantizar que la base de datos nunca sea inconsistente para el usuario al asegurar cualquier cambio hecho a una de las dos entradas se efectúe de manera automática en la otra. Este proceso se denomina *propagación de actualizaciones* – donde el término *actualización* se usa para abarcar todas las operaciones de creación, supresión y modificación - .

Sin embargo pocos son los sistemas capaces de propagar las actualizaciones de modo automático; es decir, la mayoría de los sistemas modernos de ninguna manera admiten redundancia controlada.

- *Datos Compartidos.*

No solo significa que las aplicaciones existentes pueden compartir los datos de la base de datos, si no también que es factible desarrollar nuevas aplicaciones que operen con los mismos datos almacenados. En otras palabras, las necesidades de datos de las nuevas aplicaciones pueden atenderse sin tener que crear nuevos archivos almacenados.

- *Cumplimiento de normas establecidas.*

Con un control central de la base de datos, el DBA puede garantizar que se cumplan todas las formas aplicables a la representación de los datos. Las normas aplicables pueden comprender la totalidad o parte de lo siguiente: normas de la compañía, de instalación, departamentales, industriales, nacionales o internacionales. Es muy deseable unificar los formatos de los datos almacenados como ayuda para el intercambio o migración de datos entre sistemas.

- *Restricciones de Seguridad.*

Al tener jurisdicción completa sobre los datos de operación, el DBA puede:

- a) Asegurar que el único medio de acceder la base de datos sea a través de los canales establecidos.
- b) Definir controles de autorización para que se apliquen cada vez que se intente el acceso a datos sensibles.

Diferentes controles pueden establecerse para cada tipo de acceso (recuperación modificación, supresión, etc.) a cada parte de la información de la base de datos.

- *Integridad de datos.*

El problema de la integridad es garantizar que los datos de la base de datos sean exactos. La inconsistencia entre dos entradas que representan al mismo *hecho* es un ejemplo de falta de integridad (que solo ocurre si existe redundancia en los datos almacenados). Aun cuando la redundancia se elimine, la base de datos puede contener datos incorrectos; por ejemplo, un empleado o la lista de números de empleado de un departamento específico puede incluir el número de un empleado inexistente. El control centralizado de la base de datos ayuda a evitar estas situaciones en la medida de lo posible, pues permite al DBA definir procedimientos de validación que habrán de ejecutarse cada vez que se intente una operación de actualización.

Es conveniente señalar que la integridad de los datos es más importante en un sistema de bases de datos que en un sistema de *archivos privados*, precisamente porque el primero se comparte y porque sin procedimientos de validación adecuados es posible que un programa con errores genere datos incorrectos que afecten a otros programas que utilicen esa información.

- *Equilibrio entre requerimientos contradictorios.*

Cuando conoce los requerimientos globales de la empresa, el DBA puede estructurar el sistema de bases de datos para brindar un servicio que sea el mejor para la empresa en términos globales. La tecnología de las bases de datos se ha descrito como una de las áreas de la ciencia de la computación y la información de más rápido desarrollo.

2.4.3 Modelo de Bases de Datos

Un modelo de bases de datos es básicamente una *descripción* de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos [20].

Los diferentes modelos de datos se clasifican en tres grupos diferentes:

- 1) Modelos lógicos basados en registros
 - Jerárquico
 - Red
 - Relacional

Este modelo lógico basado en registros se utiliza para describir datos en los niveles conceptual y físico y para especificar la estructura lógica global de la base de datos. Utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores. Proporcionan una descripción a nivel más alto de la implementación.

2) Modelos lógicos basados en objetos

- Entidad-relación
- Orientado a objetos

El modelo lógico basado en objetos se usa para describir datos en el nivel conceptual y en el de visión, es decir, con estos modelos representamos los datos tal y como los captamos en el mundo real, tiene una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente.

3) Modelos físicos

- Unificador
- Memoria de cuadros o elementos

Los modelos físicos también son llamados “*modelos de datos primitivos*”, describen los datos en el nivel más bajo y permiten identificar algunos detalles de implantación para el manejo del hardware de almacenamiento.

Todo modelo de Bases de Datos nos proporciona tres tipos de herramientas:

- a) Modelo Estructuras de datos con las que se puede construir la Base de Datos: tablas, árboles, etc.
- b) Diferentes tipos de restricciones (o reglas) de integridad que el SGBD tendrá que hacer cumplir a los datos: dominios, claves, etc.
- c) Una serie de operaciones para trabajar con los datos. Un ejemplo de ello, en el modelo relacional, es la operación SELECT, que sirve para seleccionar (o leer) las filas que cumplen alguna condición. Un ejemplo de operación típica del modelo jerárquico y del modelo en red podría ser la que nos dice si un determinado registro tiene “hijos” o no.

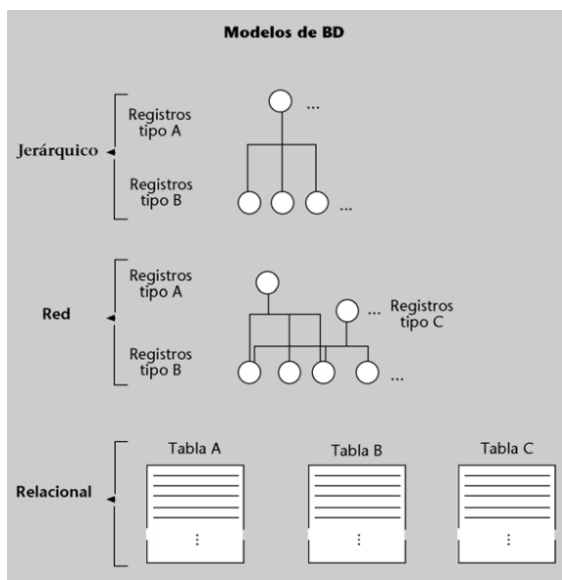


Figura 2.12 Modelos de Bases de Datos

Hablamos de modelos de BD, pero de hecho se acostumbran a denominar modelos de datos, ya que permiten modelarlos. Sin embargo, hay modelos de datos que no son utilizados por los SGBD del mercado: sólo se usan durante el proceso de análisis y diseño, pero no en las realizaciones.

Los más conocidos de estos tipos de modelos son los modelos semánticos y los funcionales. Éstos nos proporcionan herramientas muy potentes para describir las estructuras de la información del mundo real, la semántica y las interrelaciones, pero normalmente no disponen de operaciones para tratarlas. Se limitan a ser herramientas de descripción lógica. Son muy utilizados en la etapa del diseño de BD y en herramientas CASE. El más extendido de estos modelos es el conocido como modelo ER (entity-relationship).

Actualmente, la práctica más extendida en el mundo profesional de los desarrolladores de Sistemas de Información es la utilización del modelo ER durante el análisis y las primeras etapas del diseño de los datos, y la utilización del modelo relacional para acabar el diseño y construir la BD con un SGBD.

2.4.3.1 Modelo Entidad-Relación (MER)

El modelo E-R es un modelo de datos semántico cuyo aspecto yace en la representación del significado de los datos. Hay tres nociones básicas que emplea este modelo: conjuntos de entidades, conjuntos de relaciones y atributos [21].

Características del modelo E-R extendido

Aunque los conceptos básicos de E-R pueden modelar la mayoría de las características de las bases de datos, algunos aspectos de una base de datos pueden ser más adecuadamente expresados mediante ciertas extensiones del modelo E-R básico tales como: la especialización, la generalización, los conjuntos de entidades de nivel más alto y más bajo, la herencia de atributos y la agregación.

2.4.3.2 Modelo Relacional

Lo que realmente marca la diferencia entre los sistemas relacionales y los sistemas de modelo jerárquico de datos y el modelo de datos en red, es el hecho de que su creador Ted Codd, basó expresamente su funcionamiento sobre un modelo matemático muy específico: el álgebra relacional y el cálculo relacional, así como la progresiva adopción, por parte de su creador y algunos colaboradores, de un número de Reglas de Integridad Relacional y de Formas Normales.

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia

entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional [18].

Debido a que el modelo relacional está basado en la teoría de conjuntos, los datos se organizan en un tipo especial de conjunto denominado relación, que se define de la siguiente manera: *Sean los conjuntos D_1, \dots, D_n , denominados dominios, que no tienen por qué ser distintos entre sí. Una relación definida sobre D_1, \dots, D_n , es cualquier subconjunto R de D , donde n es el grado o variedad de R . Los dominios son en principio conjuntos finitos de datos. Por tanto, a menos que se indique lo contrario, las relaciones también son finitas. Los elementos de una relación se denominan **tuplas**. Formalmente, una tupla es:*

$$\langle d_1, \dots, d_n \rangle, \text{ donde } d_1 \in D_1, \dots, d_n \in D_n$$

El número de tuplas en una relación es la cardinalidad de la relación. Puesto que una relación es un conjunto, los elementos de este conjunto, las tuplas, han de ser por fuerza distintas. Esto también implica que el orden de las tuplas es irrelevante. El conjunto vacío es una relación particular: la relación nula o vacía.

Las relaciones también pueden ser vistas como tablas, en las que cada tupla es una fila de la tabla. Los nombres de las columnas de la tabla, por otra parte, son los atributos. El conjunto (ordenado) de todos los atributos de una relación R es el esquema de R . Las tuplas pueden ser consideradas como matrices de pares (atributo, valor).

Los términos formales del modelo relacional a menudo son sustituidos por otros de uso más común, debido a que estos términos son muy abstractos para ser usados en la práctica.

Tabla 2.2 Términos Relacionales y sus equivalentes

Término relacional	Equivalente
Relación	Tabla
Tupla	Fila o Registro
Cardinalidad	Número de fila o registros
Atributo	Columna o campo
Grado	Número de columnas o campos
Clave primaria	Identificador único
Dominio	Conjunto de valores legales

Tabla 2.3 Ejemplo: Relación Alumno

	Nombre - Alumno	Carrera – Alumno	Número Cta. - Alumno
Registros, Renglones, Tuplas	Axel	Ing. Eléctrico-Electrónico	405023622
	Anabel	Ing. Computación	302166239
	Juan Carlos	Ing. Computación	303489797
	Cristina	Ing. Computación	315687390
	Karen	Ing. Telecomunicaciones	405098724

Campos,
Columnas,
Atributos

Tabla Ejemplo: La Relación Alumno

Reglas de Codd

Regla 0. Cualquier DBMS que proclame ser relacional, deberá manejar, completamente, las bases de datos por medio de sus capacidades relacionales.

Regla 1 (Regla de la Información). Toda la información dentro de una base de datos relacional se representa de manera explícita a nivel lógico y exactamente de una sola manera, como valores en una tabla.

Regla 2 (Regla del acceso garantizado). Se garantiza que todos y cada uno de los datos (valor atómico) en una base de datos relacional pueden ser leídos recurriendo a una combinación de nombre tabla, valor de la llave primaria y nombre de la columna.

Regla 3 (Tratamiento sistemático de valores nulos). En un DBMS totalmente relacional se soportan los valores nulos (que son distintos de una cadena de caracteres vacía o de una cadena con caracteres en blanco o de cero o cualquier otro número). Para representar información faltante o no aplicable de una forma consistente independientemente del tipo de dato.

Regla 4 (Diccionario dinámico en línea basado en el modelo relacional). La descripción de la base de datos se representa en el nivel lógico de la misma forma que los datos ordinarios, de modo que los usuarios autorizados puedan aplicar el mismo lenguaje relacional para consultarla, que aquel que emplea con sus datos normales.

Regla 5 (Regla del sublenguaje de datos completo). Un sistema relacional debe soportar varios lenguajes y varios modos de uso terminal. Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante una sintaxis bien definida, como cadenas de caracteres y que sea completo, es decir, deberá contener un sublenguaje de datos completo llamado SQL (Structured Query Lenguaje) que comprenda como estándar lo siguiente:

- Definición de datos
- Definición de vistas

- Manipulación de datos
- Restricciones de integridad (manejo)
- Autorización
- Inicio y fin de una transacción

Regla 6 (Regla de actualización de vistas). Todas las vistas que teóricamente sean actualizables, deben ser actualizadas por medio del sistema.

Regla 7 (Inserción, actualización y borrado de alto nivel). La posibilidad de manejar una relación base o una relación derivada como un solo operador se aplica a la lectura, inserción, modificación y eliminación de datos.

Regla 8 (Independencia física de datos). Los programas de aplicación y la actividad en terminales no deberán ser afectados a nivel lógico por cambios en el almacenamiento físico de los datos o en los métodos de acceso.

Regla 9 (Independencia lógica de datos). Los programas de aplicación y la actividad en terminales no deberán ser afectados por cambios de cualquier tipo, que preserven la información y que teóricamente permitan la afectación, en las tablas base.

Regla 10 (Independencia de integridad). Las restricciones de integridad de una base de datos relacional deberán poder definirse con el mismo sublenguaje de datos relacional y deberá almacenarse en el catálogo, no en los programas de aplicación.

Regla 11 (Independencia de distribución). Un DBMS relacional tiene independencia de distribución, esto es, si se manipulan varios fragmentos de la misma base en diferentes medios de almacenamiento, tener la capacidad como si fuera una sola.

Regla 12 (Regla de la no subversión). Si un sistema relacional tiene un lenguaje de bajo nivel (que opere un registro cada vez), ese lenguaje no deberá poder emplearse para subvertir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de más alto nivel.

2.4.3.3 Normalización

La normalización es un proceso de descomposición sin pérdida, para lograr que nuestras bases de datos estén lo más óptimas posibles [17].

Una dependencia funcional se representa cuando los valores de un conjunto de atributos de una tupla determinan de manera única los valores de otro conjunto de atributos.

La teoría de normalización está basada en la observación de que cierto conjunto de relaciones presenta mejores propiedades en un medio de actualización, inserción y supresión, que las que presentan otros conjuntos de relaciones que contienen los mismos datos.

La razón de usar el procedimiento de normalización es asegurar que el modelo conceptual de la base de datos funcionará. Esto no significa que una estructura no normalizada no funcionará, si no que puede causar algunos problemas cuando los programadores de aplicación traten de modificar la base de datos.

Las formas normales son una serie de restricciones que se definen sobre las estructuras relaciones para evitar, como ya se señaló, anomalías al efectuar adiciones, eliminaciones o actualizaciones de tuplas. Con el fin de conseguir que una relación cumpla con una forma normal se efectúa un proceso de descomposición. Ésta implica dividir los atributos de una relación en dos subconjuntos (posiblemente con una intersección no vacía) sin que por ello se pierda alguna información contenida en la relación original.

Las formas de normalización fueron propuestas originalmente por Codd, en 1971 y en 1972. Posteriormente varios investigadores continuaron trabajando en esta teoría y a lo largo del tiempo han surgido varias formas de normalización que complementan y refuerzan a las enunciadas por Codd.

- **Primera Forma Normal (1FN)**

Una relación normalizada es una relación que tiene sólo valores elementales (o simples) en la intersección de cada renglón y columna. Así, una relación está en primera forma normal si no contiene grupos repetitivos.

Una entidad R está en Primer Forma Norma (1FN) si los valores, para cada atributo $A \in R$, son atómicos.

Esto implica, que los valores en el dominio no deberán ser listas o conjuntos de valores. Las actualizaciones representan un problema potencial si la entidad no está en 1NF.

- **Segunda Forma Normal (2FN)**

Para eliminar anomalías de la primera forma normal, se deben eliminar las dependencias parciales. Una relación está en segunda forma normal, si está en primera forma normal y se han eliminado las dependencias parciales.

Se entiende por dependencia parcial cuando los atributos no llave dependen sólo de una parte de una llave compuesta.

El segundo paso de la normalización es establecer las claves y relacionarlas con los campos de datos. En la primera forma normalizada, el renglón entero de la tabla (tupla) depende de todos los campos de claves. En la segunda forma normalizada, se hace un intento de establecer los campos de datos que están relacionados con alguna parte de la clave completa. Si los campos de datos sólo dependen de una parte de la clave, la clave y los campos conectados a la clave parcial son susceptibles de separarse en registros independientes. La división de la primera tabla normalizada,

en una serie de tablas en las que cada campo sólo depende de la clave completa se llama la segunda forma normalizada.

- **Tercer Forma Normal (3FN)**

Una relación está en tercera forma normal si está en segunda forma normal, esto es, cada atributo no llave depende totalmente de la llave primaria y no tiene dependencias transitivas.

Una dependencia transitiva ocurre cuando un atributo no llave depende de uno o más atributos no llave.

- **Forma Norma de Boyce-Codd (FNBC)**

La tabla se encuentra en FNBC si cada determinante, atributo que determina completamente a otro, es clave candidata.

- **Cuarta Forma Normal (4FN)**

Una entidad está en cuarta forma normal cuando:

Está en tercera forma normal y sus atributos no solamente depende de la llave primaria o compuesta, sino que también del valor de la llave (dependencia del valor).

Un atributo cuya existencia es opcional se reubica en una entidad donde este atributo es obligatorio (dependencia de la existencia).

En otras palabras, una tabla se encuentra en 4FN si, y sólo si, para cada una de sus dependencias múltiples no funcionales $X \twoheadrightarrow Y$, siendo X una súper clave que, X es o una clave candidata o un conjunto de claves primarias.

- **Quinta Forma Normal (5FN)**

Una tabla se encuentra en 5FN si:

La tabla se encuentra en 4FN

No existen relaciones de dependencias no triviales que no siguen los criterios de las claves. Una tabla que se encuentra en la 4FN se dice que está en la 5FN si, y sólo si, cada relación de dependencia se encuentra definida por las claves candidatas.

2.4.4 Niveles de abstracción de datos

Hay tres características importantes inherentes a los sistemas de bases de datos: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos. En 1975, el

comité ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los sistemas de bases de datos, que resulta muy útil a la hora de conseguir estas tres características [17].

La definición de un sistema de información es la descripción detallada de la arquitectura del sistema. Las arquitecturas de bases de datos han evolucionado mucho desde sus comienzos, aunque la considerada estándar hoy en día es la descrita por el comité ANSI/X3/SPARC (Standard Planning and Requirements Committee of the American National Standards Institute on Computers and Information Processing), que data de finales de los años setenta. Este comité propuso una arquitectura general para DBMS's basada en tres niveles o esquemas:

- **Nivel interno (de máquina):** Tiene un esquema interno que describe la estructura física de almacenamiento de base de datos. Emplea un modelo físico de datos y los únicos datos que existen están realmente en este nivel.
- **Nivel conceptual:** tiene esquema conceptual. Describe la estructura de toda la base de datos para una comunidad de usuarios. Oculta los detalles físicos de almacenamiento y trabaja con elementos lógicos como entidades, atributos y relaciones.
- **Nivel externo (de vistas):** tiene varios esquemas externos o vistas de usuario. Cada esquema describe la visión que tiene de la base de datos a un grupo de usuarios, ocultando el resto.

El objetivo de estos tres niveles de abstracción es el de separar los programas de aplicación de la base de datos física, así mismo describir las interacciones entre estos tres niveles y todos los elementos que conforman cada uno de ellos.

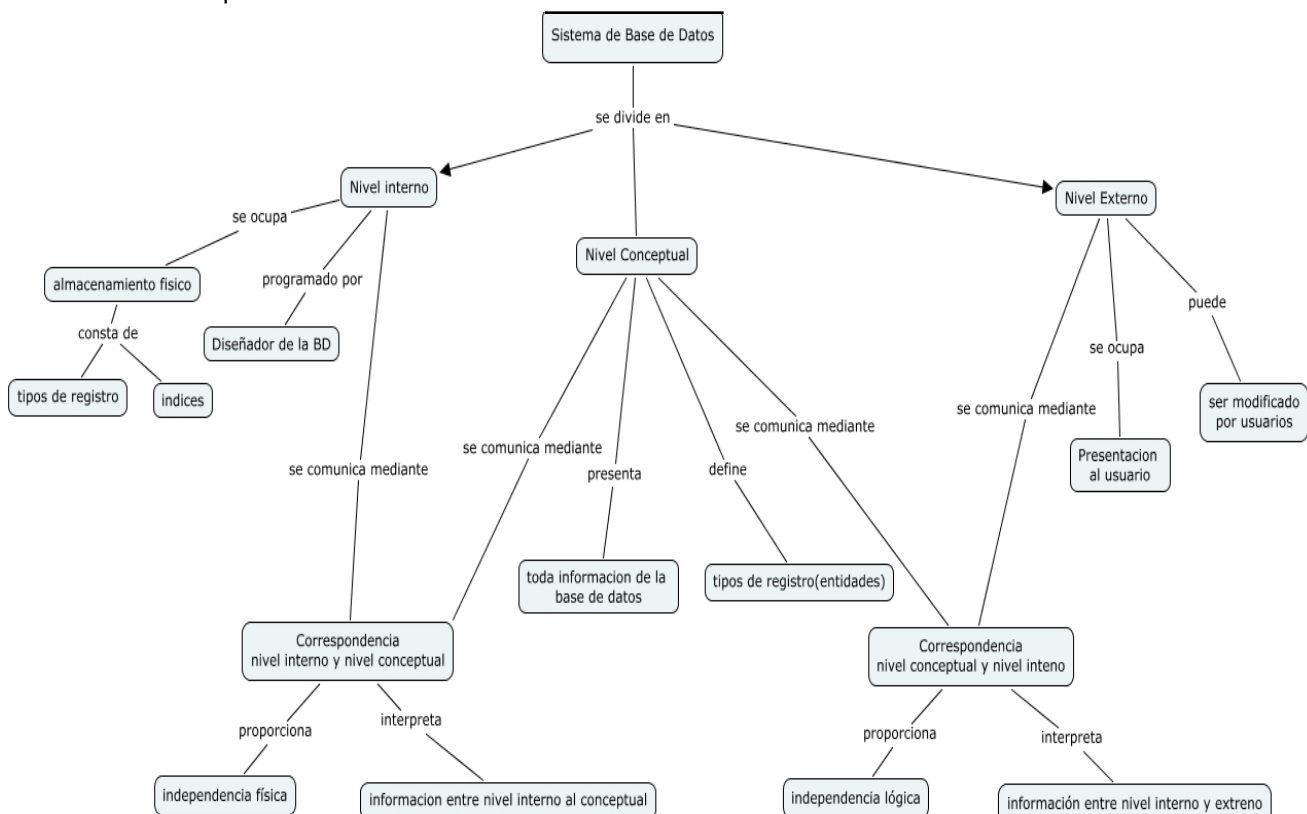


Figura 2.13 Niveles de Abstracción en una Base de Datos

Hay que destacar que los tres esquemas no son más que descripciones de los mismos datos pero con distintos niveles de abstracción. Los únicos datos que existen realmente están a nivel físico, almacenados en un dispositivo como puede ser un disco. En un SGBD basado en la arquitectura de tres niveles, cada grupo de usuarios hace referencia exclusivamente a su propio esquema externo. Por lo tanto, el SGBD debe transformar cualquier petición expresada en términos de un esquema externo a una petición expresada en términos del esquema conceptual, y luego, a una petición en el esquema interno, que se procesará sobre la base de datos almacenada. Si la petición es de una obtención (consulta) de datos, será preciso modificar el formato de la información extraída de la base de datos almacenada, para que coincida con la vista externa del usuario. El proceso de transformar peticiones y resultados de un nivel a otro se denomina correspondencia o transformación.

La arquitectura de tres niveles es útil para explicar el concepto de independencia de datos que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior [20].

Se pueden definir dos tipos de independencia de datos:

- **La independencia lógica** es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.
- **La independencia física** es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.

2.5 Lenguajes de Bases de Datos

Un sistema de bases de datos proporciona un lenguaje de definición de datos para especificar el esquema de la base de datos y un lenguaje de manipulación de datos para expresar las consultas a la base de datos y las modificaciones. Los lenguajes de definición y manipulación de datos no son dos lenguajes separados; forman parte de un único lenguaje de bases de datos, tal como SQL.

Existen tres tipos de manejadores de datos:

- DLL (Data Definition Language) Lenguaje de Definición de Datos, que permiten crear y definir nuevas bases de datos, campos e índices.
- DML (Data Manipulation Language) Lenguaje de Manipulación de Datos, que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

- DCL (Data Control Language) Lenguaje de Control de Datos, que permite la definición de lo que los usuarios pueden hacer dentro de la base de datos [17].

2.5.1 SQL

El lenguaje de consulta estructurado (SQL) es un lenguaje declarativo y normalizado de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar de manera sencilla la información de interés de una base de datos, así como también hacer cambios u operaciones sobre la misma, ya sea en los datos, o bien, en la estructura. Es un lenguaje de cuarta generación (4GL) [22].

Los orígenes del SQL están ligados a los de las bases de datos relacionales. En 1970 E.F.Codd propone el modelo relacional y asociado a este un sublenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas, los laboratorios de IBM definen el lenguaje SEQUEL (Structured English QUery Language) que más tarde sería ampliamente implementado por el SGBD experimental System R, desarrollado en 1977 también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en 1979 en un programa comercial.

El SEQUEL terminaría siendo el predecesor de SQL, siendo éste una versión evolucionada del primero. El SQL pasa a ser el lenguaje por excelencia de los diversos SGBD relacionales surgidos en los años siguientes y es por fin estandarizado en 1986 por el ANSI, dando lugar a la primera versión estándar de este lenguaje, el SQL-86 o SQL1. Al año siguiente este estándar es también adoptado por la ISO. Sin embargo este primer estándar no cubre todas las necesidades de los desarrolladores e incluye funcionalidades de definición de almacenamiento que se consideraron suprimir. Así que en 1992 se lanza un nuevo estándar ampliado y revisado del SQL llamado SQL-92 o SQL2.

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos [22].

2.5.2 Lenguaje de definición de datos

El lenguaje de definición de datos (DDL). Es utilizado para describir todas las estructuras de información y los programas que se usan para construir, actualizar e introducir la información que contiene una base de datos. El DDL permite al administrador de la base especificar los elementos de datos que la integran, su estructura y las relaciones que existen entre ellos, así como también las reglas de integridad y los controles a efectuar antes de autorizar el acceso a la base [20].

La definición de la estructura de la base de datos incluye tanto la creación inicial de los diferentes objetos que formarán la base de datos, como el mantenimiento de esa estructura. A continuación mostramos las sentencias del DDL que se utilizan para distintos objetos:

Tabla 2.4 Lenguaje de DDL

	CREATE DATABASE	CREATE TABLE	CREATE INDEX
CREATE	es la sentencia para crear una base de datos	nos permite crear una nueva tabla	crear un nuevo índice
DROP	DROP TABLE	DROP INDEX	
	Para eliminar una tabla	Para eliminar un índice	
ALTER	ALTER TABLE	ALTER INDEX	
	para modificar algo de la definición de una tabla ya creada	para modificar algo de la definición de un índice ya creado	
TRUNCATE	Borra la tabla y la vuelve a crear, no ejecuta ninguna transacción.		

2.5.3 Lenguaje de manipulación de datos

El lenguaje de manipulación de datos (DML) es utilizado para escribir programas que crean, actualizan y extraen información de las bases de datos. Siempre de acuerdo con las especificaciones y las normas de seguridad dictadas por el administrador. Un lenguaje de manipulación de datos es un lenguaje que permite a los usuarios acceder o manipular los datos organizados mediante el modelo de datos apropiado. Hay dos tipos básicamente [20]:

- **Procedimentales:** Requieren que el usuario especifique qué datos se necesitan u cómo obtener esos datos.
- **Declarativos:** Requieren que el usuario especifique qué datos se necesitan sin especificar cómo obtener esos datos.

A continuación mostramos los comandos que se utilizan para la manipulación de datos en un sistema manejador de bases de datos:

Tabla 2.5 Lenguaje de DML

SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
DELETE	Utilizado para modificar los valores de los campos y registros especificados.

UPDATE

Utilizado para eliminar registros de una tabla de una base de datos.

2.5.4 Lenguaje de control de datos

Además de definir y manipular una base de datos relacional, es importante que se establezcan mecanismos de control para resolver problemas de concurrencia de usuarios y garantizar la seguridad de los datos. El lenguaje de control de datos (*DCL* por sus siglas en inglés) es la parte de SQL que se encarga de controlar la seguridad y los permisos de la base de datos. Las directivas de DCL de SQL son más parte de cada uno de los SGBD y tienen una gran cantidad de variaciones [22]. Sus principales características para la protección de los datos son:

- Manipulación de los mismos sin autorización
- Prevención de errores
- Prevención de daños

Maneja básicamente tres niveles de seguridad para los usuarios de una base de datos:

- DB Administrator
- Dueño (*owner*)
- Público (*Public*)

Para la concurrencia de usuarios utilizaremos el concepto de transacción, y para la seguridad veremos cómo se puede autorizar y desautorizar a usuarios a acceder a la base de datos.

- **Transacciones:** Una Transacción es un conjunto de instrucciones de SQL que se ejecutan como una unidad, es decir, o todas las instrucciones se ejecutan exitosamente o ninguna de ellas se ejecuta. Si todas las instrucciones de la transacción son exitosas, se realiza un "*commit*" para que dichas instrucciones tomen un efecto permanente en la base de datos. Si ocurre algún error, se ejecuta un "*rollback*" para cancelar las instrucciones ejecutadas. Cualquier instrucción que se haya ejecutado hasta ese momento dentro de la transacción es deshecha, dejando la base de datos en el mismo estado en el que se encontraba antes de comenzar la transacción. Una transacción comienza con *BEGIN* o *BEGIN TRAN*; Una Transacción termina con *COMMIT*; o *ROLLBACK*
- **Autorización y Desautorización:** Todos los privilegios sobre la base de datos los tiene su propietario, pero no es el único que accede a ésta. Por este motivo, el SQL nos ofrece sentencias para autorizar y desautorizar a otros usuarios, por ejemplo, la sentencia utilizada para autorizar privilegios es *GRANT* y para la desautorización se utiliza *REVOKE*.

La asignación de privilegios en una base de datos se hace por medio de la siguiente instrucción:

```
GRANT lista_de_privilegios [lista_de_columnas]
ON objeto
TO lista_de_usuarios
```

[WITH GRANT OPTION];

Los privilegios que se pueden asignar son:

- Para tablas y vistas:
 - SELECT
 - DELETE
 - INSERT [*lista_de_columnas*]
 - UPDATE [*lista_de_columnas*]
 - REFERENCES [*lista_de_columnas*]

- Dominios, character sets, collations, translations
 - USAGE

Los objetos pueden ser:

- [TABLE] nombre
- DOMAIN
- CHARACTER SET
- COLLATION
- TRANSLATION

La revocación de privilegios en una base de datos se hace por medio de la siguiente instrucción:

```
REVOKE [GRANT OPTION FOR] lista_de_privilegios  
ON objeto  
FROM lista_de_usuarios [RESTRICT | CASCADE]
```

2.6 Arquitectura de 3 niveles

También conocida como arquitectura de tres capas, la arquitectura de tres capas, define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores [23]. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, nos ayuda a identificar qué puede reutilizarse, y proporciona una estructura que nos ayuda a tomar decisiones sobre qué partes comprar y qué partes construir.

Para enfrentarse a estos temas, la comunidad de software desarrolló la noción de una arquitectura de tres niveles. La aplicación se divide en tres capas lógicas distintas, cada una de

ellas con un grupo de interfaces perfectamente definido. La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo.

La capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación. La capa intermedia (lógica de aplicación) es básicamente el código al que recurre la capa de presentación para recuperar los datos deseados. La capa de presentación recibe entonces los datos y los formatea para su presentación.

Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Pueden construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación siempre que está presente una interfaz claramente definida a la capa de presentación.

Capa de presentación

Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información, la cual obtiene del usuario. En el mundo de la informática es conocida como interfaz gráfica y debe tener la característica de ser amigable, es decir, entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa intermedia o de negocio.

Capa de negocio

Es donde residen las funciones que se ejecutan, se reciben las peticiones del usuario, se procesa la información y se envían las respuestas tras el proceso. Se denomina capa de negocio o capa de lógica del negocio, porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, así como también con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de acceso a datos

Esta capa es la encargada de almacenar los datos del sistema y de los usuarios. Su función es almacenar y devolver datos a la capa de negocio, aunque para esto también es necesario en algunos casos, que tengan procedimientos almacenados y funciones dentro de la capa. En una arquitectura de tres capas, esta capa es la única que puede acceder a los mismos. Está formada por uno o varios sistemas gestores de bases de datos, localizados en un mismo servidor o en varios.

La arquitectura en 3 niveles generalmente está compartida por:

1. Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador Web) para la presentación.

2. El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
3. El manejador de datos, que proporciona al servidor de aplicaciones los datos que requiere.

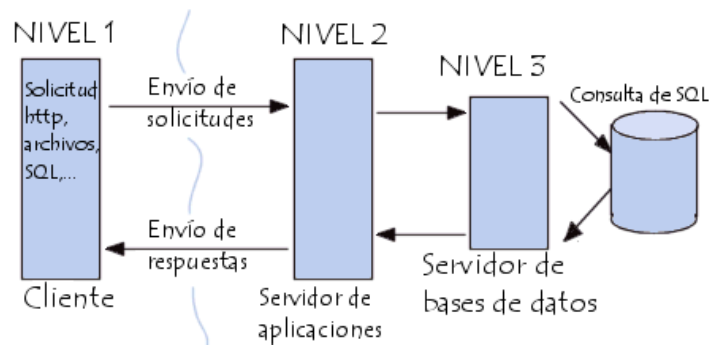


Figura 2.14 Arquitectura de 3 Niveles

2.6.1 Modelo cliente - servidor

En el mundo de TCP/IP las comunicaciones entre computadoras se rigen básicamente por un modelo llamado Cliente-Servidor, éste es un modelo que intenta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones. El término Cliente/Servidor fue usado por primera vez en 1980 para referirse a PC's en red [24].

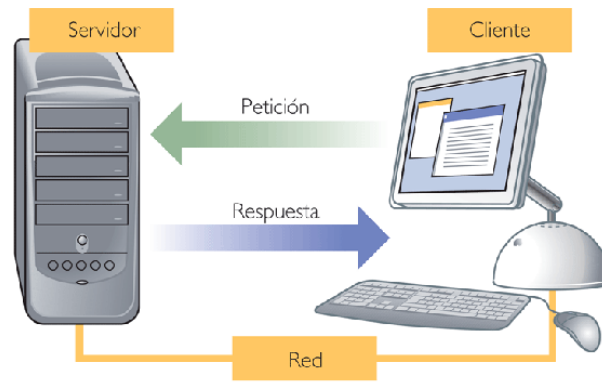


Figura 2.15 Modelo Cliente-Servidor

Este modelo Cliente-Servidor empezó a ser aceptado a finales de los 80's. Su funcionamiento es sencillo: se tiene una máquina cliente, que requiere un servicio de una máquina servidor, y éste realiza la función para la que está programado (nótese que no tienen que tratarse de máquinas diferentes; es decir, una computadora por sí sola puede ser ambos cliente y servidor dependiendo del software de configuración).

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio), como se muestra en la figura. En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.

Cliente

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término *front-end*.

El Cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término *back-end*.

El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

2.6.2 Servidor de Bases de datos

Los servidores de bases de datos surgen con motivo de la necesidad de las empresas de manejar grandes y complejos volúmenes de datos, al tiempo que requieren compartir la información con un conjunto de clientes (que pueden ser tanto aplicaciones como usuarios) de una manera segura. Ante este enfoque, un sistema de administración de bases de datos deberá ofrecer soluciones de forma fiable, rentable y de alto rendimiento. A estas tres características, le debemos añadir una más: debe proporcionar servicios de forma global y, en la medida de lo posible, independientemente de la plataforma.

Los servidores de base de datos se utilizan para la ejecución de los DBMS y que varios usuarios hagan operaciones sobre ellas al mismo tiempo, situado en un lugares diferentes; se puede acceder a las BD por terminales o equipos definidos como *cliente* que permita el acceso a ellas.

A continuación se describen brevemente dos de los manejadores más utilizados hoy en día [25]:

❖ MySQL

Es un sistema de administración de base de datos relacional, multihilo y multiusuario.

Sus principales objetivos han sido la velocidad y la robustez, por lo que se puede decir que es un SGBD sencillo y rápido que se adapta perfectamente a entornos en donde el volumen de datos es del orden de megabytes.

Uno de los atributos que más favorecen a MySQL es que las bases de datos las almacena creando un directorio por cada una de ellas, y dentro de dicho directorio, crea tres ficheros por tabla donde se almacenan los registros de la tabla y la definición de la tabla. El copiar una base de datos es tan fácil como copiar un directorio por lo que hacer copias de seguridad de las mismas es muy sencillo.

Plataformas

MySQL funciona sobre múltiples plataformas, incluyendo:

AIX	BSD	FreeBSD	HP-UX	GNU/Linux
Mac OS X	Net BSD	Novell Netware	OpenBSD	OS/2 Warp
QNX	SGI IRIX	Solaris	SunOS	SCO OpenServer
SCO UnixWare	Tru64	eBD	OpenVMS	Windows 95
Windows 98	Windows NT	Windows 2000	Windows XP	Windows Vista
Windows 7even	Windows Server(2000, 2003 y 2008)			

Entre sus características principales se encuentran:

- EL principal objetivo de este manejador es la velocidad, tanto al conectar con el servidor como al realizar consultas.
- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- Aunque se desconecte el servidor web, no suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, utilizando privilegios de usuarios.
- MySQL se comporta mejor que PostgreSQL a la hora de modificar o añadir campos a una tabla "en caliente".

Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.

❖ PostgreSQL

PostgreSQL es un sistema de administración de bases de datos relacional orientada a objetos de software libre. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

Entre sus características principales se encuentran:

Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP (IPv4 e IPv6), bloques de direcciones estilo CIDR, direcciones MAC, Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

2.6.3 Servidor Web

Un servidor web es un programa que se ejecuta en un equipo servidor, acepta las solicitudes de los exploradores Web y devuelven los resultados en forma de documentos HTML. El explorador y el servidor Web se comunican mediante el protocolo de transferencia de hipertexto (HTTP HyperText Transfer Protocol), realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente. El código recibido por el cliente es compilado y ejecutado por un navegador web.

Los servidores Web pueden disponer de un intérprete de otros lenguajes de programación que ejecutan código embebido dentro del código HTML de las páginas que contiene el sitio antes de enviar el resultado al cliente. Esto se conoce como programación de lado del servidor y utiliza lenguajes como ASP, PHP, Perl y Ajax. Las ventajas de utilizar estos lenguajes radica en la potencia de los mismos ejecutando tareas más complejas como, por ejemplo, acceder a bases de datos abstrayendo al cliente de toda la operación [26].

Podemos distinguir dos aplicaciones con un servidor, del lado del cliente y del lado del servidor.

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o JavaScript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts).
- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Un Servidor Web Local es aquel Servidor Web que reside en una red local al equipo de referencia. El Servidor web Local puede estar instalado en cualquiera de los equipos que forman parte de una red local. Es por tanto obvio, que todos los Servidores Web, son locales a la red local en la que se encuentran, o como mínimo, locales al sistema en el que están instalados.

Cuando un servidor Web se encuentra instalado en el mismo equipo desde el cual se desea acceder puede utilizarse la dirección de Loopback, 127.0.0.1 en Ipv4 y ::1 en Ipv6. El puerto TPC 80 se obvia. Los archivos se almacenan en un directorio determinado por la configuración, generalmente modificable.

Existen numerosas aplicaciones que facilitan la instalación automática de servidores web Apache y aplicaciones adicionales como MySQL y PHP (entre otros), de forma conjunta, como XAMMP, JAMP o EasyPHP. Estas aplicaciones reciben el nombre de LAMP cuando se instalan en plataformas Linux, WAMP en sistemas Windows y MAMP en sistemas Apple Macintosh [26].

CAPÍTULO 3 ANÁLISIS

Para el desarrollo del software se consideraron los siguientes aspectos: equipo disponible, la necesidad a cubrir y el trabajo previo para generar una aplicación que cumpla lo mejor posible con satisfacer la necesidad y que se sortee lo mejor posible las restricciones.

La necesidad de desarrollar este proyecto es de disponer de un software que facilite la realización de encuestas de los locales que conforman a los edificios de las dependencias así como de un sistema informático que automatice el proceso de análisis de información para la generación de propuestas de mejoramiento y disminución del consumo eléctrico.

De las restricciones que se presentan, nos encontramos con el costo para la adquisición de licencias de software privado, que corresponden a herramientas de desarrollo, como son IDE's, compiladores y generadores de interfaces gráficas.

Es necesario contar con un gestor de bases de datos relacionales (RDBMS), herramienta que permite almacenar y gestionar la información que se genere durante las encuestas y durante los análisis. Con la ventaja de que en su uso, se puede tener esquemas de almacenamiento que no contengan redundancia de datos, y que permita realizar consultas complejas, aún aquellas que no se han considerado; buscando tener una aplicación que sea flexible a necesidades futuras.

Para las herramientas de desarrollo es necesario buscar alternativas que permitan desarrollar un sistema que resulte seguro y fiable, pero sin el problema del costo de licencias del software por derecho de uso, o en su defecto, que sea un costo mínimo.

Por tal motivo para el desarrollo del sistema se utiliza software libre, ya que cumple con las necesidades, condiciones y con el análisis de requerimientos para el proceso de desarrollo, pues no se condiciona o limita el uso de los elementos resultantes.

Para el caso de la administración de la base de datos, la ventaja del software libre, es que se podrá desarrollar la aplicación y establecer una comunicación con la base de datos, sin restricciones en cuanto al número de accesos o de personas que pueden ingresar información simultáneamente.

Es decir, con el uso de herramientas de desarrollo que corresponden al software libre, la aplicación resultante no está condicionada ni limitada a un uso exclusivo o particular, lo que garantiza que la FI-UNAM puede hacer lo que crea más conveniente con el software de desarrollo con dichas herramientas sin tener que realizar un pago por su adquisición.

3.1 Análisis técnico

Con el fin de sustentar la estrategia que permita alcanzar las mejoras tecnológicas para el incremento de la productividad y el abatimiento de los costos de energía, se investigó el análisis de la situación energética a nivel nacional e internacional, del cual se deduce que la tendencia internacional está dirigida a disminuir el impacto ambiental provocado por el consumo irracional de módulos energéticos. A nivel nacional en cambio se pretende, por un lado, fortalecer el sector energético y por el otro, lograr un incremento sustancial en la eficiencia y productividad de los diferentes sectores. Un común denominador para los sectores industriales nacionales e internacionales es el interés por reducir los costos de producción y lograr un incremento en el campo tecnológico.

Una base de datos aplicada a un diagnóstico energético es una herramienta muy eficiente para poder reducir los costos de producción en el consumo de energía en Ciudad Universitaria, esta base de datos incluirá los siguientes aspectos:

- a) consumos y costos históricos de todos los tipos de energéticos utilizados durante el periodo más largo posible.
- b) Volúmenes de producción y cualquier información relacionada para el mismo periodo que el de la energía utilizada
- c) Inventario de equipos consumidores de energía.

3.1.1 Diagnóstico energético

Resulta incuestionable la importancia de la energía como insumo estratégico en el desarrollo económico de una institución. La creciente demanda de energía implica un cuidadoso diagnóstico a efecto de optimizar su uso y poder contar con una oferta adecuada a mediano y a largo plazo. De esta manera se evitarán presiones innecesarias que impidan o frenen el desarrollo equilibrado.

Los diagnósticos energéticos son estudios que permiten determinar dónde y cómo se utiliza la energía. No son una solución directa al control de costos en el uso de la energía, pero sí la herramienta más útil para lograr esa función. A través de los diagnósticos, se identifican los puntos del diagrama del proceso de mayor uso de energía haciendo resaltar aquellos donde ésta se desperdicia y aquellos donde es posible generar algún ahorro. Brindan además el apoyo tan necesario a través del cual el desarrollo de cualquier proceso puede ser concluido a buen término, en condiciones óptimas.

El diagnóstico energético es la herramienta básica para minimizar los costos y aumentar la productividad, calidad y el control del efecto sobre el medio ambiente de cualquier empresa o institución, en las nuevas condiciones que se están creando a nivel nacional e internacional [27].

No es la única herramienta si no que está acompañada de todas las actividades relacionadas con la administración, organización, seguimiento y evaluación permanente de los resultados obtenidos y

de toma de medidas correctivas, las cuales permitirán lograr la continuidad de las acciones recomendadas.

Existen tantos tipos de diagnósticos como procesos industriales, variando en tamaño, enfoque, precisión y costos, dependiendo de las fuentes y necesidades del proceso en el cual se desarrolla el mismo. Sin embargo, es conveniente dividir cualquier diagnóstico energético en tres niveles: "A", "B" y "C"

El nivel "A" provee la orientación necesaria para cumplir las funciones del departamento de conservación de energía o su equivalente. Este nivel es el menos costoso de los tres y da idea de los costos de energía. A través de este nivel, se detectan hitos (puntos de dificultad) importantes de ahorro como fugas de energía, mala operación de los equipos o instrumentos, mal funcionamiento de ellos, etcétera.

El nivel "B" proporciona información sobre el consumo de energía por áreas funcionales o procesos específicos de operación. Al nivel "B" se le reconoce como MACRODIAGNÓSTICO y detecta los subsistemas (término que se refiere a dichas áreas o procesos) de mayor desperdicio energético. Este nivel proporciona una idea cuantitativa de los ahorros potenciales de energía y, en general, de las características energéticas de cada subsistema.

El nivel "C", proporciona información precisa y comprensible, de todos y cada uno de los puntos relevantes del diagrama del proceso industrial (entradas y salidas de energía), así como las pérdidas de energía en cada uno de los equipos, a los cuales de aquí en adelante se les referirá como módulos. Se le conoce como MICRODIAGNÓSTICO, siendo el más costoso de los tres niveles, pues permite analizar y detallar todas las pérdidas de energía [2].

El objetivo de los diagnósticos, es transmitir los conocimientos necesarios para identificar y evaluar oportunidades de ahorro de energía en empresas y / o instituciones.

El diagnóstico energético está dirigido a las empresas, industrias e instalaciones públicas y privadas que tengan como objetivo prioritario reducir sus consumos de energía en general, y en particular, a sus gerentes de producción, gerentes de planta, gerentes administrativos, técnicos e ingenieros de operación, responsables del manejo y asignación de los insumos energéticos en los procesos operativos y servicios auxiliares que conforman cualquier planta industrial o instalación administrativa.

El propósito es homogeneizar los criterios de evaluación *económico – energético* que existen actualmente en el mercado [2].

3.2 Análisis del sistema

Para el desarrollo de los formatos de encuesta, primero se ha analizado y definido la estructura de la base de datos que contendrá la información recabada. Además de contener información previamente obtenida.

Dadas las necesidades del sistema se definen las entidades para analizar cómo se conformará la base de datos, que campos tendrá cada tabla y qué relación hay entre las diferentes tablas. Buscando que toda la información útil se incluya, y que además se evite la redundancia de datos.

Después de determinar y construir la base de datos, lo siguiente consiste en programar la aplicación con la que se interactuará con ésta. Para ello se utiliza PHP y HTML para construir las páginas web que capturarán, guardan e invocan los datos.

La intención es disponer de un formato que sea tan parecido como sea posible al que se utiliza en papel, pero con la ventaja de disponer exclusivamente de los campos a utilizar, lo que significa que si algún elemento, componente o artefacto no se encuentra en el espacio encuestado, no se crea en la base de datos, por lo que se evita tener espacios vacíos.

Para las pruebas y la generación del prototipo, se instala y pone en funcionamiento el servidor web Apache, que proveerá las páginas web que contiene el formato de encuesta y muestra la información.

Básicamente, se programa una página de interacción por cada necesidad presentada. Por lo que la parte principal es la programación del formato de encuesta.

Debido a los requerimientos que presenta el sistema, se realiza el análisis de toda la información necesaria para cada formato de encuesta, entonces se identifican las entidades a utilizar y se modula cada una de ellas.

Los módulos resultantes de este análisis son los siguientes:

Tabla 3.1 Módulos Generales del sistema

Módulos Generales del sistema			
ALTAS	BAJAS	ACTUALIZACIÓN	CONSULTAS
En este módulo se encuentran las entidades que hacen referencia al LEDA (Levantamiento de Datos) para las encuestas. Cuenta con el acceso para dar de alta los datos del Encuestador, de la Dependencia, del Edificio, del Administrador, así como el registro de las distintas Encuestas que se realizan para el diagnóstico energético.	Este módulo cuenta con restricciones de acceso, únicamente el administrador puede realizar la baja de datos de cualquier entidad, ya sea Encuestador, Edificio, Dependencia o Encuesta.	La actualización de datos es un módulo muy importante ya que al momento de realizar la consulta de datos, el encargado de realizar el diagnóstico energético, puede determinar si existe o no algún error de captura en las encuestas, y así, con este módulo se tiene la oportunidad de actualizar el o los datos erróneos y poder volver a consultar los datos.	Dependiendo del tipo de usuario que acceso al sistema, el usuario puede realizar la consulta de los formatos que se encuentran en el mismo. Este módulo cuenta con la ventaja de una vez hecha la consulta vía Web, esta misma puede ser vista en formato PDF, según lo requiera el administrador del sistema.

Cada *módulo general* cuenta con las siguientes entidades:

Tabla 3.2 Entidades Necesarias para el LEDA

Entidades que se requieren para el LEDA (Levantamiento de Datos)				
ENCUESTADOR	ENCUESTA	DEPENDENCIA	EDIFICIO	USUARIO

El sistema requiere de restricciones de acceso, por lo que cada módulo tiene ciertos detalles que favorecen a la seguridad del mismo, y se han dividido en dos usuarios de acceso: administrador y encuestador.

En la siguiente tabla se detallan los privilegios de acceso de cada una de las entidades:

Tabla 3.3 Privilegios de acceso de cada Entidad

Entidades	Privilegios de acceso
ENCUESTADOR	El usuario que ingresa al sistema como <i>“encuestador”</i> solo tiene el privilegio de dar de alta los datos de las entidades Encuesta, Dependencia y Edificio, así como también la actualización y consulta de datos de las mismas. Este usuario no puede realizar la baja de ningún tipo de información.
ENCUESTA	No aplica, ya que esta entidad no es un usuario, sino que es parte del LEDA
DEPENDENCIA	No aplica, ya que esta entidad no es un usuario, sino que es parte del LEDA
EDIFICIO	No aplica, ya que esta entidad no es un usuario, sino que es parte del LEDA
USUARIO	El usuario que ingresa al sistema como <i>“administrador”</i> tiene el privilegio de dar de alta los datos de las entidades Encuestador, Encuesta, Dependencia, Edificio y Usuario, así como también la baja, actualización y consulta de datos de las mismas.

Entonces teniendo en cuenta los privilegios de acceso de cada entidad a cada uno de los módulos, se organiza la información obtenida de manera tal que asegure la integridad y seguridad de los datos.

A continuación se detalla cada módulo del sistema al ingresar como **Administrador**:

➤ **Administrador**

❖ **ALTAS**

Tabla 3.4 Módulo Altas (Administrador)

Módulo	Entidad
ALTAS	ENCUESTADOR
Descripción: Esta entidad permite ingresar toda la información necesaria del usuario que realizará la encuesta. Deberá ingresar todos los datos personales que se requieran. Es importante que el usuario cuente con un identificador, como su número de cuenta, número de empleado, etc.	

Módulo	Entidad
ALTAS	DEPENDENCIA
Descripción: Esta entidad contiene la información de la dependencia a la que se le realizará el diagnóstico energético. Se necesitará de un identificador de dependencia para contar con una buena organización de la base de datos, así como conocer nombre de edificios, ubicación en la que se encuentra la dependencia, horas de trabajo y número de empleados.	

Módulo	Entidad
ALTAS	EDIFICIO
Descripción: Para dar de alta a esta entidad es necesario conocer el identificador del edificio, así como su nombre, en que dependencia se encuentra el edificio, número de locales, niveles y la orientación en la que se encuentra el edificio.	

Módulo	Entidad
ALTAS	USUARIO
Descripción: Esta entidad es la que se maneja como Administrador, para dar de alta a este usuario es necesario conocer su identificador de usuario, nombre y proporcionar un <i>login</i> y contraseña para que éste pueda ingresar al sistema.	

Módulo	Entidad
ALTAS	ENCUESTA
Descripción: En esta entidad, se requiere generar un identificador de encuesta. Esta entidad contiene toda la información para poder realizar el análisis del recinto, análisis de iluminación del recinto, fuentes de luz artificial y el tipo de encuesta.	

❖ BAJAS

Tabla 3.5 Módulo Bajas (Administrador)

Módulo	Entidad
BAJAS	ENCUESTADOR
Descripción: Para poder dar de baja a un encuestador, se necesita conocer el identificador de usuario, éste deberá ser único e irrepitable. Para así poder asegurar la integridad de datos del sistema.	

Módulo	Entidad
BAJAS	DEPENDENCIA
Descripción: Para poder dar de baja a una dependencia, se necesita conocer el identificador de dependencia, éste deberá ser único e irrepitable. Para así poder asegurar la integridad de datos del sistema.	

Módulo	Entidad
BAJAS	EDIFICIO
Descripción: Para poder dar de baja a un edificio, se necesita conocer el identificador de edificio, éste deberá ser único e irrepitable. Para así poder asegurar la integridad de datos del sistema.	

Módulo	Entidad
BAJAS	USUARIO
Descripción: Para poder dar de baja a un usuario, se necesita conocer el identificador de usuario, éste deberá ser único e irrepitable. Para así poder asegurar la integridad de datos del sistema.	

Módulo	Entidad
BAJAS	ENCUESTA
Descripción: Para poder dar de baja a una encuesta, se necesita conocer el identificador de encuesta, éste deberá ser único e irrepitable. Para así poder asegurar la integridad de datos del sistema.	

❖ ACTUALIZAR

Tabla 3.6 Módulo Actualizar (Administrador)

Módulo	Entidad
ACTUALIZAR	ENCUESTADOR
Descripción: Este módulo permite actualizar los datos del encuestador en caso de algún error que haya surgido al momento de capturar la información.	

Módulo	Entidad
ACTUALIZAR	DEPENDENCIA
Descripción: Este módulo permite actualizar los datos de la dependencia en caso de algún error que haya surgido al momento de capturar la información.	

Módulo	Entidad
ACTUALIZAR	EDIFICIO
Descripción: Este módulo permite actualizar los datos del edificio en caso de algún error que haya surgido al momento de capturar la información.	

Módulo	Entidad
ACTUALIZAR	USUARIO
Descripción: Este módulo permite actualizar los datos del usuario en caso de algún error que haya surgido al momento de capturar la información.	

Módulo	Entidad
ACTUALIZAR	ENCUESTA
Descripción: Este módulo permite actualizar los datos de la encuesta en caso de algún error que haya surgido al momento de capturar la información.	

❖ CONSULTAS

Tabla 3.7 Módulo Consultas (Administrador)

Módulo	Entidad
CONSULTAS	ENCUESTADOR
<p>Descripción: Se muestra la información detallada del encuestador, es posible delimitar la consulta de datos del encuestador, ya sea por <i>identificador del encuestador</i> o por <i>tipo de usuario</i>, sin embargo también se puede realizar la consulta de todos los encuestadores del sistema.</p>	

Módulo	Entidad
CONSULTAS	DEPENDENCIA
<p>Descripción: Este módulo muestra la información detallada de la dependencia, es posible delimitar la consulta de datos, ya sea por <i>identificador de la dependencia</i> o por <i>ubicación</i>, sin embargo también se puede realizar la consulta de todas las dependencias que fueron dadas de alta en el sistema.</p>	

Módulo	Entidad
CONSULTAS	EDIFICIO
<p>Descripción: Este módulo muestra la información detallada del edificio, es posible delimitar la consulta de datos, ya sea por <i>identificador del edificio</i> o por <i>orientación</i>, sin embargo también se puede realizar la consulta de todos los edificios que se dieron de alta en el sistema.</p>	

Módulo	Entidad
CONSULTAS	USUARIO
<p>Descripción: Con esta entidad se realiza la consulta de todos los usuarios del sistema.</p>	

Módulo	Entidad
CONSULTAS	ENCUESTA
<p>Descripción: La consulta se realiza por <i>identificador de la encuesta</i>, una vez seleccionada la encuesta a consultar es necesario seleccionar que tipo de información se desea visualizar:</p> <ul style="list-style-type: none"> - Análisis del recinto - Análisis de iluminación del recinto - Fuentes de luz artificial - Tipo de encuesta <ul style="list-style-type: none"> ▪ Aire Acondicionado ▪ Taller y/o Laboratorios ▪ Calentadores y/o Bombas ▪ Misceláneos (equipo de cómputo, equipo de papelería, electrodomésticos). 	

A continuación se detalla cada módulo del sistema al ingresar como **Encuestador**:

➤ **Encuestador**

❖ **ALTAS**

Tabla 3.8 Módulo Altas (Encuestador)

Módulo	Entidad
ALTAS	ENCUESTADOR
Descripción: Al ingresar como Encuestador al sistema, se le solicita al usuario dar de alta sus datos para poder continuar con la encuesta. Es importante que el usuario cuente con un identificador, como su número de cuenta, número de empleado, etc.	

Módulo	Entidad
ALTAS	DEPENDENCIA
Descripción: Esta entidad contiene la información de la dependencia a la que se le realizará el diagnóstico energético. Se necesitará de un identificador de dependencia para contar con una buena organización de la base de datos, así como conocer nombre de edificios, ubicación en la que se encuentra la dependencia, horas de trabajo y número de empleados.	

Módulo	Entidad
ALTAS	EDIFICIO
Descripción: Para dar de alta a esta entidad es necesario conocer el identificador del edificio, así como su nombre, en que dependencia se encuentra el edificio, número de locales, niveles y la orientación en la que se encuentra el edificio.	

Módulo	Entidad
ALTAS	USUARIO
Descripción: El usuario que ingresa como <i>encuestador</i> al sistema no tiene el permiso de dar de alta a un usuario, esto es por la integridad de datos que requiere el sistema.	

Módulo	Entidad
ALTAS	ENCUESTA
Descripción: En esta entidad, se requiere generar un identificador de encuesta. Esta entidad contiene toda la información para poder realizar el análisis del recinto, análisis de iluminación del recinto, fuentes de luz artificial y el tipo de encuesta.	

❖ BAJAS

Tabla 3.9 Módulo Bajas (Encuestador)

<p>Descripción: El usuario que ingresa como <i>encuestador al sistema</i>, no cuenta con el permiso de dar de <i>baja</i> a ninguna entidad, asegurando así el control de acceso y la integridad de datos para el correcto funcionamiento del sistema.</p>

❖ ACTUALIZAR

Tabla 3.10 Módulo Actualizar (Encuestador)

Módulo	Entidad
ACTUALIZAR	ENCUESTADOR

<p>Descripción: La información del encuestador no puede ser modificada por el usuario que ingrese como <i>encuestador</i>, con esta limitación se asegura la integridad de datos.</p>
--

Módulo	Entidad
ACTUALIZAR	DEPENDENCIA

<p>Descripción: Este módulo permite actualizar los datos de la dependencia en caso de algún error que haya surgido al momento de capturar la información.</p>
--

Módulo	Entidad
ACTUALIZAR	EDIFICIO

<p>Descripción: Este módulo permite actualizar los datos del edificio en caso de algún error que haya surgido al momento de capturar la información.</p>

Módulo	Entidad
ACTUALIZAR	USUARIO

<p>Descripción: La información del usuario no puede ser modificada por el usuario que ingrese como <i>encuestador</i>, con esta limitación se asegura la integridad de datos.</p>
--

Módulo	Entidad
ACTUALIZAR	ENCUESTA

<p>Descripción: Este módulo permite actualizar los datos de la encuesta en caso de algún error que haya surgido al momento de capturar la información.</p>

❖ CONSULTAS

Tabla 3.11 Módulo Consultas (Encuestador)

Módulo	Entidad
CONSULTAS	ENCUESTADOR
<p>Descripción: Esta entidad no puede ser consultada por el usuario que ingrese como <i>encuestador</i>, con esta limitación se asegura la integridad y confidencialidad de datos registrados por otros encuestadores o administradores.</p>	

Módulo	Entidad
CONSULTAS	DEPENDENCIA
<p>Descripción: Se muestra la información detallada de la dependencia, es posible delimitar la consulta de datos, ya sea por <i>identificador de la dependencia</i> o por <i>ubicación</i>, sin embargo también se puede realizar la consulta de todas las dependencias que fueron dadas de alta en el sistema.</p>	

Módulo	Entidad
CONSULTAS	EDIFICIO
<p>Descripción: Este módulo muestra la información detallada del edificio, es posible delimitar la consulta de datos, ya sea por <i>identificador del edificio</i> o por <i>orientación</i>, sin embargo también se puede realizar la consulta de todos los edificios que se dieron de alta en el sistema.</p>	

Módulo	Entidad
CONSULTAS	USUARIO
<p>Descripción: La información del <i>usuario</i> no puede ser consultada por aquel usuario que ingrese como <i>encuestador</i>, con esta limitación se asegura la integridad y confidencialidad de datos registrados por otros encuestadores o administradores.</p>	

Módulo	Entidad
CONSULTAS	ENCUESTA
<p>Descripción: La consulta se realiza por <i>identificador de la encuesta</i>, una vez seleccionada la encuesta a consultar es necesario seleccionar que tipo de información se desea visualizar:</p> <ul style="list-style-type: none"> - Análisis del recinto - Análisis de iluminación del recinto - Fuentes de luz artificial - Tipo de encuesta <ul style="list-style-type: none"> ▪ Aire Acondicionado ▪ Taller y/o Laboratorios ▪ Calentadores y/o Bombas <p>Misceláneos (equipo de cómputo, equipo de papelería, electrodomésticos).</p>	

3.2.1 Herramientas para el desarrollo y diseño

En el desarrollo de una aplicación de software se utilizan distintas herramientas, como son los lenguajes de programación, o bien, sistemas de administración de bases de datos, servidores, plataformas, sistemas operativos, entre muchas otras.

Existe una gran cantidad y diversidad de dichas herramientas, por lo que para la realización de este proyecto nos apoyaremos con el software libre, ya que posee con características específicas que ofrecen ventajas, ya antes mencionadas al inicio de este capítulo, y que nos brindan grandes oportunidades de desarrollo.

Por lo tanto, para elegir las herramientas que se utilizarán en el desarrollo de este sistema se han considerado los siguientes factores:

- 2) Acuerdo entre cliente y desarrollador para usar software libre
- 3) Propuesta presentada para el proyecto
- 4) Experiencia propia en el uso de las herramientas
- 5) Conocimiento de las herramientas
- 6) Fácil manejo y soporte de las herramientas
- 7) Recursos materiales disponibles
- 8) Accesibilidad y disponibilidad de las herramientas

Las herramientas seleccionadas para desarrollar este proyecto cumplen con los factores antes enlistados, las cuales se describen brevemente a continuación:

- ✓ **Linux** (Descrito en el tema 2.1.4) como sistema operativo

El desarrollo de este proyecto lo llevaremos a cabo bajo la plataforma GNU/Linux, debido a que posee varias características que nos apoyan con la seguridad y fiabilidad del sistema. Entre las varias características que posee Linux contamos que es usado en el segmento de las computadoras de escritorio, portátiles, computadoras de bolsillo, teléfonos móviles, sistemas embebidos, videoconsolas y otros dispositivos. Por lo que hace que nuestro proyecto pueda llegar a ser utilizado en dispositivos móviles portátiles.

Plataforma

Una plataforma es un sistema operativo que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones). Al definir plataformas se establecen los tipos de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario compatibles.

Utilizaremos la distribución Fedora que nos brinda Linux, para la realización del proyecto.

Distribución Fedora

Fedora es una distribución Linux para propósitos generales basada en RPM, que se caracteriza por ser un sistema estable, la cual es mantenida gracias a una comunidad internacional de ingenieros, diseñadores gráficos y usuarios que informan de fallos y prueban nuevas tecnologías.

El proyecto no busca sólo incluir software libre y de código abierto, sino ser el líder en ese ámbito tecnológico. Algo que hay que destacar es que los desarrolladores de Fedora prefieren hacer cambios en las fuentes originales en lugar de aplicar los parches específicos en su distribución, de esta forma se asegura que las actualizaciones estén disponibles para todas las variantes de Linux.

SELinux ("Security-Enhanced Linux") se destaca entre las características de seguridad de Fedora, pues implementa una gran variedad de políticas de seguridad, incluyendo control de acceso obligatorio (MAC "Mandatory Access Control"), a través de los Módulos de Seguridad de Linux que están en el núcleo Linux del sistema.

✓ Apache como servidor Web

Existen múltiples ventajas que este servidor nos ofrece para el desarrollo de este proyecto, ya que Apache es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (HyperText Markup Language). Apache es de código abierto para plataformas Unix, Windows y otras.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la fundación de software de Apache, y presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 68% de los sitios Web en el mundo y creciendo aún su cuota de mercado.

✓ MySQL (Descrito en el tema 2.5.1) como manejador de bases de datos.

Debido a que en el mundo del software libre de los manejadores de bases de datos se tiene de dos opciones viables, MySQL y PostgreSQL, como se mencionaron en el capítulo 2.6.2, se ha elegido a MySQL como manejador, ya que las ventajas que ofrece sobre PostgreSQL específicamente para este proyecto en la cuestión de la compatibilidad y la integridad con PHP, además que ya he interactuado con MySQL como Sistema Manejador de Bases de Datos en ocasiones anteriores.

✓ PHP (Descrito en el tema 2.2) como lenguaje de programación interpretado

Complementamos la programación de este proyecto con PHP ya que es un lenguaje fácil de manipular pues la mayoría de su sintaxis está basada en C, Java y Perl con algunas características propias.

✓ **HTML** (Descrito en el tema 2.2) como lenguaje de marcas

En éste proyecto utilizamos este lenguaje de programación ya que el HTML es un lenguaje de hipertexto que nos sirve para estructurar documentos. Además cabe mencionar que PHP estará embebido en páginas HTML.

Todo este conjunto de herramientas conforma un sistema LAMP, que consiste en un sistema operativo Linux, un servidor web Apache, una base de datos MySQL, y el lenguaje de script PHP. Este sistema logra un excelente funcionamiento, que brinda muchas ventajas porque aumenta la interactividad, velocidad y usabilidad en la aplicación.

CAPÍTULO 4 DISEÑO DEL SISTEMA

Para efectuar un proceso de diagnóstico energético, es necesario recabar datos que sean relevantes para dicho proceso, tarea que siempre resulta ser muy demandante, ya que requiere de un exhaustivo registro de todo aquel equipo que consuma energía; así como otros datos asociados al lugar de trabajo.

Estas actividades se realizan sobre lo que se denomina dependencia, que es un agrupamiento de edificios con una administración principal común. El levantamiento de los datos (LEDA), para las encuestas, se realiza de manera personal, es decir, un conjunto de individuos asisten a cada uno de los espacios de la dependencia a tomar los datos.

Después, los datos de las encuestas se deben pasar a alguna herramienta informática genérica para su procesamiento, análisis y generación de resultados. Tarea que, debido a la ausencia de alguna herramienta más adecuada, se realiza a mano.

El objetivo de este sistema es que se cuente con una aplicación que permita tanto recabar la información como realizar de manera automática el procesamiento, y de ser posible, el análisis de los datos para reducir el tiempo de generación de un diagnóstico energético. El proyecto será accesible desde internet, a través de cualquier navegador, lo que permite usar virtualmente cualquier computadora o dispositivo que cuente con conexión a la red, con esto se propone ofrecer las ventajas de evitar realizar una instalación de software para cada nuevo encuestador, centralizar la captura de los datos y evitar la necesidad de una sincronización con instalaciones individuales, y la posibilidad de realizar el llenado de datos desde cualquier punto. Además de que el proceso evita el tiempo de traslado de los datos de las encuestas al sistema informático, así como la disminución de errores de captura.

A continuación se muestra el diagrama general de nuestro Sistema de administración de bases de datos sobre el comportamiento del consumo de energía en edificios de C.U. :

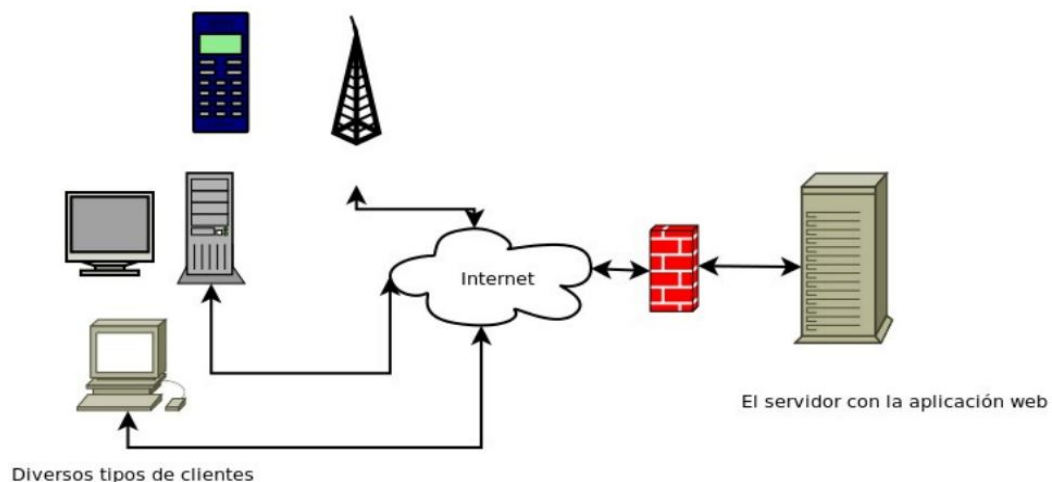


Figura 4.1 Diagrama general del SisFIUNAMAhorroDeEnergía

4.1 Diseño de la base de datos

Las bases para el diseño del sistema se establecieron tomando en cuenta la propuesta que se expuso a detalle en el capítulo III, así como los siguientes aspectos:

- 1) Implementación de módulos.- El sistema consta de una sola base de datos la cual contiene cuatro módulos generales, que son con los que se inicia el levantamiento de datos para la realización del análisis del diagnóstico energético: Alta, Baja, Actualización y Consulta de datos. Dentro de cada módulo tenemos la información del Encuestador, Encuesta, Dependencia, Edificio y Administrador.
- 2) Estructuración de la base de datos.- Para este proyecto es necesario recabar toda la información del personal tanto que está encargado del levantamiento de datos, como la del administrador del sistema. Por lo cual se desarrollan entidades que contienen la información del usuario que realiza la encuesta.
- 3) Petición de usuarios.- El usuario que está encargado de la administración del sistema, es el *administrador*, este usuario cuenta con privilegios de acceso, los cuales se detallaran más adelante. El usuario encargado de la realización de las encuestas, es el *encuestador*, es decir, este usuario se encargará de llenar los formatos que se encuentran en el sistema, recabar la información necesaria para poder efectuar un diagnóstico energético.
- 4) Manejo y experiencia con las bases de datos.- El manejo de las bases de datos consta de un trabajo arduo de investigación y práctica, el primer paso que se debe dar es la cita con el cliente para efectuar el proceso de análisis de los requerimientos y necesidades tanto de hardware como de software, el cual se complementa con la programación y diseño del sistema y la base de datos.

La implementación de los módulos es de suma importancia para la base del diseño, tanto del sistema como para la base de datos, ya que cada uno de los módulos contiene información detallada para que el diagnóstico energético sea efectivo. Es por esta misma razón que cada módulo cuenta con diferente tipo de autenticación de usuario, pues la información que se recaba debe ser verídica y confiable. La consulta de información registrada en cada entidad, no puede ser vista por cualquier usuario, ya que se manejan datos personales y datos precisos del consumo de energía en cada recinto, local, edificio y dependencia.

En base a la información obtenida del análisis, se verifican los módulos necesarios para el diseño del sistema:

Tabla 4.1 Módulos para el diseño

Módulos Generales del sistema			
ALTAS	BAJAS	ACTUALIZACIÓN	CONSULTAS

Tabla 4.2 Entidades necesarias para el diseño

Entidades que se requieren para el LEDA (Levantamiento de Datos)				
ENCUESTADOR	ENCUESTA	DEPENDENCIA	EDIFICIO	USUARIO

La información que se maneja en cada entidad, determina la relación existente en cada encuesta, es decir, en cada uno de los formatos del sistema la información del encuestador, encuesta, dependencia, edificio y usuario, están relacionados entre sí por un identificador de cada entidad, asegurando así la veracidad de datos en el sistema.

A continuación se detalla cada una de las entidades que se han analizado, para asegurar la correcta función de los módulos dentro del sistema:

➤ **Administrador**

❖ **ALTAS**

Tabla 4.3 Diseño del módulo Altas (Administrador)

Módulo	Entidad
ALTAS	ENCUESTADOR
<p>Descripción: Para esta entidad se requieren los siguientes datos:</p> <ul style="list-style-type: none"> - Un identificador de usuario (número de cuenta, número de empleado, etc.) - Apellido paterno, materno y nombres - Tipo de usuario (encuestador o administrador) - Correo electrónico - Número telefónico - Login y contraseña - Tipo de prestador, es decir, que tipo de servicios está prestando para este diagnóstico, puede ser voluntario, trabajador, honorario o si se encuentra realizando su servicio social 	

Módulo	Entidad
ALTAS	DEPENDENCIA
<p>Detalles: En este formulario se solicita llenar los campos proporcionando la siguiente información:</p> <ul style="list-style-type: none"> - Un identificador de la dependencia, que será conformada por las siglas de la dependencia – nombre del edificio – nombre del local - Un identificador del edificio, conformado por el nombre del edificio – nombre del local - Nombre de la dependencia - Nombre del edificio - La ubicación en la que se encuentra, por ejemplo, Ciudad 	

Universitaria
<ul style="list-style-type: none"> - Deberá capturar las horas normales en que la dependencia se encuentra brindando cualquier tipo de atención - La captura de las horas laborales en la dependencia - El horario de comida que se maneja en la dependencia - El número de empleados que se encuentran - Población flotante, es decir, cuanta gente entra y sale de la dependencia y hace uso de equipos consumibles de energía

Módulo	Entidad
ALTAS	EDIFICIO
<p>Detalles: Para esta entidad se solicitan los datos:</p> <ul style="list-style-type: none"> - Un identificador del edificio, conformado por el nombre del edificio – nombre del local - Nombre del edificio - Nombre de la dependencia - Número de locales (salones, laboratorios, salas) - Número de niveles - Nombre del local - Orientación del edificio (norte, sur, este, oeste) 	

Módulo	Entidad
ALTAS	USUARIO
<p>Detalles: Esta entidad es la que se maneja como Administrador, para este usuario se necesitan los siguientes datos:</p> <ul style="list-style-type: none"> - Un identificador de usuario (número de cuenta, número de empleado, etc.) - Nombre - Contraseña - Tipo de usuario (administrador o encuestador) 	

Módulo	Entidad
ALTAS	ENCUESTA
<p>Detalles: Esta entidad sin duda es la que requiere de más formatos, pues es aquí donde se recaba la información del consumo de energía en cada recinto, los datos que se requieren para completar los formatos de la encuesta son:</p> <ul style="list-style-type: none"> - Análisis del recinto - Análisis de iluminación del recinto - Fuentes de luz artificial - Tipo de encuesta 	

En las siguientes tablas se detalla cada uno de los formatos que conforman a la entidad Encuesta y toda la información necesaria para darla de alta.

Análisis del recinto

Detalles: Es toda aquella información general que tiene un recinto:

- Identificador de encuesta, que está conformada por el número de encuesta_siglas de la dependencia
- Identificador de encuestador
- Identificador de la dependencia
- Nombre de la dependencia
- Nombre del edificio
- Orientación
- Nivel
- Identificador del local
- Nombre del local
- Fecha, hora y año de realización de la encuesta
- Estado del clima
- Número de ocupantes del recinto
- Tipo de uso que se le da al recinto (salones, oficinas, taller)
- Edad promedio de los ocupantes del recinto
- Tipo de atmósfera, es decir cuanta cantidad de polvo existe en el recinto
- Tonalidad del muro, techo y piso
- Dimensiones del local en metros (ancho, alto, largo)
- Comentarios acerca del recinto

Análisis de iluminación del recinto

Detalles: La información que se recaba en este formulario es acerca del tipo de luz natural que existe en el recinto:

- Identificador de iluminación, que se conforma de número de encuesta_ilu
- Identificador de encuesta, que está conformada por el número de encuesta_siglas de la dependencia
- Dimensiones del domo o tragaluz (largo, ancho)
- Tipo de acceso de iluminación (unilateral, bilateral, multilateral)
- Tipo de cristales en las ventanas
- Altura al plano de trabajo
- Dimensión y orientación de ventanas (largo, alto, norte, sur)
- Nivel del piso hacia la ventana
- Estado de limpieza de las ventanas
- Altura o distancia entre el tragaluz y el plano de trabajo
- Comentarios acerca de las ventas del recinto

Fuentes de luz artificial

Detalles: La información que se recaba en este formulario es acerca del tipo de luz artificial que existe en el recinto:

- Identificador de iluminación artificial, que se conforma de número de encuesta_artificial
- Identificador de iluminación, que se conforma de número de encuesta_ilu
- Número total de interruptores existentes en el recinto
- Número de luminarias (focos, lámparas) por interruptor
- Horas de uso diarias de la luz en el recinto
- Especificación del tipo de fuentes de luz artificial (lámparas fluorescentes, focos incandescentes, lámparas de vapor, aditivos metálicos, lámparas dicroicas u otras)

Tipo de encuesta

Detalles: En este formulario nos encontramos con la especificación del consumo de equipos que consumen energía eléctrica.

- Aire Acondicionado
- Taller y/o Laboratorios
- Calentadores y/o Bombas
- Misceláneos (equipo de cómputo, equipo de papelería, electrodomésticos)

En cada uno de estos formatos se llenan los campos de información que se requieren para el análisis:

- El identificador de equipo depende si es aire acondicionado, taller, laboratorio, calentador, misceláneo, este identificador es conformado por el número de encuesta_tipo de equipo
- Identificador de encuesta, que está conformada por el número de encuesta_siglas de la dependencia
- Nombre del equipo
- Marca del equipo
- Modelo del equipo
- Capacidad
- Antigüedad
- Potencia (kW)
- Horas, meses, días en que se usa el equipo
- Una descripción breve del equipo
- Cantidad de equipos en el recinto

❖ BAJAS

Tabla 4.4 Diseño del módulo Bajas (Administrador)

Módulo	Entidad
BAJAS	ENCUESTADOR
<p>Detalles: Para dar de baja esta entidad simplemente se requiere:</p> <ul style="list-style-type: none"> - El identificador de usuario (número de cuenta, número de empleado, etc.) <p>*Tomando en cuenta que el identificador de usuario es único e irrepitable</p>	

Módulo	Entidad
BAJAS	DEPENDENCIA
<p>Detalles: Para dar de baja esta entidad simplemente se requiere:</p> <ul style="list-style-type: none"> - El identificador de dependencia que será conformada por las siglas de la dependencia – nombre del edificio – nombre del local <p>*Tomando en cuenta que el identificador de dependencia es único e irrepitable</p>	

Módulo	Entidad
BAJAS	EDIFICIO
<p>Detalles: Para dar de baja esta entidad simplemente se requiere:</p> <ul style="list-style-type: none"> - Un identificador del edificio, conformado por el nombre del edificio – nombre del local <p>*Tomando en cuenta que el identificador de edificio es único e irrepitable</p>	

Módulo	Entidad
BAJAS	USUARIO
<p>Detalles: Para dar de baja esta entidad simplemente se requiere:</p> <ul style="list-style-type: none"> - Un identificador de usuario (número de cuenta, número de empleado, etc.) <p>*Tomando en cuenta que el identificador de usuario es único e irrepitable</p>	

Módulo	Entidad
BAJAS	ENCUESTA
<p>Detalles: Para dar de baja esta entidad simplemente se requiere:</p> <ul style="list-style-type: none"> - Identificador de encuesta, conformado por siglas de la dependencia - nombre del edificio – número de encuesta (tres dígitos) <p>*Tomando en cuenta que el identificador de encuesta es único e irrepitable</p>	

❖ ACTUALIZAR

Tabla 4.5 Diseño del módulo Actualizar (Administrador)

Módulo	Entidad
ACTUALIZAR	ENCUESTADOR
Detalles: Para dar actualizar los datos de esta entidad simplemente se requiere: <ul style="list-style-type: none"> - La información que se desea sustituir del encuestador 	

Módulo	Entidad
ACTUALIZAR	DEPENDENCIA
Detalles: Para dar actualizar los datos de esta entidad simplemente se requiere: <ul style="list-style-type: none"> - La información que se desea sustituir de la dependencia 	

Módulo	Entidad
ACTUALIZAR	EDIFICIO
Detalles: Para dar actualizar los datos de esta entidad simplemente se requiere: <ul style="list-style-type: none"> - La información que se desea sustituir del edificio 	

Módulo	Entidad
ACTUALIZAR	USUARIO
Detalles: Para dar actualizar los datos de esta entidad simplemente se requiere: <ul style="list-style-type: none"> - La información que se desea sustituir del usuario 	

Módulo	Entidad
ACTUALIZAR	ENCUESTA
Detalles: Para dar actualizar los datos de esta entidad simplemente se requiere: <ul style="list-style-type: none"> - La información que se desea sustituir de la encuesta 	

❖ CONSULTAS

Tabla 4.6 Diseño del módulo Consultas (Administrador)

Módulo	Entidad
CONSULTAS	ENCUESTADOR
<p>Detalles: Es posible delimitar la consulta de datos del encuestador y así evitar información no deseada.</p> <p>La consulta puede hacerse por <i>identificador del encuestador</i> o por <i>tipo de usuario</i>, sin embargo también se puede realizar la consulta de todos los encuestadores del sistema, para esto necesario contar con la siguiente información:</p> <ul style="list-style-type: none"> - Un identificador de usuario (número de cuenta, número de empleado, etc.) - Tipo de usuario (encuestador o administrador) 	

Módulo	Entidad
CONSULTAS	DEPENDENCIA
<p>Detalles: Es posible delimitar la consulta de datos de la dependencia y así evitar información no deseada.</p> <p>La consulta puede hacerse por <i>identificador de la dependencia</i> o por <i>ubicación</i>, sin embargo también se puede realizar la consulta de todas las dependencias que fueron dadas de alta en el sistema, para esto necesario contar con la siguiente información:</p> <ul style="list-style-type: none"> - El identificador de dependencia que será conformada por las siglas de la dependencia – nombre del edificio – nombre del local - La ubicación en la que se encuentra, por ejemplo, Ciudad Universitaria 	

Módulo	Entidad
CONSULTAS	EDIFICIO
<p>Detalles: Es posible delimitar la consulta de datos del edificio y así evitar información no deseada.</p> <p>La consulta puede hacerse por <i>identificador del edificio</i> o por <i>orientación</i>, sin embargo también se puede realizar la consulta de todos los edificios que se dieron de alta en el sistema, para esto necesario contar con la siguiente información:</p> <ul style="list-style-type: none"> - Un identificador del edificio, conformado por el nombre del edificio – nombre del local - Orientación del edificio (norte, sur, este, oeste) 	

Módulo	Entidad
CONSULTAS	USUARIO
<p>Detalles: En este módulo y con esta entidad se realiza la consulta de todos los usuarios del sistema.</p>	

Módulo	Entidad
CONSULTAS	ENCUESTA
<p>Detalles: La consulta se realiza por <i>identificador de la encuesta</i>, una vez seleccionada la encuesta a consultar es necesario seleccionar que tipo de información se desea visualizar:</p> <ul style="list-style-type: none"> - Análisis del recinto - Análisis de iluminación del recinto - Fuentes de luz artificial - Tipo de encuesta <ul style="list-style-type: none"> ▪ Aire Acondicionado ▪ Taller y/o Laboratorios ▪ Calentadores y/o Bombas ▪ Misceláneos (equipo de cómputo, equipo de papelería, electrodomésticos) <p>Para esto es necesario proporcionar la siguiente información:</p> <ul style="list-style-type: none"> - Identificador de encuesta, conformado por siglas de la dependencia - nombre del edificio – número de encuesta (tres dígitos) - Identificador de iluminación, que se conforma de número de encuesta_ilu - Identificador de iluminación artificial, que se conforma de número de encuesta_artificial <p>Para el tipo de encuesta es necesario contar con la siguiente información:</p> <ul style="list-style-type: none"> - El identificador de equipo depende si es aire acondicionado, taller, laboratorio, calentador, misceláneo, este identificador es conformado por el número de encuesta_ <i>tipo de equipo</i> 	

Al ingresar como **Encuestador** al sistema, se le solicita al usuario dar de alta sus datos para poder continuar con la encuesta, las acciones que puede realizar este usuario en cada módulo se describen a continuación:

➤ **Encuestador**

❖ **ALTAS**

Tabla 4.7 Diseño del módulo Altas (Encuestador)

Módulo	Entidad
ALTAS	ENCUESTADOR
<p>Detalles: El usuario que ingresa como <i>encuestador</i> al sistema debe proporcionar la siguiente información:</p> <ul style="list-style-type: none"> - Su identificador de usuario (número de cuenta, número de empleado, etc.) 	

- Apellido paterno, materno y nombres
- Tipo de usuario (encuestador o administrador)
- Correo electrónico
- Número telefónico
- Login y contraseña
- Tipo de prestador, es decir, que tipo de servicios está prestando para este diagnóstico, puede ser voluntario, trabajador, honorario o si se encuentra realizando su servicio social

Módulo	Entidad
ALTAS	DEPENDENCIA
<p>Detalles: Para dar de alta a esta entidad, se solicita llenar los campos proporcionando la siguiente información:</p> <ul style="list-style-type: none"> - Un identificador de la dependencia, que será conformada por las siglas de la dependencia – nombre del edificio – nombre del local - Un identificador del edificio, conformado por el nombre del edificio – nombre del local - Nombre de la dependencia - Nombre del edificio - La ubicación en la que se encuentra, por ejemplo, Ciudad Universitaria - Deberá capturar las horas normales en que la dependencia se encuentra brindando cualquier tipo de atención - La captura de las horas laborales en la dependencia - El horario de comida que se maneja en la dependencia - El número de empleados que se encuentran - Población flotante, es decir, cuanta gente entra y sale de la dependencia y hace uso de equipos consumibles de energía 	

Módulo	Entidad
ALTAS	EDIFICIO
<p>Detalles: Para dar de alta a esta entidad se solicitan los datos:</p> <ul style="list-style-type: none"> - Un identificador del edificio, conformado por el nombre del edificio – nombre del local - Nombre del edificio - Nombre de la dependencia - Número de locales (salones, laboratorios, salas) - Número de niveles - Nombre del local - Orientación del edificio (norte, sur, este, oeste) 	

Módulo	Entidad
ALTAS	USUARIO
Detalles: El usuario que ingresa como <i>encuestador</i> al sistema no tiene el permiso de dar de alta a un usuario, esto es por la integridad de datos que requiere el sistema.	

Módulo	Entidad
ALTAS	ENCUESTA
Detalles: Los datos que se requieren para dar de alta la encuesta son: <ul style="list-style-type: none"> - Identificador de encuesta, conformado por siglas de la dependencia - nombre del edificio – número de encuesta (tres dígitos) - Análisis del recinto - Análisis de iluminación del recinto - Fuentes de luz artificial - Tipo de encuesta 	

En las siguientes tablas se detalla cada uno de los formatos que conforman a la entidad **Encuesta** y toda la información necesaria para darla de alta.

Análisis del recinto
Detalles: Es toda aquella información general que tiene un recinto: <ul style="list-style-type: none"> - Identificador de encuesta, que está conformada por el número de encuesta_siglas de la dependencia - Identificador de encuestador - Identificador de la dependencia - Nombre de la dependencia - Nombre del edificio - Orientación - Nivel - Identificador del local - Nombre del local - Fecha, hora y año de realización de la encuesta - Estado del clima - Número de ocupantes del recinto - Tipo de uso que se le da al recinto (salones, oficinas, taller) - Edad promedio de los ocupantes del recinto - Tipo de atmósfera, es decir cuanta cantidad de polvo existe en el recinto - Tonalidad del muro, techo y piso - Dimensiones del local en metros (ancho, alto, largo) - Comentarios acerca del recinto

Análisis de iluminación del recinto

Detalles: La información que se recaba en este formulario es acerca del tipo de luz natural que existe en el recinto:

- Identificador de iluminación, que se conforma de número de encuesta_ilu
- Identificador de encuesta, que está conformada por el número de encuesta_siglas de la dependencia
- Dimensiones del domo o tragaluz (largo, ancho)
- Tipo de acceso de iluminación (unilateral, bilateral, multilateral)
- Tipo de cristales en las ventanas
- Altura al plano de trabajo
- Dimensión y orientación de ventanas (largo, alto, norte, sur)
- Nivel del piso hacia la ventana
- Estado de limpieza de las ventanas
- Altura o distancia entre el tragaluz y el plano de trabajo
- Comentarios acerca de las ventas del recinto

Fuentes de luz artificial

Detalles: La información que se recaba en este formulario es acerca del tipo de luz artificial que existe en el recinto:

- Identificador de iluminación artificial, que se conforma de número de encuesta_artificial
- Identificador de iluminación, que se conforma de número de encuesta_ilu
- Número total de interruptores existentes en el recinto
- Número de luminarias (focos, lámparas) por interruptor
- Horas de uso diarias de la luz en el recinto
- Especificación del tipo de fuentes de luz artificial (lámparas fluorescentes, focos incandescentes, lámparas de vapor, aditivos metálicos, lámparas dicróicas u otras)

Tipo de encuesta

Detalles: En este formulario nos encontramos con la especificación del consumo de equipos que consumen energía eléctrica.

- Aire Acondicionado
- Taller y/o Laboratorios
- Calentadores y/o Bombas
- Misceláneos (equipo de cómputo, equipo de papelería, electrodomésticos)

En cada uno de estos formatos se llenan los campos de información que se requieren para el análisis:

- El identificador de equipo depende si es aire acondicionado,

<p>tallero, laboratorio, calentador, misceláneo, este identificador es conformado por el número de encuesta_ <i>tipo de equipo</i></p> <ul style="list-style-type: none"> - Identificador de encuesta, que está conformada por el número de encuesta_siglas de la dependencia - Nombre del equipo - Marca del equipo - Modelo del equipo - Capacidad - Antigüedad - Potencia (kW) - Horas, meses, días en que se usa el equipo - Una descripción breve del equipo - Cantidad de equipos en el recinto

❖ **BAJAS**

Tabla 4.8 Diseño del módulo Bajas (Encuestador)

<p>Detalles: El usuario que ingresa como <i>encuestador al sistema</i>, no cuenta con el permiso de dar de <i>baja</i> a ninguna entidad, asegurando así el control de acceso y la integridad de datos para el correcto funcionamiento del sistema.</p>
--

❖ **ACTUALIZAR**

Tabla 4.9 Diseño del módulo Actualizar (Encuestador)

Módulo	Entidad
ACTUALIZAR	ENCUESTADOR
<p>Detalles: La información del encuestador no puede ser modificada por el usuario que ingrese como <i>encuestador</i>, con esta limitación se asegura la integridad de datos.</p>	

Módulo	Entidad
ACTUALIZAR	DEPENDENCIA
<p>Detalles: Para dar actualizar los datos de esta entidad simplemente se requiere:</p> <ul style="list-style-type: none"> - La información que se desea sustituir de la dependencia 	

Módulo	Entidad
ACTUALIZAR	EDIFICIO
<p>Detalles: Para dar actualizar los datos de esta entidad simplemente se requiere:</p> <ul style="list-style-type: none"> - La información que se desea sustituir del edificio 	

Módulo	Entidad
ACTUALIZAR	USUARIO
Detalles: La información del usuario no puede ser modificada por el usuario que ingrese como <i>encuestador</i> , con esta limitación se asegura la integridad de datos.	

Módulo	Entidad
ACTUALIZAR	ENCUESTA
Detalles: Para dar actualizar los datos de esta entidad simplemente se requiere: - La información que se desea sustituir de la encuesta	

❖ CONSULTAS

Tabla 4.10 Diseño del módulo Consultas (Encuestador)

Módulo	Entidad
CONSULTAS	ENCUESTADOR
Detalles: La información del <i>encuestador</i> no puede ser consultada por el usuario que ingrese como <i>encuestador</i> , con esta limitación se asegura la integridad y confidencialidad de datos registrados por otros encuestadores o administradores.	

Módulo	Entidad
CONSULTAS	DEPENDENCIA
Detalles: Es posible delimitar la consulta de datos de la dependencia y así evitar información no deseada. La consulta puede hacerse por <i>identificador de la dependencia</i> o por <i>ubicación</i> , sin embargo también se puede realizar la consulta de todas las dependencias que fueron dadas de alta en el sistema, para esto necesario contar con la siguiente información: - El identificador de dependencia que será conformada por las siglas de la dependencia – nombre del edificio – nombre del local - La ubicación en la que se encuentra, por ejemplo, Ciudad Universitaria	

Módulo	Entidad
CONSULTAS	EDIFICIO
Detalles: Es posible delimitar la consulta de datos del edificio y así evitar información no deseada. La consulta puede hacerse por <i>identificador del edificio</i> o por <i>orientación</i> , sin embargo también se puede realizar la consulta de todos los edificios que se dieron de alta en el sistema, para esto necesario contar con la siguiente información:	

<ul style="list-style-type: none"> - Un identificador del edificio, conformado por el nombre del edificio – nombre del local - Orientación del edificio (norte, sur, este, oeste)

Módulo	Entidad
CONSULTAS	USUARIO
<p>Detalles: La información del <i>usuario</i> no puede ser consultada por aquel usuario que ingrese como <i>encuestador</i>, con esta limitación se asegura la integridad y confidencialidad de datos registrados por otros encuestadores o administradores.</p>	

Módulo	Entidad
CONSULTAS	ENCUESTA
<p>Detalles: La consulta se realiza por <i>identificador de la encuesta</i>, una vez seleccionada la encuesta a consultar es necesario seleccionar que tipo de información se desea visualizar:</p> <ul style="list-style-type: none"> - Análisis del recinto - Análisis de iluminación del recinto - Fuentes de luz artificial - Tipo de encuesta <ul style="list-style-type: none"> ▪ Aire Acondicionado ▪ Taller y/o Laboratorios ▪ Calentadores y/o Bombas ▪ Misceláneos (equipo de cómputo, equipo de papelería, electrodomésticos) <p>Para esto es necesario proporcionar la siguiente información:</p> <ul style="list-style-type: none"> - Identificador de encuesta, conformado por siglas de la dependencia - nombre del edificio – número de encuesta (tres dígitos) - Identificador de iluminación, que se conforma de número de encuesta_ilu - Identificador de iluminación artificial, que se conforma de número de encuesta_artificial <p>Para el tipo de encuesta es necesario contar con la siguiente información:</p> <ul style="list-style-type: none"> - El identificador de equipo depende si es aire acondicionado, taller, laboratorio, calentador, misceláneo, este identificador es conformado por el número de encuesta_ <i>tipo de equipo</i> 	

Cada una de las entidades muestra información relevante que permite realizar un diagnóstico energético, sin embargo, mucha de esta información se encuentra repetida en cada uno de los formatos que se utilizan para el LEDA (Levantamiento de Datos). Esto afecta el funcionamiento del sistema, haciendo las búsquedas muy lentas y con la gran posibilidad de que al consultar los datos, el sistema muestre datos erróneos.

Es por eso que el siguiente paso para obtener un sistema de administración de bases de datos eficaz y confiable es realizar la normalización de información.

4.2 Normalización de la base de datos

Actualmente el levantamiento de datos (LEDA) para las encuestas se realiza en papel para después ser almacenado en hojas de Excel y así poder efectuar el diagnóstico energético. El formato de las encuestas contiene mucha información redundante (como se pudo observar el tema anterior), es por eso que para el sistema se realizará la normalización de estos datos y así evitar, como ya se señaló en el **tema 2.4.3.3**, anomalías al efectuar altas, bajas o actualizaciones de información.

A continuación se presentan los formatos que se utilizan hoy en día para el LEDA y la normalización de cada formato de encuesta, para llegar a establecer un diseño confiable del sistema y generar una base de datos sin información redundante, sin anomalías y efectuar la consulta de datos más rápida.

FORMATOS QUE SE UTILIZAN ACTUALMENTE PARA EL LEDA

El encargado del análisis del diagnóstico energético proporciona al encuestador un paquete de 5 hojas para el levantamiento de datos, cada hoja consta de un formato de encuesta que se realiza solo para un recinto, tomando en cuenta que cada dependencia está conformada por varios edificios y cada edificio cuenta con hasta cientos de recintos, es por esto que la realización de encuestas se vuelve poco tediosa, pues para poder realizar la encuesta en tan solo un edificio se requieren cientos de hojas.

La captura de información, como se mencionó con anterioridad, se realiza a mano, entonces se cuenta con un alto margen de error, pues muchas veces se suelen confundir números con letras y viceversa. En cada uno de los formatos de captura es necesario repetir información, como por ejemplo: la fecha de realización de la encuesta, el nombre del responsable de la encuesta, el nombre del edificio, la dependencia y la hora de elaboración del análisis, entre muchos otros datos redundantes.

A continuación se presentan los formatos de captura que se utilizan, el detalle de cada uno de ellos y enseguida las tablas que le corresponden, así como la normalización de información de cada tabla obtenida.

El primer formato consta de tres secciones:

- Datos generales de encuesta: El encuestador comienza la encuesta llenando los datos del formulario donde se solicita la información general del mismo, del edificio, de la dependencia, así como también la hora de elaboración del análisis.

- Datos generales del recinto: En esta misma encuesta se solicitan los datos generales del recinto, uso predominante, número de personas en el recinto, edad promedio de los ocupantes, dimensiones del local y comentarios que se puedan hacer acerca del recinto en cuestión.
- Datos de iluminación del recinto: También se requieren los datos de la iluminación tanto natural como artificial, número de interruptores, número de ventanas, orientación de ventanas, tipo de lámparas, tipo de focos, y horas de uso diarias del recinto.

En la siguiente figura se muestran las tres diferentes secciones de la encuesta, remarcadas con un recuadro de color:

- Datos generales de encuesta
- Datos generales del recinto
- Datos de iluminación del recinto

Fecha:	Responsable de la encuesta:	No.				
Empresa						
Edificio:	Nivel:					
Identificación del local:						
Hora de elaboración del análisis:						
Generalidades:						
1. Uso preponderante del recinto						
Aula ()	Oficina ()	Taller () Laboratorio () Servicios ()				
2. Número de personas en el recinto:						
3. Edad promedio de los ocupantes						
Menos de 40 ()	40 a 55 ()	Mayores a 55 ()				
4. Dimensiones del local (en metros)						
L =	A =	H =				
5. Altura de la fuente luminosa al plano de trabajo: (Estimada en m)						
Comentarios:						
Iluminación Artificial						
16. Número total de interruptores						
17. Número de fuentes luminosas por interruptor:						
18. Horas de uso diarias del recinto:						
19. Especificación de fuentes de luz artificial						
Tipo de Luminario	Instaladas	Útiles	L-V (h/d)	S (h/d)	D(h/d)	Observaciones
Fluorescente lineal de 4 x 40						
Fluorescente lineal de 2 x 40						
Fluorescente lineal de 2 x 75						
Fluorescente lineal de 1 x 75						
Fluorescente lineal de 1 x 40						
Fluorescente lineal de 4 x 20						
Fluorescente compacta de 1 x 18						
Fluorescente compacta de 2 x 18						
Focos Incandescentes 100 W						
Focos Incandescentes 75 W						
Focos Incandescentes 60 W						
Vapor de Mercurio 400 W						
Vapor de Mercurio 250 W						
Vapor de Sodio de Alta Presión 400						
Vapor de Sodio de Alta Presión 250						
Vapor de Sodio de Baja Presión 180						
Aditivos Metálicos 400 W						
Aditivos Metálicos 250 W						
Lámparas dicróicas 50 W						
Otro: especificar						

Figura 4.2 Primer formato de captura (Datos generales de encuesta)

Tabla 4.11 Tablas correspondientes al primer formato de captura

ENCUESTA	GENERALIDADES	ILUMINACIÓN
Fecha	Uso recinto	Numero interruptores
Responsable de encuesta	Personas en recinto	Fuentes luminosas
Número de encuesta	Edad promedio	Horas de uso
Empresa	Dimensiones	Luz artificial
Edificio	Altura del local	Tipo luminario
Nivel	Comentarios	
Identificación del local		
Hora de elaboración		

El segundo formato de captura, consta de dos secciones:

- Datos generales de encuesta: El encuestador comienza la encuesta llenando los datos del formulario donde se solicita la información general del mismo, del edificio, de la dependencia, así como también la hora de elaboración del análisis. **En este formato la información de datos generales de encuesta es redundante.**
- Misceláneos: Esta sección maneja demasiada información, por lo que para generar una consulta rápida, es necesario dividir la información en tres tablas: **equipo de cómputo, equipo de papelería y electrodomésticos.**

Las tablas que corresponden a este segundo formato de captura, sin aplicar normalización son dos:

Tabla 4.12 Tablas correspondientes al segundo formato de captura

GENERALIDADES	MISCELÁNEOS
Uso recinto	eq_compu
Personas en recinto	eq_papel
Edad promedio	electrodomésticos
Dimensiones	
Altura del local	
Comentarios	

● Datos generales de encuesta

● Misceláneos

Fecha:	Responsable de la encuesta:	No.				
Empresa						
Edificio:	Nivel:					
Identificación del local:						
Hora de elaboración del análisis:						
Misceláneos y cómputo						
Misceláneos y cómputo	Potencia (W)	Cantidad	L-V (h/d)	S (h/d)	D(h/d)	Observaciones
Computadora escritorio (CPU y monitor)						
Computadora portátil						
Impresora laser						
Impresora inyección						
Impresora multifuncional						
Scanner						
Plotter						
Regulador						
No-break						
cafetera						
calefactor						
despachador de café						
despachador de golosinas						
despachador de refrescos						
destructor de papel						
enfriador-calentador						
equipo de video						
estereo						
fax						
fotocopiadora						
grabadora						
guillotina						
horno de microondas						
máquina de escribir						
parrila eléctrica						
proyector (cañón)						
radio						
radio grabadora						
refrigerador						
reproductor de DVD						
sacapuntas						
secadora de manos						
sumadora						
teléfono inalámbrico						
televisor						
tostador						
triturador						
unidad de CD-RW						
vaporizador						
ventilador						
video VHS						
otro, especificar						

Figura 4.3 Segundo formato de captura (Misceláneos)

El tercer formato de captura, consta de dos secciones:

- Datos generales de encuesta: El encuestador comienza la encuesta llenando los datos del formulario donde se solicita la información general del mismo, del edificio, de la dependencia, así como también la hora de elaboración del análisis. **En este formato la información de datos generales de encuesta es redundante.**

- Equipo de bombeo y/o calefactores: Esta sección maneja la información detallada de las características de consumo energético de un equipo de bombeo y/o calefactores.

● Datos generales de encuesta

● Equipo de Bombeo y/o calefactores

FORMATO DE CAPTURA EQUIPOS DE BOMBEO Y/O CALEFACTORES									
FECHA									
DEPENDENCIA									
ENCUESTADOR									
Edificio	Nivel	Nombre del recinto	Marca	Potencia demandada (HP)	Potencia demandada (kW)	meses de uso/año	hora/día	kWh/mes	Comentarios

Figura 4.5 Cuarto formato de captura (Bombeo y/o Calefactores)

Las tablas que corresponden a este cuarto formato de captura, sin aplicar normalización son dos:

Tabla 4.14 Tablas correspondientes al cuarto formato de captura

GENERALIDADES
Uso recinto
Personas en recinto
Edad promedio
Dimensiones
Altura del local
Comentarios

CAL_BOMBEO
Marca
Pot_dem
Pot_demk
Mes_uso
Hr_día
Kw_ohm
comm

El quinto formato de captura, consta de dos secciones:

- Datos generales de encuesta: El encuestador comienza la encuesta llenando los datos del formulario donde se solicita la información general del mismo, del edificio, de la dependencia, así como también la hora de elaboración del análisis. **En este formato la información de datos generales de encuesta es redundante.**
- Equipo de taller o laboratorio: Esta sección maneja la información detallada de las características de consumo energético de un equipo de taller o laboratorio.

Comenzando con el proceso de normalización para cada módulo.

Ahora se aplica el proceso de las *formas normales*, mencionado en el **tema 2.4.3.3**, eliminando datos que se repiten en cada formato de las encuestas y se agregan a una nueva tabla, creando así las tablas con las que se diseñará el diagrama E-R con sus llaves primarias y llaves foráneas.

Se cuentan con las siguientes tablas:

Tabla 4.16 Tablas resultantes del análisis de requerimientos

ENCUESTA	GENERALIDADES	ILUMINACIÓN	MISCELÁNEOS
Fecha	Uso recinto	Numero interruptores	eq_compu
Responsable de encuesta	Personas en recinto	Fuentes luminosas	eq_papel
Número de encuesta	Edad promedio	Horas de uso	electrodomésticos
Empresa	Dimensiones	Luz artificial	
Edificio	Altura del local	Tipo luminario	
Nivel	Comentarios	llu_natural	
Identificación del local			
Hora de elaboración			

TALLER_LAB	AIRE_ACOND	CAL_BOMBEO
Equipo	Equipo	Marca
Cantidad	Descripción	Pot_dem
Potencia	Cantidad	Pot_demk
Hora_dia	Antigüedad	Mes_uso
Kw	Capacidad TR	Hr_dia
Dia_mes	Marca	Kw_ohm
Mes_año	Modelo	comm
Dia_op	REE	
Hr_op	Pot_maxima	
Hr_dis_distr	Horas/días	
Fc	Meses al año	
Kw_demand	Días/meses	
Kwh_mes		
comm		

Primera forma normal (1FN) para la tabla Encuesta.

Una tabla está en 1FN si sus atributos contienen valores atómicos, es decir solo puede contener un solo valor.

La tabla **encuesta** contiene atributos en los cuales pueden contener más de un solo valor, por lo que se viola la 1FN.

Tabla 4.17 Tabla Encuesta

ENCUESTA
Fecha
Responsable de encuesta
Número de encuesta
Empresa
Edificio
Nivel
Identificación del local
Hora de elaboración

Solución: Se duplican los registros con valores repetidos y se separa el atributo que viola 1FN en una tabla. Se crean llaves primarias.

De esta tabla se separan los atributos: *responsable de encuesta*, *empresa*, *edificio*, *identificación del local* y *nivel*. Por lo tanto se generan nuevas tablas: Encuesta_recinto, Dependencia, Edificio y Encuestador.

Entonces las tablas quedan de la siguiente forma:

Tabla 4.18 1FN de la Tabla Encuesta

ENCUESTA_RECINTO
Id_enc
Fecha
Número de encuesta
Hora de elaboración
Identificación del local

EDIFICIO
Id_edif
Edificio
Nivel

DEPENDENCIA
Id_dep
Nom_dep

ENCUESTADOR
Id_encuestador
Responsable de encuesta

Segunda forma normal (2FN) para la tabla Encuesta.

Se crea una nueva relación con los atributos que tienen dependencia incompleta:

Tabla 4.19 2FN de la Tabla Encuesta

ENCUESTA_RECINTO	EDIFICIO	DEPENDENCIA	ENCUESTADOR
Id_enc	Id_edif	Id_dep	Id_encuestador
Id_dep	Edificio	Id_edif	Responsable de encuesta
Id_encuestador	Nivel	nom_dep	
Fecha			
Número de encuesta			
Hora de elaboración			
Identificación del local			

Primera forma normal (1FN) para la tabla Generalidades.

La tabla **generalidades** es redundante para formato de captura, es por esta razón que aplicando la 1FN, los atributos que violan la 1FN se separan. En esta ocasión ya no es necesario crear otra tabla, simplemente se agregan los atributos a las tablas antes creadas, asegurando que no se viole la 1FN

Tabla 4.20 Tabla Generalidades

GENERALIDADES
Uso recinto
Personas en recinto
Edad promedio
Dimensiones
Altura del local
Comentarios

Por lo tanto, los atributos: *uso recinto*, *edad promedio*, *dimensiones*, *altura del local* y *comentarios*, se agregan a la tabla **encuesta_recinto**, y el atributo: *personas en recinto*, lo agregamos a la tabla **dependencia** con un nombre de atributo más específico *num_empleados*. Se crean llaves primarias.

Quedando así las tablas:

Tabla 4.21 1FN de la Tabla Generalidades

ENCUESTA_RECINTO	DEPENDENCIA	EDIFICIO	ENCUESTADOR
Id_enc	Id_dep	Id_edif	Id_encuestador
Id_dep	Id_edif	Edificio	Responsable de encuesta
Id_encuestador	nom_dep	Nivel	
Fecha	Num_empleados		
Número de encuesta			
Hora de elaboración			
Identificación del local			
Usore			
Edad_prom			
Dimensiones			
Altura_local			
comentarios			

Primera forma normal (1FN) para la tabla Iluminación.

La tabla **iluminación** contiene atributos donde puede contener más de un valor, por lo que viola la 1FN, es necesario identificar el tipo de iluminación en el recinto, ya que para el diagnóstico energético, se analiza la iluminación artificial y natural.

Tabla 4.22 Tabla Iluminación

ILUMINACIÓN
Numero interruptores
Fuentes luminosas
Horas de uso
Luz artificial
Tipo luminario
llu_natural

Los atributos: *número interruptores, fuentes luminosas, horas de uso y luz artificial* se separan de la tabla *iluminación* para crear la tabla **encuesta_ilu_artificial** y la tabla **encuesta_iluminacion**. Se crean llaves primarias.

Por lo tanto las tablas quedan de la siguiente manera:

Tabla 4.23 1FN de la Tabla Iluminación

ENCUESTA_ILUMINACION
Id_ilu
llu_natural

ENCUESTA_ILU_ARTIFICIAL
Id_ilu_art
Num_tot_int
Luminter
Hr_usod

Segunda forma normal (2FN) para la tabla Iluminación.

Se crea una nueva relación con los atributos que tienen dependencia incompleta:

Tabla 4.24 2FN de la Tabla Iluminación

ENCUESTA_ILUMINACION
Id_ilu
Id_enc
llu_natural

ENCUESTA_ILU_ARTIFICIAL
Id_ilu_art
Id_ilu
Num_tot_int
Luminter
Hr_usod

Para las tablas **misceláneos**, **aire acondicionado**, **taller_laboratorio** y **cal_bombeo**, se analiza hasta la 2FN. En la **1FN** se agregan llaves primarias y para la **2FN** se crean relaciones con los atributos incompletos. Quedando entonces las tablas con la siguiente estructura:

Tabla 4.25 Las tablas misceláneos, aire_acondicionado, taller_laboratorio, cal_bombeo Normalizadas

TALLER_LABORATORIO
Id_taller
Id_encuestador
Equipo
Cantidad
Potencia
Hora_dia
Kw
Dia_mes
Mes_año
Dia_op
Hr_op
Hr_dis_distr
Fc
Kw_demand
Kwh_mes
comm

AIRE_ACONDICIONADO
Id_aire
Id_encuestador
Equipo
Descripción
Cantidad
Antigüedad
Capacidad TR
Marca
Modelo
REE
Pot_maxima
Horas/días
Meses al año
Días/meses

CAL_BOMBEO
Id_bom
Id_encuestador
Marca
Pot_dem
Pot_demk
Mes_uso
Hr_dia
Kw_ohm
comm

MISCELÁNEOS
Id_misc
Id_enc
eq_compu
eq_papel
electrodomésticos

Cada uno de estas entidades y atributos se realizaron en base a los formatos de captura proporcionados por el cliente, sin embargo, al momento de hacer la revisión de información, el cliente sugirió integrar más datos en cada una de las tablas para poder efectuar el diagnóstico energético más eficaz.

A continuación las tablas que se obtuvieron después de aplicar la normalización de los formatos de encuesta:

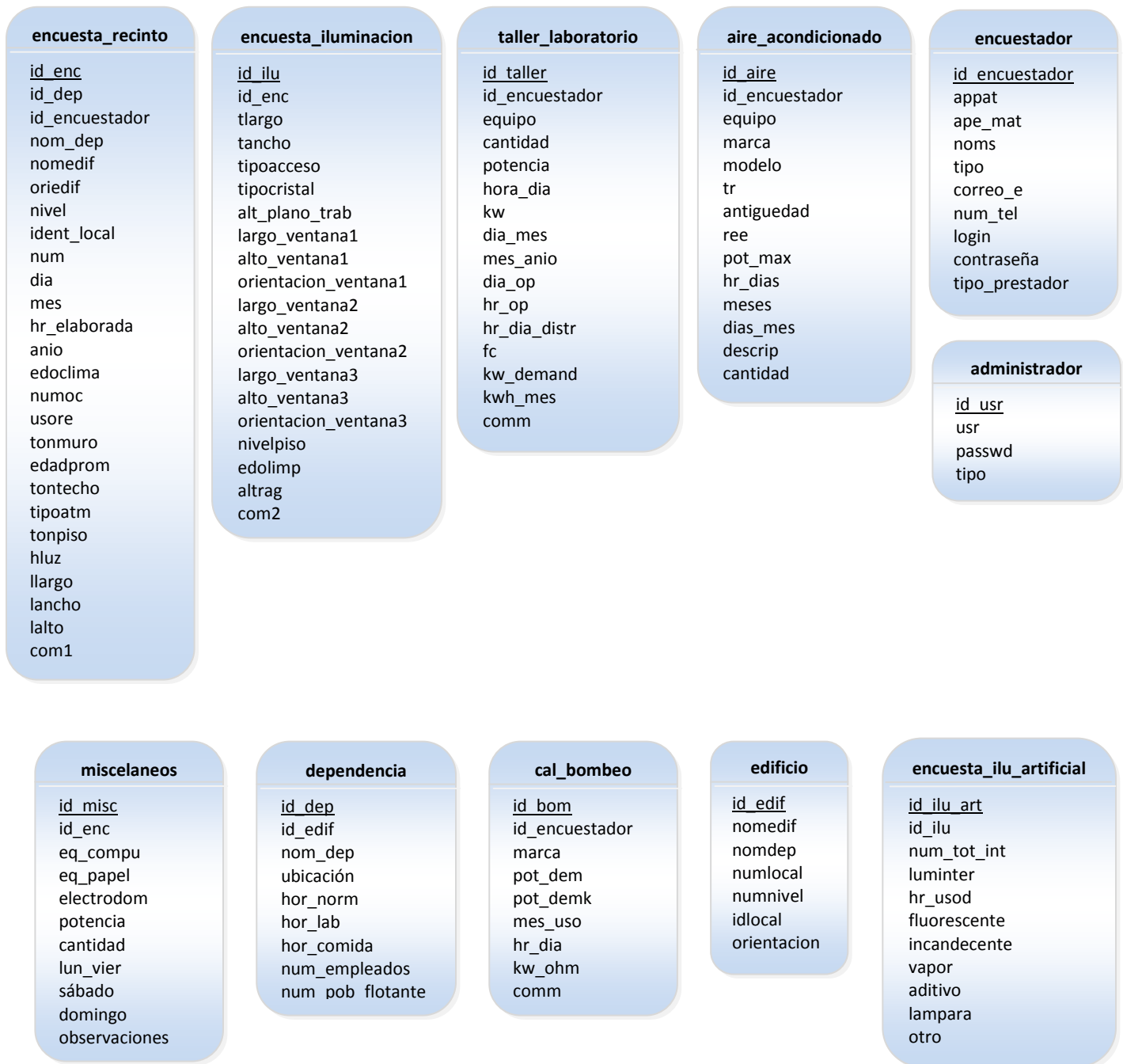


Figura 4.7 Tablas obtenidas después de la Normalización

4.2.1 Modelo Entidad – Relación de cada módulo

Para comprender mejor el uso de las tablas mencionadas en el capítulo anterior y con que otros módulos se relacionan, es importante conocer el Modelo Entidad-Relación de cada uno de ellos. Nos basamos en este modelo ya que nos permiten representar las entidades relevantes del sistema de información así como sus interrelaciones y propiedades.

Un modelo E-R, consta de una colección de objetos básicos, llamados entidades, atributos y de relaciones entre esos objetos.

Entidad: Se puede definir como Entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar información.

Entidad = TABLA

Atributos: Los Atributos son características o propiedades asociadas a la entidad que toman valor en una instancia particular. Ejemplo: nombre, cédula, teléfono.

Atributos = CAMPOS

Relación: Se entiende por Relación a la asociación entre 2 o más entidades.

Tipos de Relaciones (Cardinalidad):

Relación Uno a Uno: Cuando un registro de una tabla sólo puede estar relacionado con un único registro de la otra tabla y viceversa.

Uno a Uno

1 1

Relación Uno a Muchos: Cuando un registro de una tabla (tabla secundaria) sólo puede estar relacionado con un único registro de la otra tabla (tabla principal) y un registro de la tabla principal puede tener más de un registro relacionado en la tabla secundaria.

Uno a Muchos

1 ∞

Relación Muchos a Muchos: Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos (Tabla débil o de vinculación) que incluya los pares de valores relacionados entre sí.

Muchos a Muchos

∞ ∞

Figura 4.8 Tipo de Relaciones en un MER

El nombre de tabla débil deviene que con sus atributos propios no se puede encontrar la clave, por estar asociada a otra entidad.

El sistema se encuentra soportado por una base de datos, la cual se planteó bajo el modelo relacional, por lo que a continuación se presenta la estructura lógica general de la base de datos, es decir, el modelo entidad-relación, el cual es necesario para poder obtener las tablas correspondientes al modelo relacional:

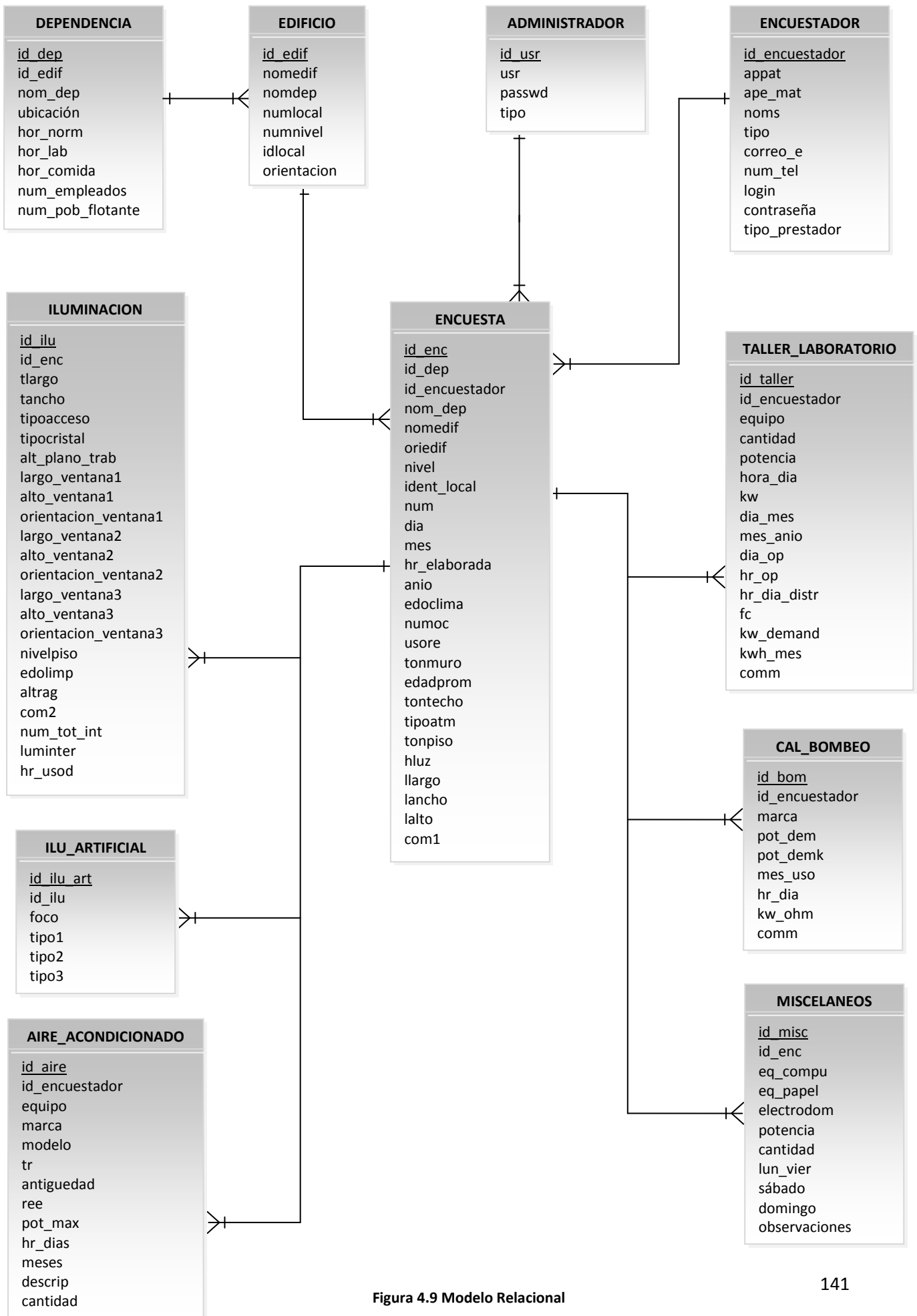


Figura 4.9 Modelo Relacional

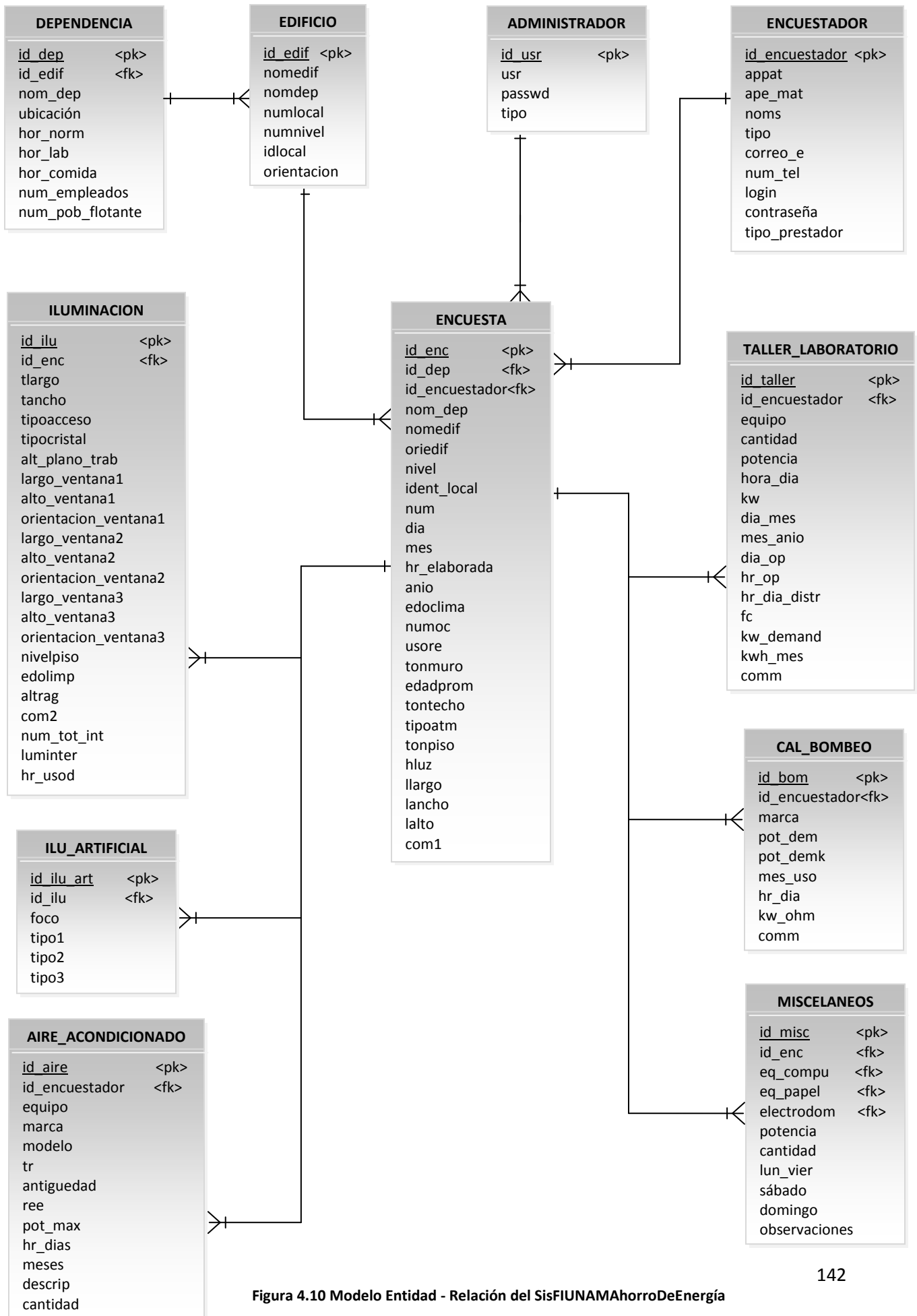


Figura 4.10 Modelo Entidad - Relación del SisFIUNAMAhorroDeEnergía

4.3 Diccionario de datos

Un SGBD debe proporcionar un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios. Este catálogo es lo que se denomina **diccionario de datos** y contiene información que describe los datos de la base de datos (*metadatos*). Normalmente, un diccionario de datos almacena: Nombre, tipo y tamaño de los datos. Nombre de las relaciones entre los datos, restricciones de integridad sobre los datos, nombre de los usuarios autorizados a acceder a la base de datos, esquema externo, conceptual e interno, y correspondencia entre los esquemas. Esto ayuda a mantener el control sobre los datos.

Las redundancias y las inconsistencias se pueden identificar más fácilmente ya que los datos están centralizados. Se puede tener un historial de los cambios realizados sobre la base de datos. El impacto que puede producir un cambio se puede determinar antes de que sea implementado, ya que el diccionario de datos mantiene información sobre cada tipo de dato, todas sus relaciones y todos sus usuarios. Se puede hacer respetar la seguridad, se garantiza la integridad y se puede proporcionar información para auditorías.

Las tablas implementadas para el desarrollo del sistema se basaron en el modelo entidad-relación obtenido en el **tema 4.2.1**, en las cuales los valores de longitud y tipo de atributo son directamente correspondientes con la información que se requiere para el análisis del diagnóstico energético.

A continuación se realizará la descripción e implementación del diccionario de datos para el SisFIUnamAhorroDeEnergía.

Tabla encuesta_recinto - tiene relación con las tablas: edificio, administrador, encuestador, encuesta_iluminación, encuesta_ilu_artificial, aire_acondicionado, misceláneos, cal_bombeo y taller_laboratorio.

Tabla 4.26 Diccionario de Datos de la Tabla encuesta_recinto

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
id_enc	Identificador de encuesta	varchar	30	not_null	✓	
id_dep	Identificador de dependencia	varchar	30	not_null		✓
id_encuestador	Identificador de encuestador	varchar	10	not_null		✓
nom_dep	Nombre de la dependencia	varchar	30	null		
nom_edif	Nombre del edificio	varchar	10	null		
oriedif	Orientación del edificio	varchar	10	null		
nivel	Nivel del recinto	varchar	10	null		
ident_local	Identificador del recinto	varchar	10	null		
num	Número del recinto	varchar	5	null		
dia	Día en que se realizó la encuesta	varchar	2	null		
mes	Hora en que se realizó la encuesta	varchar	15	null		

hr_elaborada	Hora en que se realizó la encuesta	date		null		
anio	Año en que se realizó la encuesta	varchar	4	null		
edoclima	Estado del clima mientras se realiza la encuesta	varchar	15	null		
numoc	Número de ocupantes del recinto	int	3	null		
usore	Uso que se le da al recinto	varchar	10	null		
tonmuro	Tonalidad del muro	varchar	10	null		
edadprom	Edad promedio de los ocupantes del recinto	int	2	null		
tontecho	Tonalidad del techo	varchar	10	null		
tipoatm	Tipo de atmosfera que se encuentra mientras se realiza la encuesta	varchar	15	null		
tonpiso	Tonalidad del piso del recinto	varchar	10	null		
hluz	Altura de la fuente luminosa al plano de trabajo	int	5	null		
llargo	Dimensiones del local en metros (largo)	int	5	null		
lancho	Dimensiones del local en metros (ancho)	int	5	null		
lalto	Dimensiones del local en metros (alto)	int	5	null		
com1	Comentarios sobre el recinto	varchar	30	null		

Tabla encuesta_ iluminación – tiene relación con la tabla: encuesta_recinto

Tabla 4.27 Diccionario de Datos de la Tabla encuesta_ iluminación

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
id_ilu	Identificador de encuesta	varchar	20	not_null	✓	
id_enc	Identificador de dependencia	varchar	30	not_null		✓
tlargo	Dimensión de la lámpara (largo)	int	6	null		
tancho	Dimensión de la lámpara (ancho)	int	6	null		
tipoacceso	Tipo de acceso de iluminación	varchar	15	null		
tipocristal	Tipo de cristal de la ventana	varchar	15	null		
alt_plano_trab	Altura al plano de trabajo	int	6	null		
largo_ventana1	Dimensión y orientación de las ventanas	int	6	null		
alto_ventana1	Dimensión y orientación de las ventanas	int	6	null		
orientacion_ventana1	Dimensión y orientación de las ventanas	varchar	15	null		
largo_ventana2	Dimensión y orientación de las ventanas	int	6	null		
alto_ventana2	Dimensión y orientación de las ventanas	int	6	null		

orientacion_ventana2	Dimensión y orientación de las ventanas	varchar	15	null		
largo_ventana3	Dimensión y orientación de las ventanas	int	6	null		
alto_ventana3	Dimensión y orientación de las ventanas	int	6	null		
orientacion_ventana3	Dimensión y orientación de las ventanas	varchar	15	null		
nivelpiso	Nivel de piso a ventana	int	6	null		
edolimp	Estado de limpieza del recinto	varchar	10	null		
altrag	Altura o distancia entre el tragaluz y el plano de trabajo	int	6	null		
com2	Comentarios	varchar	30	null		

Tabla taller_laboratorio – tiene relación con la tabla: encuesta_recinto

Tabla 4.28 Diccionario de Datos de la Tabla taller_laboratorio

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
id_taller	Identificador del taller	varchar	10	not_null	✓	
id_encuestador	Identificador de encuestador	varchar	10	not_null		✓
equipo	Equipo de laboratorio o taller	varchar	10	null		
cantidad	Cantidad de equipos	int	5	null		
potencia	Potencia que genera el equipo	int	5	null		
hora_dia	Hora y día en que se realiza la encuesta	datetime		null		
kw	Kilowatts instalados	int	5	null		
dia_mes	Kilowatts por día y mes	datetime		null		
mes_anio	Kilowatts por mes y año	date		null		
dia_op	Kilowatts por día y año de operación	date		null		
hr_op	Kilowatts por hora y año de operación	time		null		
hr_dia_distr	Kilowatts por hora y día distribuidos	datetime		null		
fc	Factor de carga	int	5	null		
kw_deman	Kilowatts demandados	int	5	null		
kwh_mes	Kilowatts por mes	int	5	null		
comm	comentarios	varchar	30	null		

Tabla aire_acondicionado – tiene relación con la tabla: encuesta_recinto

Tabla 4.29 Diccionario de Datos de la Tabla aire_acondicionado

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
-----------	-------------	------	----------	-------	----------------	---------------

<u>id_aire</u>	Identificador de encuesta	varchar	10	not_null	✓	
id_encuestador	Identificador de encuestador	varchar	10	not_null		✓
equipo	Nombre de equipo	varchar	15	null		
marca	Marca del equipo	varchar	30	null		
modelo	Modelo del equipo	varchar	30	null		
tr	Tonelada de refrigeración	int	5	null		
antigüedad	Antigüedad del equipo	int	5	null		
ree	Relación de Eficiencia Energética	int	5	null		
pot_max	Potencia máxima del equipo	int	5	null		
hr_dias	Horas y días en uso del equipo	datetime		null		
meses	Meses que ha sido usado el equipo	varchar	10	null		
descrip	Descripción del equipo	varchar	30	null		
cantidad	Cantidad de equipos en el recinto	varchar	5	null		

Tabla encuestador – tiene relación con la tabla: encuesta_recinto

Tabla 4.30 Diccionario de Datos de la Tabla encuestador

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
<u>id_encuestador</u>	Identificador de encuestador	varchar	10	not_null	✓	
appat	Apellido paterno del encuestador	char	20	null		
ape_mat	Apellido materno del encuestador	char	20	null		
noms	Nombre del encuestador	char	30	null		
tipo	Tipo de usuario	char	17	null		
correo_e	Correo electrónico del encuestador	varchar	30	null		
num_tel	Número telefónico del encuestador	int	20	null		
login	Nombre de usuario	varchar	10	not_null		
contraseña	Contraseña para ingresar	varchar	10	not_null		
tipo_prestador	Tipo de prestador	varchar	20	not_null		

Tabla administrador – tiene relación con la tabla: encuesta_recinto

Tabla 4.31 Diccionario de Datos de la Tabla administrador

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
<u>id</u>	Identificador de usuario	int	5	not_null	✓	
id_usr	Identificador de administrador	varchar	15	not_null		✓
usr	Nombre de administrador	varchar	20	not_null		
passwd	Contraseña para ingresar al sistema	varchar	10	not_null		
tipo	Tipo de usuario	varchar	15	not_null		

Tabla consumo_computo – tiene relación con la tabla: encuesta_recinto

Tabla 4.32 Diccionario de Datos de la Tabla consumo_computo

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
<u>id_compu</u>	Identificador de equipo de computo	varchar	15	not_null	✓	
id_enc	Identificador de encuesta	varchar	11	not_null		✓
potencia	Potencia que genera	int	5	not_null		
cantidad	Cantidad de equipo	int	5	not_null		
L-V h/d	Potencia demandada de lunes a viernes	time		not_null		
S h/d	Potencia demandada los sábados	time		not_null		
D h/d	Potencia demandada los domingos	time		not_null		
observaciones	Observaciones del equipo	varchar	20	null		

Tabla consumo_papelera – tiene relación con la tabla: encuesta_recinto

Tabla 4.333 Diccionario de Datos de la Tabla consumo_papelera

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
<u>id_papel</u>	Identificador de papelería	varchar	15	not_null	✓	
id_enc	Identificador de encuesta	varchar	11	not_null		✓
potencia	Potencia que genera	int	5	not_null		
cantidad	Cantidad de equipo	int	5	not_null		
L-V h/d	Potencia demandada de lunes a viernes	time		not_null		
S h/d	Potencia demandada los sábados	time		not_null		
D h/d	Potencia demandada los domingos	time		not_null		
observaciones	Observaciones del equipo	varchar	20	null		

Tabla consumo_electro – tiene relación con la tabla: encuesta_recinto

Tabla 4.344 Diccionario de Datos de la Tabla consumo_electro

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
<u>id_electro</u>	Identificador de electrodomésticos	varchar	15	not_null	✓	
Id_despa	Identificador de despachadores	varchar	15	not_null		✓
id_enc	Identificador de encuesta	varchar	11	not_null		✓
potencia	Potencia que genera	int	5	not_null		
cantidad	Cantidad de equipo	int	5	not_null		
L-V h/d	Potencia demandada de lunes a viernes	time		not_null		

S h/d	Potencia demandada los sábados	time		not_null		
D h/d	Potencia demandada los domingos	time		not_null		
observaciones	Observaciones del equipo	varchar	20	null		

Tabla dependencia – tiene relación con la tabla: edificio

Tabla 4.35 Diccionario de Datos de la Tabla dependencia

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
<u>id_dep</u>	Identificador de dependencia	varchar	11	not_null	✓	
id_edif	Identificador de edificio	varchar	11	not_null		✓
nom_dep	Nombre de la dependencia	varchar	30	null		
nom_edif	Nombre del edificio	varchar	30	null		
ubicacion	Ubicación de la dependencia	varchar	20	null		
hor_norm	Horas normales de trabajo	int	5	null		
hor_lab	Horas laborales	int	5	null		
hor_comida	Horario de comida	time		null		
num_empleados	Número de empleados en la dependencia	int	5	null		
num_pob_flotante	Número de población flotante en la dependencia	int	5	null		

Tabla cal_bombeo – tiene relación con la tabla: encuesta_recinto

Tabla 4.36 Diccionario de Datos de la Tabla cal_bombeo

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
<u>id_bom</u>	Identificador de calefactor	varchar	10	not_null	✓	
id_encuestador	Identificador de encuestador	varchar	10	not_null		✓
marca	Marca del equipo	varchar	20	null		
pot_dem	Potencia demandada del equipo (HP)	float		null		
pot_demk	Potencia demandada del equipo (kW)	float		null		
mes_uso	Meses de uso al año	int	3	null		
hr_dia	Hora y día en que se usa	datetime		null		
kw_ohm	Kilowatts al mes	int	5	null		
comm	comentarios	varchar	20	null		

Tabla edificio – tiene relación con la tabla: encuesta_recinto, dependencia

Tabla 4.377 Diccionario de Datos de la Tabla edificio

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
id_edif	Identificador del edificio	varchar	11	not_null	✓	
nom_edif	Nombre del edificio	varchar	10	null		
nom_dep	Nombre de la dependencia	varchar	30	null		
numlocal	Número de locales	int	5	null		
numnivel	Número de niveles	int	2	null		
idlocal	Nombre del local	varchar	20	null		
orientacion	Orientación del edificio	varchar	15	null		

Tabla encuesta_ilu_artificial – tiene relación con la tabla: encuesta_recinto

Tabla 4.388 Diccionario de Datos de la Tabla encuesta_ilu_artificial

Atributos	Descripción	Tipo	Longitud	Nulos	Llave primaria	Llave Foránea
id_ilu_art	Identificador de iluminación artificial	varchar	10	not_null	✓	
id_ilu	Identificador de iluminación	varchar	5	not_null		✓
num_tot_int	Número total de interruptores	int	6	null		
luminter	Número de luminarias por interruptor	int	6	null		
hr_usod	Horas de uso diarias	int	4	null		
fluorescente	Tipo de foco fluorescente	varchar	15	null		
incandescente	Tipo de foco incandescente	varchar	10	null		
vapor	Tipo de vapor	varchar	25	null		
aditivo	Tipo de aditivo metálico	varchar	15	null		
lampara	Tipo de lámpara dicróica	varchar	15	null		
otro	Otro tipo de luz artificial	varchar	30	null		

CAPÍTULO 5 IMPLEMENTACIÓN DEL SISTEMA

El paso inicial fue instalar una máquina virtual, en este proceso fue necesario apoyarse con la máquina virtual VMWare Workstation V.7, ya que permite tener el sistema operativo Linux y el sistema operativo Windows sin ninguna partición en el equipo de cómputo, la ventaja de instalar una máquina virtual se detalló en el **tema 2.1.3**, además de que este software es libre, es decir, no tiene costo alguno, y se puede descargar desde la red. Una vez instalada la máquina virtual, se configuró un sistema LAMP, el cual fue descrito en el capítulo 3.

EL sistema se instaló de manera local en una laptop, es decir, en:

<http://127.0.0.1/SisFIUnamAhorroDeEnerqia/>

Para después implementar el sistema en un ambiente de producción, es decir, subirlo a la Web, para que este levantamiento de datos pueda ser realizado y consultado desde cualquier equipo de cómputo que cuente con acceso a internet.

El sistema consta de 3 secciones, la primer sección es la encargada de verificar que exista el usuario en el sistema y que su contraseña sea válida, esta es la sección de autenticación, la segunda sección se encuentra conformada por los módulos y entidades diseñadas para el proceso del Levantamiento de Datos (LEDA), y la tercer sección es la de generación de reporte de la encuesta.

En la siguiente figura se representan las tres secciones que se implementaron para el sistema de administración de bases de datos para el consumo de energía en edificios de Ciudad Universitaria:

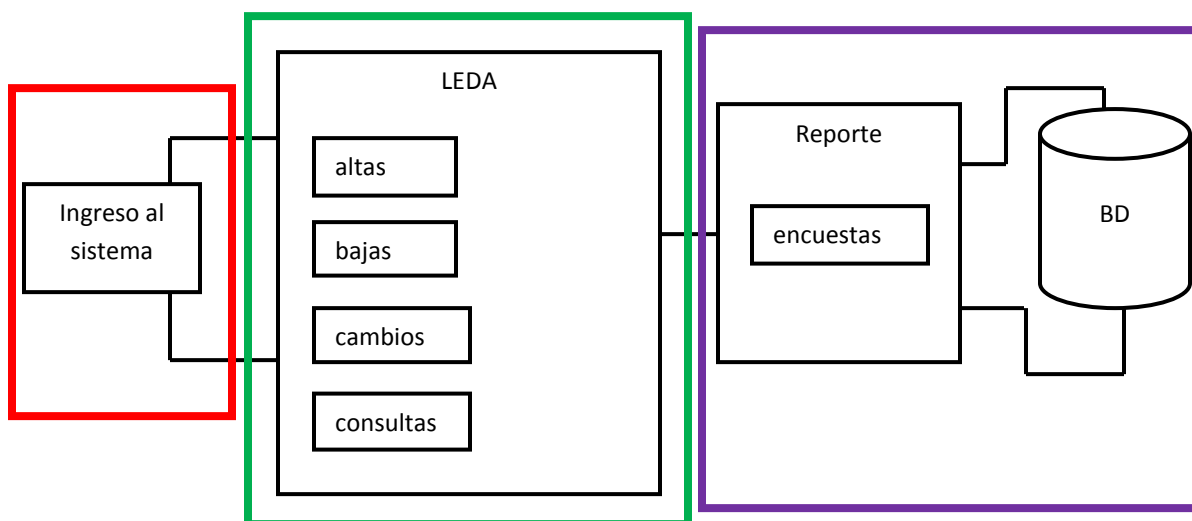


Figura 5.1 Secciones generales del SisFIUNAMAhorroDeEnerqia

● Sección de Ingreso

En este módulo asociamos los privilegios correspondientes a la cuenta de usuario, es decir el proceso de autenticación para ingresar al sistema, ya que éste cuenta con filtros de información para no afectar la seguridad y exposición de datos.

● Sección de captura de LEDA

Es la parte que contiene elementos visuales y que está vinculada a la captura de la encuesta. En esta sección se dispone de varias opciones que permiten manipular los datos, tanto de las encuestas, como de los usuarios que pueden hacer uso del sistema, y sus respectivos privilegios.

● Sección de reporte de encuesta

Esta última sección presentará el resultado del análisis de la información, mediante tablas que corresponden a los criterios establecidos por el especialista, que permita mostrar los resultados obtenidos del análisis y realizar conclusiones y recomendaciones.

5.1 Implementación de la Base de Datos

Es necesario contar con filtros de información al momento de ingresar al sistema, este proceso se encarga de validar autenticación de usuario, para después efectuar las operaciones disponibles en el sistema y así poder generar un reporte.

En este tema se describirá de forma particular cada sección del sistema SisFIUnamAhorroDeEnergía.

Sección de ingreso

El usuario accede al sistema proporcionando los datos “login” y contraseña; con estos datos el sistema realiza la validación mediante una consulta a la base de datos si corresponde a un administrador o si es encuestador, además de consultar si la contraseña es correcta.

De esta manera, el usuario se puede encontrar en la posibilidad de comenzar a realizar sus peticiones al sistema, dependiendo si es administrador o encuestador, como pueden ser Alta, Baja, Cambio, Consulta. El sistema informa al usuario el resultado de cada proceso que solicita, mediante mensajes desplegados en la pantalla del explorador.

Respecto a los filtros de información, niveles de acceso a los datos:

Se proveen las características funcionales y de procedimiento, necesarias para establecer, mantener y liberar conexiones físicas entre el dispositivo y el punto de conexión de red, es decir, se debe autorizar el acceso únicamente a los usuarios que ingresen correctamente su *login* y el

contraseña que les corresponde, para garantizar que sólo el interesado efectúa este trámite de acuerdo a sus necesidades. Esta información se debe validar con el contenido de la base de datos

Por lo tanto, ya se cuenta con la forma de conectar físicamente dos nodos, el sistema y la base de datos, y de transferir un mensaje de datos entre ellos, manejando direccionamiento, control de errores y control de autenticación.

Dichas funciones cumplen con la recomendación del modelo OSI mencionado a detalle en el **tema 2.3**.

Sección de captura de LEDA

La captura del LEDA es la parte que contiene elementos visuales y que está vinculada a la captura de la encuesta. En esta sección se dispone de varias opciones que permiten manipular los datos, tanto de las encuestas, como de los usuarios que pueden hacer uso del sistema, y sus respectivos privilegios. Los datos que se capturan van desde el nombre de dependencias, edificios, hasta el tipo de luminaria que ocupan, orientación de ventanas y tipo de equipos que consumen electricidad, etc. Todo lo que se requiere para que se pueda realizar un análisis de diagnóstico energético.

Esta sección es de las más importantes tanto para la implementación del sistema como de la base de datos. En esta sección se almacenan los módulos generales y entidades de la base de datos que fueron previamente analizadas, normalizadas y diseñadas para su correcta implementación.

A grandes rasgos los módulos generales del sistema son:

Tabla 5.1 Módulos a Implementar

Módulos Generales del sistema			
ALTAS	BAJAS	ACTUALIZACIÓN	CONSULTAS

Y las entidades que corresponden a cada módulo son los siguientes:

Tabla 5.2 Entidades a Implementar

Entidades que se requieren para el LEDA (Levantamiento de Datos)				
ENCUESTADOR	ENCUESTA	DEPENDENCIA	EDIFICIO	USUARIO

Sección de reporte de encuesta

Esta última sección presentará el resultado de la información mediante tablas que correspondan con los criterios establecidos por el administrador del sistema, para mostrar los resultados obtenidos de la encuesta y realizar conclusiones y recomendaciones.

Previamente se crearán plantillas con aquellos elementos que son constantes en todo reporte, de esta forma la sección de generación de reportes, una vez validada la información, integrará en el reporte tanto alguna de las plantillas, como las conclusiones y resultados; generando un documento en formato pdf.

5.1.1 Autenticación de usuarios a la Base de Datos

Es muy importante contar con un control de autenticación de usuarios para mantener la seguridad de nuestro sistema, de lo contrario nos arriesgaríamos a que cualquier usuario que pueda ingresar al sistema modifique o consulte datos de la base que puedan estar restringidos.

En nuestro sistema existen dos tipos de usuarios:

- **Administradores:** además de ingresar datos de encuestas, tiene privilegios para dar de baja encuestas y usuarios. Puede realizar la consulta de más datos que se ingresan a la base.
- **Encuestadores:** solo puede ingresar y actualizar datos referentes a las encuestas.

El método de autenticación utilizado en el sistema se basa en la verificación de login y contraseña, el valor de estos atributos son proporcionados por el administrador del sistema.

El sistema de identificación cuenta con la característica de ser fiable y es aceptada por los usuarios, ya que son ellos quienes utilizan el sistema.

El proceso general de autenticación consta de los siguientes pasos:

- 1) El usuario solicita acceso al sistema.
- 2) El sistema solicita al usuario que se autentique.
- 3) El usuario aporta las credenciales (login y contraseña) que lo identifican y permiten verificar la autenticidad de la identificación.
- 4) El sistema valida según la información que existe en la base de datos si las credenciales aportadas son suficientes para dar acceso al usuario o no.
- 5) Si dicha información almacenada en la base de datos, indica que el usuario está iniciando sesión como *tipo=Administrador*, entonces se realiza la conexión a la base de datos por medio de código *php*, y permite el acceso al sistema mostrando las operaciones que puede realizar este usuario. Este usuario tiene el privilegio de dar de alta los datos de las entidades Encuestador, Encuesta, Dependencia, Edificio y Usuario, así como también la baja, actualización y consulta de datos de las mismas.

- 6) En caso contrario, si dicha información almacenada en la base de datos, indica que el usuario está iniciando sesión como *tipo=Encuestador*, entonces de igual manera, se realiza la conexión a la base de datos por medio de código *php*, y permite el acceso al sistema mostrando las operaciones que puede realizar este usuario. Este usuario solo tiene el privilegio de dar de alta los datos de las entidades Encuesta, Dependencia y Edificio, así como también la actualización y consulta de datos de las mismas y no puede realizar la baja de ningún tipo de información, asegurando así la confidencialidad, la integridad y la disponibilidad de los datos de las encuestas almacenadas en el sistema.

Permisos a objetos de la base de datos

La concesión de acceso de usuario a una base de datos implica tres pasos. Primero, se debe crear un inicio de sesión. El inicio de sesión permite al usuario conectarse a la de base de datos del sistema. Luego, se debe configurar el inicio de sesión como un usuario de la base de datos. Y, por último, se debe conceder al usuario permiso a objetos de la base de datos.

A continuación se describen los pasos para otorgar el permiso de acceso a objetos:

- Inicio de sesión
- Conceder acceso a la base de datos
- Creación de vistas
- Conceder acceso a un objeto de la base de datos

✓ Inicio de sesión

El inicio de sesión representa la identidad del usuario como una cuenta de acceso al sistema, o a la base de datos.

De forma predeterminada, los administradores del sistema tienen acceso total a la base de datos. Para asegurar la integridad y confidencialidad de datos dentro del sistema, es necesario crear un usuario con menos privilegios, en este caso es el encuestador; por tanto, se deberá crear una nueva cuenta de autenticación del sistema local en el equipo. Para hacerlo, debe ser un administrador del equipo.

Para esto el administrador del equipo con la sentencia CREATE LOGIN de MySQL, crea al usuario con menos privilegios.

✓ Acceso a la base de datos

Ahora el usuario creado con menos privilegios tiene acceso al sistema, sin embargo no cuenta con el permiso para tener acceso a la base de datos.

Para conceder el permiso al usuario del acceso a la base de datos el administrador usa la instrucción CREATE USER para asignar su inicio de sesión a un usuario denominado *encuestador*.

✓ Creación de vistas

Ahora ya el usuario denominado *encuestador*, puede tener acceso a la base de datos, puede que desee crear algunos objetos de base de datos, como una vista o un procedimiento almacenado y concederle a *encuestador* acceso a los mismos. Una vista es una instrucción SELECT almacenada y un procedimiento almacenado es una o varias instrucciones SQL que se ejecutan como un lote.

Las vistas se consultan como las tablas y no aceptan parámetros. Los procedimientos almacenados son más complejos que las vistas. Los procedimientos almacenados pueden tener parámetros de entrada y salida y pueden contener instrucciones para controlar el flujo del código, como instrucciones IF y WHILE. Una práctica recomendable de programación es usar procedimientos almacenados para realizar todas las tareas repetitivas en la base de datos.

✓ Creación de vistas

Como administrador, puede ejecutar la instrucción SELECT desde cualquier tabla, cualquier vista y ejecutar el procedimiento; sin embargo, el usuario *encuestador* no puede hacerlo. Para conceder a *encuestador* los permisos necesarios, se usa la instrucción GRANT.

Acerca de GRANT

Para ejecutar un procedimiento almacenado, debe tener permiso EXECUTE. Para tener acceso a datos y cambiarlos, debe tener permisos SELECT, INSERT, UPDATE y DELETE. La instrucción GRANT también se usa para otros permisos, como el permiso para crear tablas.

5.2 Implementación del Sitio Web

Una de las herramientas más utilizadas para la publicación y transferencia de la información generada y recopilada en los sistemas de información, tanto por particulares, como por entes públicos y empresas privadas, es la Internet. Es por eso que para nuestro sistema implementaremos el sitio web, y así poder hacer más eficaz el análisis del diagnóstico energético.

Una vez que ya tenemos implementado el sistema y el diseño de la estructura lógica de la base de datos para nuestro sistema, se comienza con la programación web, programación del back-end y el front-end. Como ya se mencionó en el capítulo III, las herramientas HTML y PHP para la programación, brindan apoyo para la realización del sistema.

A continuación se detalla el proceso general del SisFIUnamAhorroDeEnergía

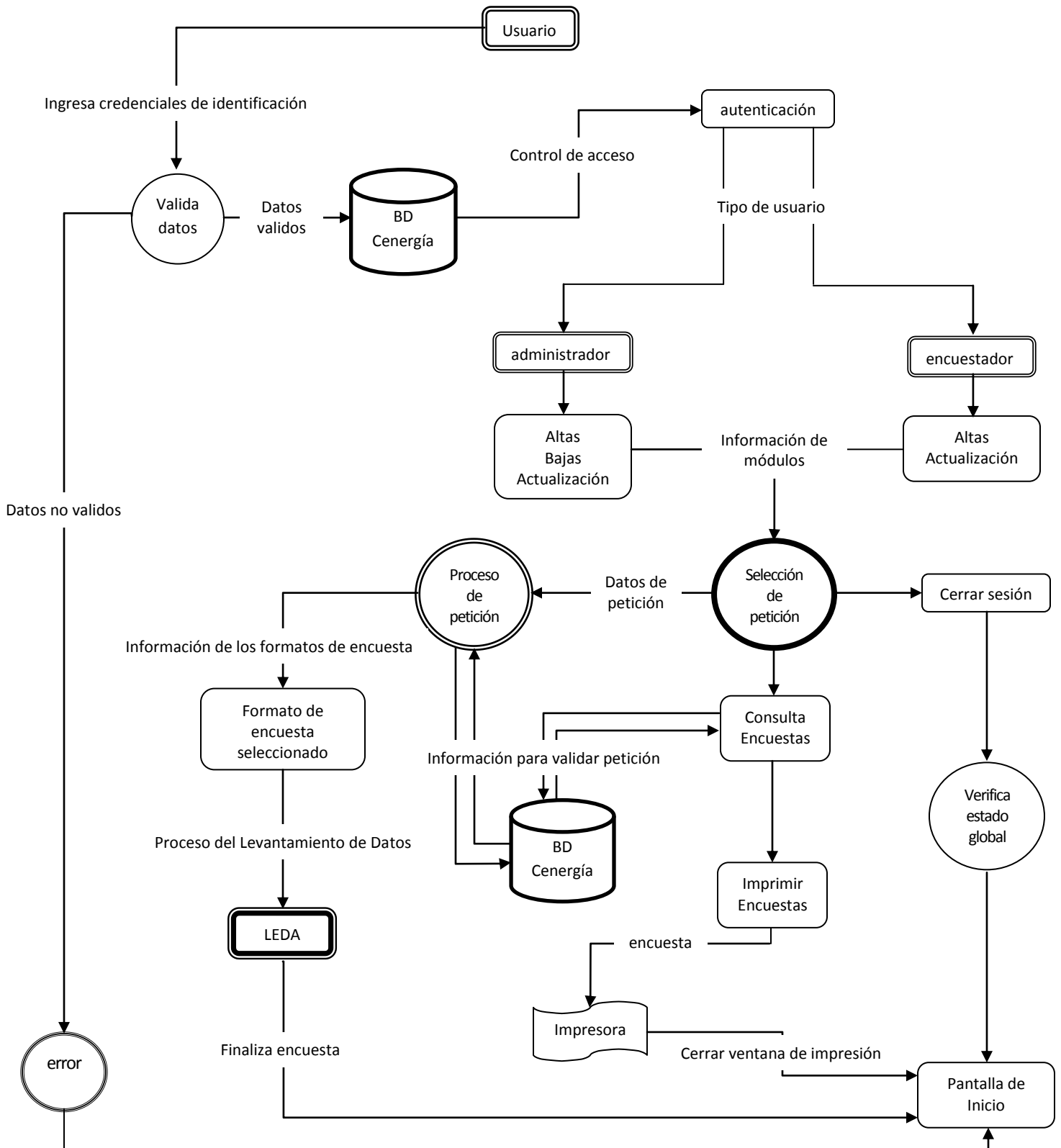


Figura 5.2 Proceso general del SisFIUNAMAHorroDeEnergía

Una vez detallado el proceso del sistema y en base al diseño de cada módulo se realiza la implementación de cada uno de ellos.

A continuación se detalla la implementación de cada módulo del sistema al ingresar como **Administrador**:

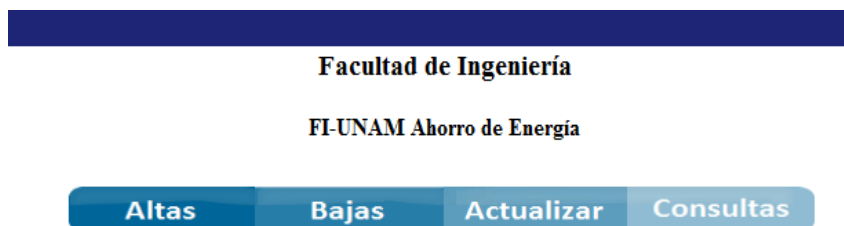


Figura 5.3 Implementación de módulos (Administrador)

❖ Altas

Se programa el módulo *altas* en php y html



Figura 5.4 Implementación de Entidades para el módulo Altas (Administrador)

Tabla 5.3 Implementación del módulo Altas (Administrador)

Módulo	Entidad
ALTAS	ENCUESTADOR
Descripción: Esta entidad permite ingresar toda la información necesaria del usuario que realizará la encuesta. Deberá ingresar todos los datos personales que se requieran. Es importante que el usuario cuente con un identificador, como su número de cuenta, número de empleado, etc.	

ALTA DE DATOS PARA EL ENCUESTADOR	
IDENTIFICADOR DEL ENCUESTADOR:	<input type="text"/>
APELLIDO PATERNO:	<input type="text"/>
APELLIDO MATERNO:	<input type="text"/>
NOMBRE(S):	<input type="text"/>
TELÉFONO:	<input type="text"/>
EMAIL:	<input type="text"/>
LOGIN:	<input type="text"/>
CONTRASEÑA:	<input type="text"/>
TIPO:	SELECCIONA UNA OPCION ▼
TIPO DE PRESTADOR:	SELECCIONA UNA OPCION ▼
<input type="button" value="ALTA"/> <input type="button" value="LIMPIAR"/>	

Módulo	Entidad																								
ALTAS	DEPENDENCIA																								
<p>Descripción: Esta entidad contiene la información de la dependencia a la que se le realizará el diagnóstico energético. Se necesitará de un identificador de dependencia para contar con una buena organización de la base de datos, así como conocer nombre de edificios, ubicación en la que se encuentra la dependencia, horas de trabajo y número de empleados.</p>																									
<table border="1"> <thead> <tr> <th colspan="2">ALTA DE DATOS PARA LA DEPENDENCIA</th> </tr> </thead> <tbody> <tr> <td>IDENTIFICADOR DE LA DEPENDENCIA:</td> <td><input type="text"/></td> </tr> <tr> <td>IDENTIFICADOR DEL EDIFICIO:</td> <td><input type="text"/></td> </tr> <tr> <td>DEPENDENCIA:</td> <td><input type="text"/></td> </tr> <tr> <td>EDIFICIO:</td> <td><input type="text"/></td> </tr> <tr> <td>UBICACIÓN:</td> <td><input type="text"/></td> </tr> <tr> <td>HORAS NORMALES:</td> <td><input type="text"/></td> </tr> <tr> <td>HORAS LABORABLES:</td> <td><input type="text"/></td> </tr> <tr> <td>HORARIO DE COMIDA:</td> <td><input type="text"/></td> </tr> <tr> <td>NÚMERO DE EMPLEADOS:</td> <td><input type="text"/></td> </tr> <tr> <td>POBLACIÓN FLOTANTE:</td> <td><input type="text"/></td> </tr> <tr> <td colspan="2"> <input type="button" value="ALTA"/> <input type="button" value="LIMPIAR"/> </td> </tr> </tbody> </table>		ALTA DE DATOS PARA LA DEPENDENCIA		IDENTIFICADOR DE LA DEPENDENCIA:	<input type="text"/>	IDENTIFICADOR DEL EDIFICIO:	<input type="text"/>	DEPENDENCIA:	<input type="text"/>	EDIFICIO:	<input type="text"/>	UBICACIÓN:	<input type="text"/>	HORAS NORMALES:	<input type="text"/>	HORAS LABORABLES:	<input type="text"/>	HORARIO DE COMIDA:	<input type="text"/>	NÚMERO DE EMPLEADOS:	<input type="text"/>	POBLACIÓN FLOTANTE:	<input type="text"/>	<input type="button" value="ALTA"/> <input type="button" value="LIMPIAR"/>	
ALTA DE DATOS PARA LA DEPENDENCIA																									
IDENTIFICADOR DE LA DEPENDENCIA:	<input type="text"/>																								
IDENTIFICADOR DEL EDIFICIO:	<input type="text"/>																								
DEPENDENCIA:	<input type="text"/>																								
EDIFICIO:	<input type="text"/>																								
UBICACIÓN:	<input type="text"/>																								
HORAS NORMALES:	<input type="text"/>																								
HORAS LABORABLES:	<input type="text"/>																								
HORARIO DE COMIDA:	<input type="text"/>																								
NÚMERO DE EMPLEADOS:	<input type="text"/>																								
POBLACIÓN FLOTANTE:	<input type="text"/>																								
<input type="button" value="ALTA"/> <input type="button" value="LIMPIAR"/>																									

Módulo	Entidad
ALTAS	EDIFICIO
<p>Descripción: Para dar de alta a esta entidad es necesario conocer el identificador del edificio, así como su nombre, en que dependencia se encuentra el edificio, número de locales, niveles y la orientación en la que se encuentra el edificio.</p>	

ALTA DE DATOS PARA EL EDIFICIO

IDENTIFICADOR DEL EDIFICIO:
EDIFICIO:
DEPENDENCIA:
NÚMERO DE LOCALES:
NÚMERO DE NIVELES:
NOMBRE DEL LOCAL:
ORIENTACIÓN:

Módulo	Entidad
ALTAS	USUARIO
<p>Descripción: Esta entidad es la que se maneja como Administrador, para dar de alta a este usuario es necesario conocer su identificador de usuario, nombre y proporcionar un <i>login</i> y contraseña para que éste pueda ingresar al sistema.</p>	
<div style="background-color: #4a90e2; color: white; padding: 5px; margin-bottom: 10px;"> ALTA DE DATOS PARA USUARIO </div> <p> IDENTIFICADOR DEL USUARIO: <input type="text"/> NOMBRE: <input type="text"/> CONTRASEÑA: <input type="text"/> TIPO: <input type="text" value="SELECCIONA UNA OPCION"/> </p> <p style="text-align: center;"> <input type="button" value="ALTA"/> <input type="button" value="LIMPIAR"/> </p>	

Módulo	Entidad
ALTAS	ENCUESTA
<p>Descripción: En esta entidad, se requiere generar un identificador de encuesta. Esta entidad contiene toda la información para poder realizar el análisis del recinto, análisis de iluminación del recinto, fuentes de luz artificial y el tipo de encuesta.</p>	
<p>Comenzamos con el análisis de área del recinto:</p>	

ANÁLISIS DE ÁREA DEL RECINTO

Identificador de Encuesta:

Id del Encuestador: Id de la Dependencia:
 Dependencia: Edificio:
 Orientación: NORTE Nivel:
 Identificador de local:
 Número del local: Fecha: 01 enero
 Hora: (en formato 24 hrs) Año:
 Estado del Clima: SOLEADO

NOTA: Todas las dimensiones se darán en metros

Número de ocupantes Uso del Recinto Aulas
 Tonalidad de muro Claro Edad promedio de ocupantes Menores de 40
 Tonalidad de techo Claro Tipo de Atmósfera Sin polvo
 Tonalidad de piso Claro Altura de la fuente luminosa

Dimensiones del local en metros
 Largo Ancho Alto

Iluminación natural del recinto

ILUMINACIÓN NATURAL DEL RECINTO

Id Iluminación Id Encuesta

Dimensiones del domo o tragaluz
 Largo Ancho

Tipo de acceso de Iluminación Unilateral
 Tipo de cristales en ventana: No hay ventanas
 Altura al Plano de Trabajo:

Dimensión y orientación de ventanas
 Largo Alto Orientación NORTE
 Largo Alto Orientación NORTE
 Largo Alto Orientación NORTE

Nivel de piso a ventana:
 Estado de limpieza Bueno
 Altura o distancia entre el tragaluz y el plano de trabajo
 Comentarios:

Iluminación artificial del recinto

ILUMINACIÓN ARTIFICIAL DEL RECINTO

Id Luz Artificial Id Iluminación

Número total de interruptores

Luminarias por interruptor

Horas de uso diarias

Especificación de fuentes de luz artificial

Fluorescentes: No Aplica

Focos Incandescentes: No Aplica

Vapor: No Aplica

Aditivos Metálicos: No Aplica

Lámpara Dicroica: No Aplica

Otro:

Tipo de encuesta a realizar

SELECCIONE LA ENCUESTA QUE DESEA LLENAR

Aire Acondicionado

Taller-Laboratorio

Calentadores y Bombas

Misceláneos

Equipo de Aire acondicionado

ACONDICIONAMIENTO AMBIENTAL

Identificador de Equipo Identificador de Encuestador

Equipo	Marca	Modelo	Capacidad TR	Antigüedad del equipo [años]	REE	Potencia Max. [kW]	Horas Día	Meses al año	Días Mes	Descripción	Cantidad
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Equipo de Taller o Laboratorio

TALLER O LABORATORIO														
							Identificador de Equipo							
							Identificador de Encuestador							
Equipo	Cantidad	Potencia [w]	Hora /Dias	kW instalados	Dia /Mes	Mes /Año	Dia /Año de operación	Hora /Año de operación	Hora /Dias distribuidas al año	FC	kW demanda	kWh /Mes	Comentarios	

Equipo de bombeo o calefactores

BOMBEO O CALEFACTORES							
			Identificador de Equipo				
			Identificador de Encuestador				
Marca	Potencia demandada (HP)	Potencia demandada (kW)	Mes de uso /Año	Hora /Dia	kWh /mes	Comentarios	

Misceláneos



Seleccionando equipo de cómputo



Equipo de Cómputo

- SELECCIONA UNA OPCION
- COMPUTADORA
- IMPRESORA
- SCANNER
- PLOTTER
- REGULADOR
- NO-BREAK
- UNIDAD CD-RW

Seleccionando equipo de papelería



Equipo de Papelería

- SELECCIONA UNA OPCION
- DESTRUCTOR DE PAPEL
- FAX
- FOTOCOPIADORA
- GUILLOTINA
- MÁQUINA DE ESCRIBIR
- PROYECTOR
- REPRODUCTOR DVD
- SACAPUNTAS
- SUMADORA
- TRITURADOR

Seleccionando electrodomésticos

ALTA DE DATOS PARA ELECTRODOMÉSTICOS

Electrodomésticos: SELECCIONA UNA OPCION ▾
SELECCIONA UNA OPCION
CAFETERA
CALEFACTOR
ENFRIADOR/CALENTADOR
EQUIPO DE VIDEO
ESTÉREO
GRABADORA
HORNO DE MICROONDAS
PARRILLA ELÉCTRICA
RADIO
RADIO-GRABADORA
REFRIGERADOR
SECADORA DE MANOS
TELÉFONO INALÁMBRICO
TELEVISOR
TOSTADOR
VAPORIZADOR
VENTILADOR
VIDEO VHS

ALTA DE DATOS PARA ELECTRODOMÉSTICOS

Electrodomésticos: SELECCIONA UNA OPCION ▾

Despachador: SELECCIONA UNA OPCION ▾
SELECCIONA UNA OPCION
CAFÉ
GOLOSINAS
REFRESCOS

❖ Bajas

Se programa el módulo *bajas* en php y html



Figura 5.5 Implementación de Entidades para el módulo Bajas (Administrador)

Tabla 5.4 Implementación del módulo Bajas (Administrador)

Módulo	Entidad																								
BAJAS	ENCUESTADOR																								
Descripción: Muestra una lista de todos los encuestadores que se encuentran registrados en la base de datos, entonces se selecciona que encuestador es aquel que se desea dar de baja del sistema.																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="6" style="background-color: #ADD8E6;">BAJA DE DATOS PARA EL ENCUESTADOR</th> </tr> <tr> <th style="background-color: #ADD8E6;">Cuenta del Encuestador</th> <th style="background-color: #ADD8E6;">Apellido Paterno</th> <th style="background-color: #ADD8E6;">Apellido Materno</th> <th style="background-color: #ADD8E6;">Nombre(s)</th> <th style="background-color: #ADD8E6;">Tipo de Usuario</th> <th style="background-color: #ADD8E6;">Baja</th> </tr> </thead> <tbody> <tr> <td>405023822</td> <td>Pozas</td> <td>Montoya</td> <td>Karen</td> <td>USUARIO</td> <td style="text-align: center;">BORRAR</td> </tr> <tr> <td>302166239</td> <td>Rendon</td> <td>Salgado</td> <td>Axel</td> <td>USUARIO</td> <td style="text-align: center;">BORRAR</td> </tr> </tbody> </table>		BAJA DE DATOS PARA EL ENCUESTADOR						Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	Baja	405023822	Pozas	Montoya	Karen	USUARIO	BORRAR	302166239	Rendon	Salgado	Axel	USUARIO	BORRAR
BAJA DE DATOS PARA EL ENCUESTADOR																									
Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	Baja																				
405023822	Pozas	Montoya	Karen	USUARIO	BORRAR																				
302166239	Rendon	Salgado	Axel	USUARIO	BORRAR																				

Módulo	Entidad
BAJAS	DEPENDENCIA
Descripción: Muestra una lista de todas las dependencias que se encuentran registradas en la base de datos, entonces se selecciona aquella dependencia que se desea dar de baja del sistema.	

BAJA DE DATOS PARA LA DEPENDENCIA										
Identificador de Dependencia	Identificador de Edificio	Dependencia	Edificio	Ubicación	Horas Normales	Horas Laborales	Horario de Comida	Número de Empleados	Población Flotante	Baja
CELE-B-200	B-200	CELE	B	Ciudad Universitaria	12	8	13:00:00	1234	12345	BORRAR
FI-A-001	A-001	Facultad de Ingeniería	A	Ciudad Universitaria	12	8	13:00:00	1234	12345	BORRAR
FI-K-USECAD	K-USECAD	Facultad de Ingeniería	K	Ciudad Universitaria	12	8	13:00:00	1234	12345	BORRAR

Módulo	Entidad						
BAJAS	EDIFICIO						
Descripción: Muestra una lista de todos los edificios que se encuentran registrados en la base de datos, entonces se selecciona el edificio que se desea dar de baja del sistema.							
BAJA DE DATOS PARA EDIFICIO							
Identificador del Edificio	Edificio	Dependencia	Número de Locales	Número de Niveles	Nombre del Local	Orientación	Baja
A-001	A	Facultad de Ingeniería	50	4	001	Sur	BORRAR
B-200	B	CELE	50	2	200	Sur	BORRAR
K-USECAD	K	Facultad de Ingeniería	5	1	usecad	norte	BORRAR

Módulo	Entidad			
BAJAS	USUARIO			
Descripción: Muestra una lista de todos los edificios que se encuentran registrados en la base de datos, entonces se selecciona el edificio que se desea dar de baja del sistema.				
BAJA DE DATOS PARA EL USUARIO				
Identificador de Usuario	Nombre de Usuario	Contraseña	Tipo de Usuario	Baja
302166239	Jossavet	123pat0	ADMINISTRADOR	BORRAR
302166239KJ	karla	123hola	USUARIO	BORRAR

Módulo	Entidad
BAJAS	ENCUESTA
Descripción: Muestra una lista de todas las encuestas que se encuentran registrados en la base de datos, entonces se selecciona la encuesta que se desea borrar del sistema.	

BAJA DE DATOS DE LA ENCUESTA									
Encuesta	Nombre de la Dependencia	Nombre del Edificio	Día	Mes	Hora en la que se elaboro la Encuesta	Año	Estado del Clima	Uso del recinto	Encuesta Completa
001_FI	Facultad de Ingenieria	A	05	11	19:16:00	2012	NUBLADO	Aulas	BORRAR
002_CELE	CELE	B	05	11	19:16:00	2012	NUBLADO	Aulas	BORRAR

❖ Actualizar

Se programa el módulo *actualizar* en php y html

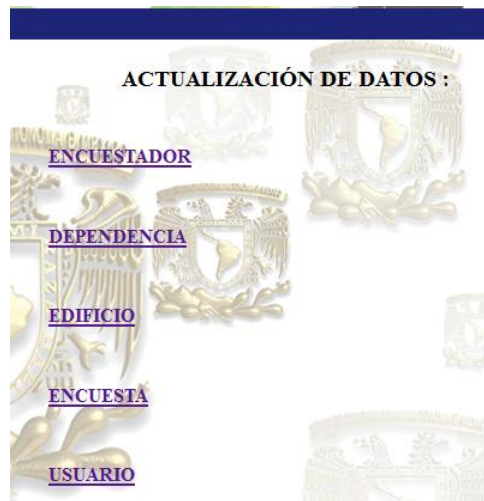


Figura 5.6 Implementación de Entidades para el módulo Actualizar (Administrador)

Tabla 5.5 Implementación del módulo Actualizar (Administrador)

Módulo	Entidad
ACTUALIZAR	ENCUESTADOR
Descripción: Muestra una lista de todos los encuestadores registrados en el sistema, una vez seleccionado el encuestador, se proporciona la información que se desea modificar.	

ACTUALIZACION DE DATOS PARA EL ENCUESTADOR

Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	E-Mail	Numero de Teléfono	Login	Contraseña	Tipo de Prestador	Actualizar
405023822	Pozas	Montoya	Karen	USUARIO	san_kisme@hotmail.com	5523254209	karen	karen	HONORARIO	ACTUALIZA
302166239	Rendon	Salgado	Axel	USUARIO	jossavet@gmail.com	5527065443	axel	AXELJOSS	VOLUNTARIO	ACTUALIZA

ACTUALIZAR DATOS DEL ENCUESTADOR

IDENTIFICADOR DEL ENCUESTADOR:

APELLIDO PATERNO:

APELLIDO MATERNO:

NOMBRE(S):

TELÉFONO:

EMAIL:

LOGIN:

CONTRASEÑA:

TIPO:

TIPO DE PRESTADOR:

Módulo

Entidad

ACTUALIZAR

DEPENDENCIA

Descripción: Muestra una lista de todas las dependencias registradas en el sistema, una vez seleccionada la dependencia, se proporciona la información que se desea modificar.

ACTUALIZACION DE DATOS PARA LA DEPENDENCIA

Identificador de Dependencia	Identificador de Edificio	Dependencia	Edificio	Ubicación	Horas Normales	Horas Laborales	Horario de Comida	Número de Empleados	Población Flotante	Actualizar
CELE-B-200	B-200	CELE	B	Ciudad Universitaria	12	8	13:00:00	1234	12345	ACTUALIZA
FI-A-001	A-001	Facultad de Ingeniería	A	Ciudad Universitaria	12	8	13:00:00	1234	12345	ACTUALIZA
FI-K-USECAD	K-USECAD	Facultad de Ingeniería	K	Ciudad Universitaria	12	8	13:00:00	1234	12345	ACTUALIZA

ACTUALIZACION DE DATOS PARA LA SIGUIENTE DEPENDENCIA

IDENTIFICADOR DE LA DEPENDENCIA:

IDENTIFICADOR DEL EDIFICIO:

DEPENDENCIA:

EDIFICIO:

UBICACIÓN:

HORAS NORMALES:

HORAS LABORABLES:

HORARIO DE COMIDA:

NÚMERO DE EMPLEADOS:

POBLACIÓN FLOTANTE:

Módulo	Entidad																																
ACTUALIZAR	EDIFICIO																																
Descripción: Muestra una lista de todos los edificios registrados en el sistema, una vez seleccionado el edificio, se proporciona la información que se desea modificar.																																	
<div style="background-color: #ADD8E6; padding: 5px; border: 1px solid black;">ACTUALIZACION DE DATOS DEL EDIFICIO</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Identificador del Edificio</th> <th>Edificio</th> <th>Dependencia</th> <th>Número de Locales</th> <th>Número de Niveles</th> <th>Nombre del Local</th> <th>Orientación</th> <th>Actualizar</th> </tr> </thead> <tbody> <tr> <td>A-001</td> <td>A</td> <td>Facultad de Ingeniería</td> <td>50</td> <td>4</td> <td>001</td> <td>Sur</td> <td>ACTUALIZA</td> </tr> <tr> <td>B-200</td> <td>B</td> <td>CELE</td> <td>50</td> <td>2</td> <td>200</td> <td>Sur</td> <td>ACTUALIZA</td> </tr> <tr> <td>K-USECAD</td> <td>K</td> <td>Facultad de Ingeniería</td> <td>5</td> <td>1</td> <td>usecad</td> <td>norte</td> <td>ACTUALIZA</td> </tr> </tbody> </table> <div style="background-color: #ADD8E6; padding: 5px; border: 1px solid black; margin-top: 10px;">ACTUALIZACION DE DATOS PARA EL SIGUIENTE EDIFICIO</div> <p style="text-align: center;"> IDENTIFICADOR DEL EDIFICIO: <input type="text" value="A-001"/> EDIFICIO: <input type="text" value="A"/> DEPENDENCIA: <input type="text" value="Facultad de Ingeniería"/> NÚMERO DE LOCALES: <input type="text" value="50"/> NÚMERO DE NIVELES: <input type="text" value="4"/> NOMBRE DEL LOCAL: <input type="text" value="001"/> ORIENTACIÓN: <input type="text" value="Sur"/> </p> <p style="text-align: center;"> <input type="button" value="ACTUALIZAR"/> <input type="button" value="LIMPIAR"/> </p>		Identificador del Edificio	Edificio	Dependencia	Número de Locales	Número de Niveles	Nombre del Local	Orientación	Actualizar	A-001	A	Facultad de Ingeniería	50	4	001	Sur	ACTUALIZA	B-200	B	CELE	50	2	200	Sur	ACTUALIZA	K-USECAD	K	Facultad de Ingeniería	5	1	usecad	norte	ACTUALIZA
Identificador del Edificio	Edificio	Dependencia	Número de Locales	Número de Niveles	Nombre del Local	Orientación	Actualizar																										
A-001	A	Facultad de Ingeniería	50	4	001	Sur	ACTUALIZA																										
B-200	B	CELE	50	2	200	Sur	ACTUALIZA																										
K-USECAD	K	Facultad de Ingeniería	5	1	usecad	norte	ACTUALIZA																										

Módulo	Entidad															
ACTUALIZAR	USUARIO															
Descripción: Muestra una lista de todos los usuarios registrados en el sistema, una vez seleccionado el usuario, se proporciona la información que se desea modificar.																
<div style="background-color: #ADD8E6; padding: 5px; border: 1px solid black;">ACTUALIZACION DE DATOS DE ADMINISTRADOR</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Identificador de Usuario</th> <th>Nombre de Usuario</th> <th>Contraseña</th> <th>Tipo de Usuario</th> <th>Actualizar</th> </tr> </thead> <tbody> <tr> <td>302166239</td> <td>Jossavet</td> <td>123pat0</td> <td>ADMINISTRADOR</td> <td>ACTUALIZA</td> </tr> <tr> <td>302166239KJ</td> <td>karla</td> <td>123hola</td> <td>USUARIO</td> <td>ACTUALIZA</td> </tr> </tbody> </table> <div style="background-color: #ADD8E6; padding: 5px; border: 1px solid black; margin-top: 10px;">ACTUALIZAR DATOS DEL USUARIO</div> <p style="text-align: center;"> IDENTIFICADOR DEL USUARIO: <input type="text" value="302166239"/> NOMBRE: <input type="text" value="Jossavet"/> CONTRASEÑA: <input type="text" value="123pat0"/> TIPO: ADMINISTRADOR <input type="button" value="v"/> </p> <p style="text-align: center;"> <input type="button" value="ACTUALIZAR"/> <input type="button" value="LIMPIAR"/> </p>		Identificador de Usuario	Nombre de Usuario	Contraseña	Tipo de Usuario	Actualizar	302166239	Jossavet	123pat0	ADMINISTRADOR	ACTUALIZA	302166239KJ	karla	123hola	USUARIO	ACTUALIZA
Identificador de Usuario	Nombre de Usuario	Contraseña	Tipo de Usuario	Actualizar												
302166239	Jossavet	123pat0	ADMINISTRADOR	ACTUALIZA												
302166239KJ	karla	123hola	USUARIO	ACTUALIZA												

Módulo	Entidad																														
ACTUALIZAR	ENCUESTA																														
Descripción: Muestra una lista de todas las encuestas registradas en el sistema, una vez seleccionado la encuesta, se proporciona la información que se desea modificar.																															
<div style="background-color: #ADD8E6; padding: 5px; border: 1px solid black; margin: 10px auto; width: 80%;">ACTUALIZACION DE DATOS DE LA ENCUESTA</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Encuesta</th> <th>Nombre de la Dependencia</th> <th>Nombre del Edificio</th> <th>Día</th> <th>Mes</th> <th>Hora en la que se elaboro la Encuesta</th> <th>Año</th> <th>Estado del Clima</th> <th>Uso del recinto</th> <th>Encuesta Completa</th> </tr> </thead> <tbody> <tr> <td>001_FI</td> <td>Facultad de Ingenieria</td> <td>A</td> <td>05</td> <td>11</td> <td>19:16:00</td> <td>2012</td> <td>NUBLADO</td> <td>Aulas</td> <td>ACTUALIZA</td> </tr> <tr> <td>002_CELE</td> <td>CELE</td> <td>B</td> <td>05</td> <td>11</td> <td>19:16:00</td> <td>2012</td> <td>NUBLADO</td> <td>Aulas</td> <td>ACTUALIZA</td> </tr> </tbody> </table>		Encuesta	Nombre de la Dependencia	Nombre del Edificio	Día	Mes	Hora en la que se elaboro la Encuesta	Año	Estado del Clima	Uso del recinto	Encuesta Completa	001_FI	Facultad de Ingenieria	A	05	11	19:16:00	2012	NUBLADO	Aulas	ACTUALIZA	002_CELE	CELE	B	05	11	19:16:00	2012	NUBLADO	Aulas	ACTUALIZA
Encuesta	Nombre de la Dependencia	Nombre del Edificio	Día	Mes	Hora en la que se elaboro la Encuesta	Año	Estado del Clima	Uso del recinto	Encuesta Completa																						
001_FI	Facultad de Ingenieria	A	05	11	19:16:00	2012	NUBLADO	Aulas	ACTUALIZA																						
002_CELE	CELE	B	05	11	19:16:00	2012	NUBLADO	Aulas	ACTUALIZA																						

❖ Consultas

Se programa el módulo *consultas* en php y html

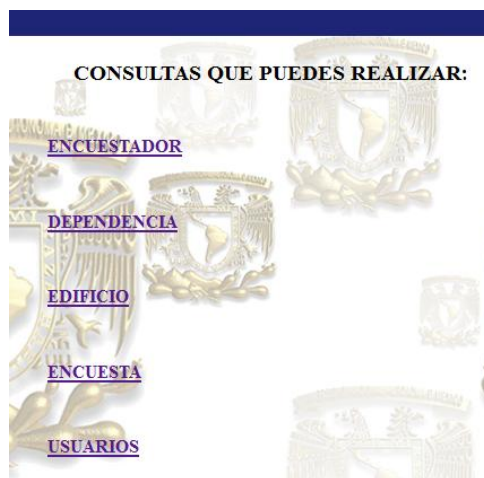


Figura 5.7 Implementación de Entidades para el módulo de Consultas (Administrador)

Tabla 5.6 Implementación del módulo Consultas (Administrador)

Módulo	Entidad
CONSULTAS	ENCUESTADOR
Descripción: El módulo de consultas se encuentra delimitado, para esta entidad, ya sea por <i>identificador del encuestador</i> o por <i>tipo de usuario</i> , sin embargo también se puede realizar la consulta de todos los encuestadores del sistema.	

CONSULTAR POR:

[IDENTIFICADOR DEL ENCUESTADOR](#)

[TIPO DE USUARIO](#)

[TODOS](#)

CONSULTA DE DATOS PARA EL ENCUESTADOR

IDENTIFICADOR DEL ENCUESTADOR:

CONSULTA DE DATOS PARA EL ENCUESTADOR

TIPO DE USUARIO:

DATOS DEL ENCUESTADOR

Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	e-mail	Teléfono	Login	Contraseña	Tipo de Prestador
405023822	Pozas	Montoya	Karen	USUARIO	san_kisme@hotmail.com	5523254209	karen	karen	HONORARIO
302166239	Rendon	Salgado	Axel	USUARIO	jossavet@gmail.com	5527065443	axel	AXELJOSS	VOLUNTARIO

Módulo	Entidad
CONSULTAS	DEPENDENCIA

Descripción: El módulo de consultas se encuentra delimitado, para esta entidad, ya sea por *dependencia* o por *ubicación*, sin embargo también se puede realizar la consulta de todas las dependencias del sistema.

CONSULTAR POR:

[DEPENDENCIA](#)

[UBICACIÓN](#)

[TODOS](#)

CONSULTA DE DATOS PARA LA DEPENDENCIA

DEPENDENCIA:

CONSULTA DE DATOS PARA LA DEPENDENCIA

UBICACIÓN:

ENVIAR

LIMPIAR

DATOS DE LA DEPENDENCIA

Identificador de Dependencia	Identificador de Edificio	Dependencia	Edificio	Ubicación	Horas Normales	Horas Laborales	Horario de Comida	Número de Empleados	Población Flotante
CELE-B-200	B-200	CELE	B	Ciudad Universitaria	12	8	13:00:00	1234	12345
FI-A-001	A-001	Facultad de Ingeniería	A	Ciudad Universitaria	12	8	13:00:00	1234	12345
FI-K-USECAD	K-USECAD	Facultad de Ingeniería	K	Ciudad Universitaria	12	8	13:00:00	1234	12345

Módulo

CONSULTAS

Entidad

EDIFICIO

Descripción: El módulo de consultas se encuentra delimitado, para esta entidad, ya sea por *identificador del edificio* o por *orientación*, sin embargo también se puede realizar la consulta de todos los edificios que se dieron de alta en el sistema.

CONSULTAR POR:

EDIFICIO

ORIENTACIÓN

TODOS

CONSULTA DE DATOS PARA EL EDIFICIO

IDENTIFICADOR DEL EDIFICIO:

ENVIAR

LIMPIAR

CONSULTA DE DATOS PARA EL EDIFICIO

ORIENTACIÓN DEL EDIFICIO:

ENVIAR

LIMPIAR

DATOS DEL EDIFICIO

Identificador del Edificio	Edificio	Dependencia	Número de Locales	Número de Niveles	Nombre del Local	Orientación
A-001	A	Facultad de Ingeniería	50	4	001	Sur
B-200	B	CELE	50	2	200	Sur
K-USECAD	K	Facultad de Ingeniería	5	1	usecad	norte

Módulo	Entidad			
CONSULTAS	USUARIO			
Descripción: En este módulo y con esta entidad se realiza la consulta de todos los usuarios del sistema.				
USUARIOS DEL SISTEMA				
Número	Identificador de Usuario	Nombre de Usuario	Contraseña	Tipo de Usuario
30	302166239	Jossavet	123pat0	ADMINISTRADOR
34	302166239KJ	karla	123hola	USUARIO

Módulo	Entidad								
CONSULTAS	ENCUESTA								
Descripción: La consulta se realiza por <i>identificador de la encuesta</i> , una vez seleccionada la encuesta a consultar es necesario seleccionar que tipo de información se desea visualizar: <ul style="list-style-type: none"> - Análisis del recinto - Análisis de iluminación del recinto - Fuentes de luz artificial - Tipo de encuesta <ul style="list-style-type: none"> ▪ Aire Acondicionado ▪ Taller y/o Laboratorios ▪ Calentadores y/o Bombas ▪ Misceláneos (equipo de cómputo, equipo de papelería, electrodomésticos). 									
DATOS DE LA ENCUESTA									
Encuesta	Nombre de la Dependencia	Nombre del Edificio	Día	Mes	Hora en la que se elaboro la Encuesta	Año	Estado del Clima	Uso del recinto	Encuesta Completa
001_FI	Facultad de Ingeniería	A	05	11	19:16:00	2012	NUBLADO	Aulas	Ver
002_CELE	CELE	B	05	11	19:16:00	2012	NUBLADO	Aulas	Ver

A continuación se detalla cada módulo del sistema al ingresar como **Encuestador**, tomando en cuenta que este usuario cuenta con menos privilegios que el usuario que ingresa como *Administrador*. Este usuario no tiene permiso de dar de baja a ninguna entidad, es por eso que el módulo **Bajas** no se muestra al inicio de sesión:

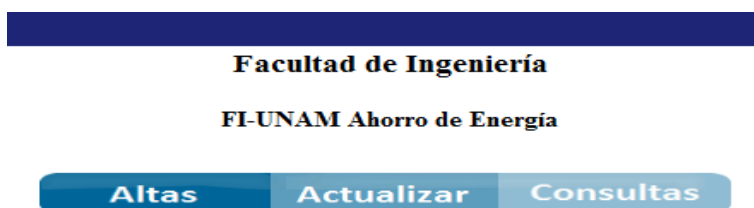


Figura 5.8 Implementación de módulos (Encuestador)

➤ Encuestador

❖ Altas

Se programa el módulo altas en php y html



Figura 5.9 Implementación de Entidades para el módulo de Altas (Encuestador)

Tabla 5.7 Implementación del módulo Altas (Encuestador)

Módulo	Entidad
ALTAS	ENCUESTADOR
<p>Descripción: Al ingresar como Encuestador al sistema, se le solicita al usuario dar de alta sus datos para poder continuar con la encuesta. Es importante que el usuario cuente con un identificador, como su número de cuenta, número de empleado, etc.</p>	

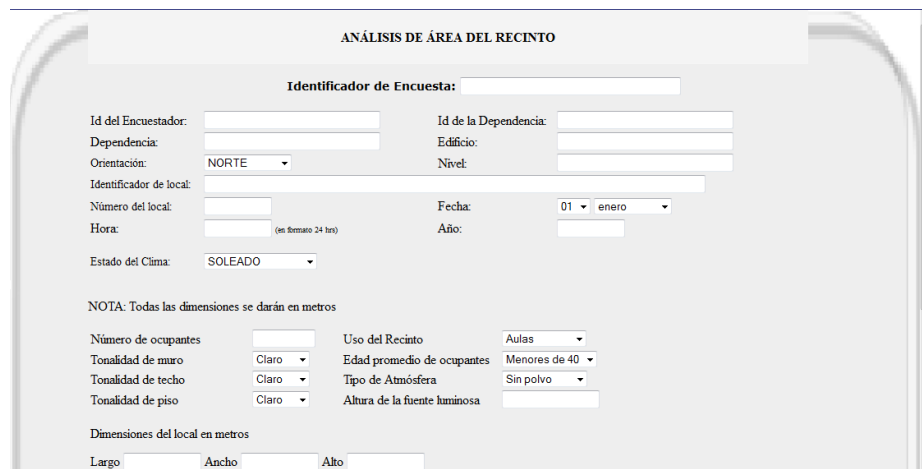
Módulo	Entidad
ALTAS	DEPENDENCIA
<p>Descripción: Esta entidad contiene la información de la dependencia a la que se le realizará el diagnóstico energético. Se necesitará de un identificador de dependencia para contar con una buena organización de la base de datos, así como conocer nombre de edificios, ubicación en la que se encuentra la dependencia, horas de trabajo y número de empleados.</p>	
<div style="background-color: #4a90e2; color: white; padding: 5px; margin-bottom: 10px;">ALTA DE DATOS PARA LA DEPENDENCIA</div> <p>IDENTIFICADOR DE LA DEPENDENCIA: <input type="text"/></p> <p>IDENTIFICADOR DEL EDIFICIO: <input type="text"/></p> <p>DEPENDENCIA: <input type="text"/></p> <p>EDIFICIO: <input type="text"/></p> <p>UBICACIÓN: <input type="text"/></p> <p>HORAS NORMALES: <input type="text"/></p> <p>HORAS LABORABLES: <input type="text"/></p> <p>HORARIO DE COMIDA: <input type="text"/></p> <p>NÚMERO DE EMPLEADOS: <input type="text"/></p> <p>POBLACIÓN FLOTANTE: <input type="text"/></p> <p style="text-align: center;"> <input type="button" value="ALTA"/> <input type="button" value="LIMPIAR"/> </p>	

Módulo	Entidad
ALTAS	EDIFICIO
<p>Descripción: Para dar de alta a esta entidad es necesario conocer el identificador del edificio, así como su nombre, en que dependencia se encuentra el edificio, número de locales, niveles y la orientación en la que se encuentra el edificio.</p>	
<div style="background-color: #4a90e2; color: white; padding: 5px; margin-bottom: 10px;">ALTA DE DATOS PARA EL EDIFICIO</div> <p>IDENTIFICADOR DEL EDIFICIO: <input type="text"/></p> <p>EDIFICIO: <input type="text"/></p> <p>DEPENDENCIA: <input type="text"/></p> <p>NÚMERO DE LOCALES: <input type="text"/></p> <p>NÚMERO DE NIVELES: <input type="text"/></p> <p>NOMBRE DEL LOCAL: <input type="text"/></p> <p>ORIENTACIÓN: <input type="text"/></p> <p style="text-align: center;"> <input type="button" value="ALTA"/> <input type="button" value="LIMPIAR"/> </p>	

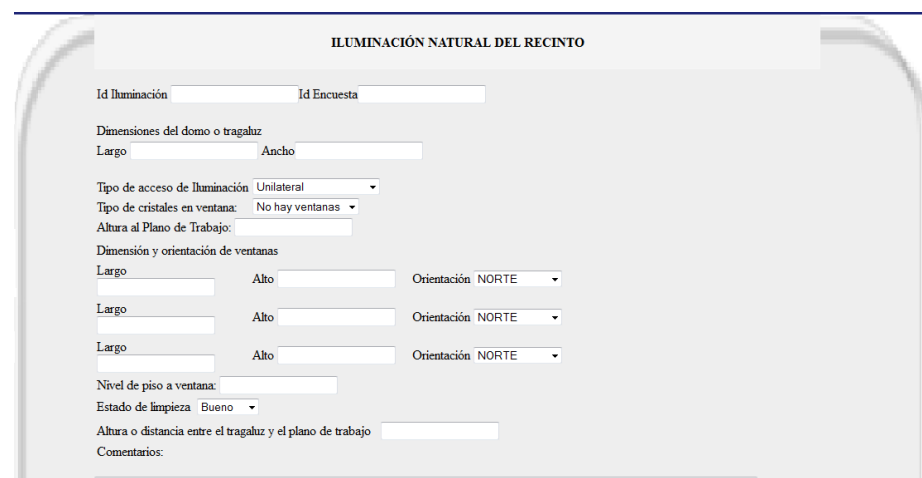
Módulo	Entidad
ALTAS	USUARIO
Descripción: El usuario que ingresa como <i>encuestador</i> al sistema no tiene el permiso de dar de alta a un usuario, esto es por la integridad de datos que requiere el sistema.	

Módulo	Entidad
ALTAS	ENCUESTA
Descripción: En esta entidad, se requiere generar un identificador de encuesta. Esta entidad contiene toda la información para poder realizar el análisis del recinto, análisis de iluminación del recinto, fuentes de luz artificial y el tipo de encuesta.	

Comenzamos con el análisis de área del recinto:



Iluminación natural del recinto



Iluminación artificial del recinto

ILUMINACIÓN ARTIFICIAL DEL RECINTO

Id Luz Artificial Id Iluminacion

Número total de interruptores

Luminarias por interruptor

Horas de uso diarias

Especificación de fuentes de luz artificial

Fluorescentes: No Aplica

Focos Incandescentes: No Aplica

Vapor: No Aplica

Aditivos Metálicos: No Aplica

Lámpara Dicroica: No Aplica

Otro:

Tipo de encuesta a realizar

SELECCIONE LA ENCUESTA QUE DESEA LLENAR

Aire Acondicionado

Taller-Laboratorio

Calentadores y Bombas

Misceláneos

Equipo de Aire acondicionado

ACONDICIONAMIENTO AMBIENTAL

Identificador de Equipo

Identificador de Encuestador

Equipo	Marca	Modelo	Capacidad TR	Antigüedad del equipo [años]	REE	Potencia Max. [kW]	Horas Dia	Meses al año	Días Mes	Descripción	Cantidad
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Equipo de Taller o Laboratorio

TALLER O LABORATORIO															
							Identificador de Equipo								Identificador de Encuestador
Equipo	Cantidad	Potencia [w]	Hora /Dias	kW instalados	Dia /Mes	Mes /Año	Dia /Año de operación	Hora /Año de operación	Hora /Dias distribuidas al año	FC	kW demanda	kWh /Mes	Comentarios		
<input type="button" value="Insertar"/>															

Equipo de bombeo o calefactores

BOMBEO O CALEFACTORES							
			Identificador de Equipo				Identificador de Encuestador
Marca	Potencia demandada (HP)	Potencia demandada (kW)	Mes de uso /Año	Hora /Dia	kWh /mes	Comentarios	
<input type="button" value="Insertar"/>							

Misceláneos

MISCELÁNEOS	
 <p>EQUIPO DE COMPUTO</p> <p>EQUIPO DE PAPELERÍA</p> <p>ELECTRODOMÉSTICOS</p>	

Seleccionando equipo de cómputo

ÉQUIPO DE CÓMPUTO

Equipo de Cómputo SELECCIONA UNA OPCION ▾
SELECCIONA UNA OPCION
COMPUTADORA
IMPRESORA
SCANNER
PLOTTER
REGULADOR
NO-BREAK
UNIDAD CD-RW

Seleccionando electrodomésticos

ALTA DE DATOS PARA ELECTRODOMÉSTICOS

Electrodomésticos SELECCIONA UNA OPCION ▾
SELECCIONA UNA OPCION
Despachador: CAFETERA
CALEFACTOR
ENFRIADOR/CALENTADOR
EQUIPO DE VIDEO
ESTÉREO
GRABADORA
HORNO DE MICROONDAS
PARRILLA ELÉCTRICA
RADIO
RADIO-GRABADORA
REFRIGERADOR
SECADORA DE MANOS
TELÉFONO INALÁMBRICO
TELEVISOR
TOSTADOR
VAPORIZADOR
VENTILADOR
VIDEO VHS

ALTA DE DATOS PARA ELECTRODOMÉSTICOS

Electrodomésticos SELECCIONA UNA OPCION ▾
Despachador: SELECCIONA UNA OPCION ▾
SELECCIONA UNA OPCION
CAFÉ
GOLOSINAS
REFRESCOS

Seleccionando equipo de Papelería

ALTA DE DATOS PARA EL ÉQUIPO DE PAPELERÍA

Equipo de Papelería SELECCIONA UNA OPCION ▾
SELECCIONA UNA OPCION
DESTRUCTOR DE PAPEL
FAX
FOTOCOPIADORA
GUILLOTINA
MÁQUINA DE ESCRIBIR
PROYECTOR
REPRODUCTOR DVD
SACAPUNTAS
SUMADORA
TRITURADOR

❖ Bajas

Tabla 5.8 Implementación del módulo Bajas (Encuestador)

Descripción: El usuario que ingresa como *encuestador al sistema*, no cuenta con el permiso de dar de *baja* a ninguna entidad, asegurando así el control de acceso y la integridad de datos para el correcto funcionamiento del sistema.

❖ Actualizar

Se programa el módulo *actualizar* en php y html



Figura 5.10 Implementación de Entidades para el módulo de Altas (Encuestador)

Tabla 5.9 Implementación del módulo Actualizar (Encuestador)

Módulo	Entidad
ACTUALIZAR	ENCUESTADOR
Descripción: La información del encuestador no puede ser modificada por el usuario que ingrese como <i>encuestador</i> , con esta limitación se asegura la integridad de datos.	

Módulo	Entidad
ACTUALIZAR	DEPENDENCIA
Descripción: Muestra una lista de todas las dependencias registradas en el sistema, una vez seleccionada la dependencia, se proporciona la información que se desea modificar.	

ACTUALIZACION DE DATOS PARA LA DEPENDENCIA

Identificador de Dependencia	Identificador de Edificio	Dependencia	Edificio	Ubicación	Horas Normales	Horas Laborales	Horario de Comida	Número de Empleados	Población Flotante	Actualizar
CELE-B-200	B-200	CELE	B	Ciudad Universitaria	12	8	13:00:00	1234	12345	ACTUALIZA
FI-A-001	A-001	Facultad de Ingeniería	A	Ciudad Universitaria	12	8	13:00:00	1234	12345	ACTUALIZA
FI-K-USECAD	K-USECAD	Facultad de Ingeniería	K	Ciudad Universitaria	12	8	13:00:00	1234	12345	ACTUALIZA

ACTUALIZACION DE DATOS PARA LA SIGUIENTE DEPENDENCIA

IDENTIFICADOR DE LA DEPENDENCIA:

IDENTIFICADOR DEL EDIFICIO:

DEPENDENCIA:

EDIFICIO:

UBICACIÓN:

HORAS NORMALES:

HORAS LABORABLES:

HORARIO DE COMIDA:

NÚMERO DE EMPLEADOS:

POBLACIÓN FLOTANTE:

Módulo	Entidad
ACTUALIZAR	EDIFICIO

Descripción: Muestra una lista de todos los edificios registrados en el sistema, una vez seleccionado el edificio, se proporciona la información que se desea modificar.

ACTUALIZACION DE DATOS DEL EDIFICIO

Identificador del Edificio	Edificio	Dependencia	Número de Locales	Número de Niveles	Nombre del Local	Orientación	Actualizar
A-001	A	Facultad de Ingeniería	50	4	001	Sur	ACTUALIZA
B-200	B	CELE	50	2	200	Sur	ACTUALIZA
K-USECAD	K	Facultad de Ingeniería	5	1	usecad	norte	ACTUALIZA

ACTUALIZACION DE DATOS PARA EL SIGUIENTE EDIFICIO

IDENTIFICADOR DEL EDIFICIO:

EDIFICIO:

DEPENDENCIA:

NÚMERO DE LOCALES:

NÚMERO DE NIVELES:

NOMBRE DEL LOCAL:

ORIENTACIÓN:

Módulo	Entidad
ACTUALIZAR	USUARIO
Descripción: La información del usuario no puede ser modificada por el usuario que ingrese como <i>encuestador</i> , con esta limitación se asegura la integridad de datos.	

Módulo	Entidad																														
ACTUALIZAR	ENCUESTA																														
Descripción: Muestra una lista de todas las encuestas registradas en el sistema, una vez seleccionado la encuesta, se proporciona la información que se desea modificar.																															
<div style="background-color: #ADD8E6; padding: 5px; border: 1px solid black; margin: 10px auto; width: 60%;">ACTUALIZACION DE DATOS DE LA ENCUESTA</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Encuesta</th> <th>Nombre de la Dependencia</th> <th>Nombre del Edificio</th> <th>Día</th> <th>Mes</th> <th>Hora en la que se elaboro la Encuesta</th> <th>Año</th> <th>Estado del Clima</th> <th>Uso del recinto</th> <th>Encuesta Completa</th> </tr> </thead> <tbody> <tr> <td>001_FI</td> <td>Facultad de Ingenieria</td> <td>A</td> <td>05</td> <td>11</td> <td>19:16:00</td> <td>2012</td> <td>NUBLADO</td> <td>Aulas</td> <td>ACTUALIZA</td> </tr> <tr> <td>002_CELE</td> <td>CELE</td> <td>B</td> <td>05</td> <td>11</td> <td>19:16:00</td> <td>2012</td> <td>NUBLADO</td> <td>Aulas</td> <td>ACTUALIZA</td> </tr> </tbody> </table>		Encuesta	Nombre de la Dependencia	Nombre del Edificio	Día	Mes	Hora en la que se elaboro la Encuesta	Año	Estado del Clima	Uso del recinto	Encuesta Completa	001_FI	Facultad de Ingenieria	A	05	11	19:16:00	2012	NUBLADO	Aulas	ACTUALIZA	002_CELE	CELE	B	05	11	19:16:00	2012	NUBLADO	Aulas	ACTUALIZA
Encuesta	Nombre de la Dependencia	Nombre del Edificio	Día	Mes	Hora en la que se elaboro la Encuesta	Año	Estado del Clima	Uso del recinto	Encuesta Completa																						
001_FI	Facultad de Ingenieria	A	05	11	19:16:00	2012	NUBLADO	Aulas	ACTUALIZA																						
002_CELE	CELE	B	05	11	19:16:00	2012	NUBLADO	Aulas	ACTUALIZA																						

❖ Consultas

Se programa el módulo *consultas* en php y html



Figura 5.11 Implementación de Entidades para el módulo de Consultas (Encuestador)

Tabla 5.10 Implementación del módulo Consultas (Encuestador)

Módulo	Entidad
CONSULTAS	ENCUESTADOR
Descripción: La información del <i>encuestador</i> no puede ser consultada por el usuario que ingrese como <i>encuestador</i> , con esta limitación se asegura la integridad y confidencialidad de datos registrados por otros encuestadores o administradores.	

Módulo	Entidad																																								
CONSULTAS	DEPENDENCIA																																								
<p>Descripción: El módulo de consultas se encuentra delimitado, para esta entidad, ya sea por <i>dependencia</i> o por <i>ubicación</i>, sin embargo también se puede realizar la consulta de todos las dependencias del sistema.</p>																																									
<p>The screenshot shows a web interface for searching by dependency. It features a header 'CONSULTAR POR:' with three options: 'DEPENDENCIA', 'UBICACIÓN', and 'TODOS'. The 'DEPENDENCIA' option is selected. Below this, there is a blue bar with the text 'CONSULTA DE DATOS PARA LA DEPENDENCIA'. A search field labeled 'DEPENDENCIA:' is followed by 'ENVIAR' and 'LIMPIAR' buttons. A second blue bar also displays 'CONSULTA DE DATOS PARA LA DEPENDENCIA'. Below this, a search field labeled 'UBICACIÓN:' is followed by 'ENVIAR' and 'LIMPIAR' buttons. At the bottom, a purple bar is labeled 'DATOS DE LA DEPENDENCIA'.</p>																																									
<table border="1"> <thead> <tr> <th>Identificador de Dependencia</th> <th>Identificador de Edificio</th> <th>Dependencia</th> <th>Edificio</th> <th>Ubicación</th> <th>Horas Normales</th> <th>Horas Laborales</th> <th>Horario de Comida</th> <th>Número de Empleados</th> <th>Población Flotante</th> </tr> </thead> <tbody> <tr> <td>CELE-B-200</td> <td>B-200</td> <td>CELE</td> <td>B</td> <td>Ciudad Universitaria</td> <td>12</td> <td>8</td> <td>13:00:00</td> <td>1234</td> <td>12345</td> </tr> <tr> <td>FI-A-001</td> <td>A-001</td> <td>Facultad de Ingeniería</td> <td>A</td> <td>Ciudad Universitaria</td> <td>12</td> <td>8</td> <td>13:00:00</td> <td>1234</td> <td>12345</td> </tr> <tr> <td>FI-K-USECAD</td> <td>K-USECAD</td> <td>Facultad de Ingeniería</td> <td>K</td> <td>Ciudad Universitaria</td> <td>12</td> <td>8</td> <td>13:00:00</td> <td>1234</td> <td>12345</td> </tr> </tbody> </table>		Identificador de Dependencia	Identificador de Edificio	Dependencia	Edificio	Ubicación	Horas Normales	Horas Laborales	Horario de Comida	Número de Empleados	Población Flotante	CELE-B-200	B-200	CELE	B	Ciudad Universitaria	12	8	13:00:00	1234	12345	FI-A-001	A-001	Facultad de Ingeniería	A	Ciudad Universitaria	12	8	13:00:00	1234	12345	FI-K-USECAD	K-USECAD	Facultad de Ingeniería	K	Ciudad Universitaria	12	8	13:00:00	1234	12345
Identificador de Dependencia	Identificador de Edificio	Dependencia	Edificio	Ubicación	Horas Normales	Horas Laborales	Horario de Comida	Número de Empleados	Población Flotante																																
CELE-B-200	B-200	CELE	B	Ciudad Universitaria	12	8	13:00:00	1234	12345																																
FI-A-001	A-001	Facultad de Ingeniería	A	Ciudad Universitaria	12	8	13:00:00	1234	12345																																
FI-K-USECAD	K-USECAD	Facultad de Ingeniería	K	Ciudad Universitaria	12	8	13:00:00	1234	12345																																

Módulo	Entidad
CONSULTAS	EDIFICIO
<p>Descripción: El módulo de consultas se encuentra delimitado, para esta entidad, ya sea por <i>identificador del edificio</i> o por <i>orientación</i>, sin embargo también se puede realizar la consulta de todos los edificios que se dieron de alta en el sistema.</p>	

CONSULTAR POR:

[EDIFICIO](#)

[ORIENTACIÓN](#)

[TODOS](#)

CONSULTA DE DATOS PARA EL EDIFICIO

IDENTIFICADOR DEL EDIFICIO:

ENVIAR LIMPIAR

CONSULTA DE DATOS PARA EL EDIFICIO

ORIENTACIÓN DEL EDIFICIO:

ENVIAR LIMPIAR

DATOS DEL EDIFICIO

Identificador del Edificio	Edificio	Dependencia	Número de Locales	Número de Niveles	Nombre del Local	Orientación
A-001	A	Facultad de Ingeniería	50	4	001	Sur
B-200	B	CELE	50	2	200	Sur
K-USECAD	K	Facultad de Ingeniería	5	1	usecad	norte

Módulo	Entidad
CONSULTAS	USUARIO
Descripción: La información del <i>usuario</i> no puede ser consultada por aquel usuario que ingrese como <i>encuestador</i> , con esta limitación se asegura la integridad y confidencialidad de datos registrados por otros encuestadores o administradores.	

Módulo	Entidad
CONSULTAS	ENCUESTA
Descripción: La consulta se realiza por <i>identificador de la encuesta</i> , una vez seleccionada la encuesta a consultar es necesario seleccionar que tipo de información se desea visualizar:	
<ul style="list-style-type: none"> - Análisis del recinto - Análisis de iluminación del recinto - Fuentes de luz artificial - Tipo de encuesta <ul style="list-style-type: none"> ▪ Aire Acondicionado ▪ Taller y/o Laboratorios ▪ Calentadores y/o Bombas 	
Misceláneos (equipo de cómputo, equipo de papelería, electrodomésticos).	

DATOS DE LA ENCUESTA

Encuesta	Nombre de la Dependencia	Nombre del Edificio	Día	Mes	Hora en la que se elaboro la Encuesta	Año	Estado del Clima	Uso del recinto	Encuesta Completa
001_FI	Facultad de Ingeniería	A	05	11	19:16:00	2012	NUBLADO	Aulas	Ver
002_CELE	CELE	B	05	11	19:16:00	2012	NUBLADO	Aulas	Ver

CAPÍTULO 6 PRUEBAS Y MANTENIMIENTO DEL SISTEMA

Las pruebas de los sistemas se requiere para asegurar que cada componente del sistema esté en operación como debe y que el sistema en su conjunto se desempeñe exactamente de acuerdo con los requerimientos locales específicos, para esto debemos examinar el código de cada módulo del sistema, para garantizar su lógica y que se hayan seguido los estándares debidos de diseño y construcción, aplicar evaluaciones de calidad para determinar si se han satisfecho los criterios de prueba y verificar que lo que entra es lo que sale, introduciendo información conocida y verificando que el resultado sea consecuente con ella.

Después de que el sistema ha sido probado e implantado, se les debe seguir dando mantenimiento para asegurar que continúe operando en el nivel mostrado durante la etapa de prueba. Las rutinas de mantenimiento variarán de acuerdo con el tipo y complejidad de la tecnología.

6.1 Pruebas del sistema

Una vez instalado el sistema se han realizado las siguientes pruebas, estas mismas se realizaron durante la etapa de desarrollo e implementación del mismo:

Tabla 6.1 Pruebas del sistema

Acción	Prueba	Resultado
Ingresar	Escribir la dirección URL del sitio WEB	Visualizar la página principal del nuestro sistema
Autenticar	Ingresar usuario y/o contraseña inválidas	El sistema mandará un mensaje de error, indicando que los datos son incorrectos
	Ingresar usuario y/o contraseña válidas	El sistema mostrará la página principal con las opciones correspondientes al tipo de usuario que está accedando
Ingreso al LEDA	Acceso al menú de opciones según el tipo de usuario	El sistema mostrará la página de menú, donde, dependiendo el usuario, podemos dar de Alta, baja, actualizar o consultar información de la base de datos

Insertar datos	<p>Ingresar datos válidos correspondientes al tipo de dato</p> <p>Ingresar datos que no sean válidos dentro del sistema</p> <p>No ingresar datos</p> <p>Llenar todos los campos y seleccionar <i>limpiar</i></p>	<p>El sistema mandará un mensaje haciéndole saber al usuario que los datos fueron dados de alta correctamente</p> <p>El sistema envía un mensaje advirtiendo al usuario que los datos son incorrectos, favor de verificar</p> <p>El sistema envía advertencia de campos vacíos</p> <p>El sistema limpiará todos los campos existentes</p>
Actualizar datos	<p>Agregar información de encuestadores, usuarios, dependencias, edificios y/o encuestas, donde los campos se encontraban vacíos o erróneos</p>	<p>El sistema desplegará una lista de todos aquellos registros existentes en la base de datos, al seleccionar <i>actualizar</i> nos mostrará los datos y podremos entonces actualizar la información de encuestadores, usuarios, dependencias, edificios y/o encuestas</p>
Consultar información	<p>Consultar información de encuestadores, usuarios, dependencias, edificios y/o encuestas</p>	<p>El sistema desplegará una lista de todos aquellos registros existentes en la base de datos, según la consulta que se está realizando, ya sea de encuestadores, usuarios, dependencias, edificios y/o encuestas</p>
Visualizar encuesta	<p>Consulta de datos de encuesta</p>	<p>El sistema desplegará una lista de todas aquellas encuestas existentes en la base de datos, al seleccionar <i>ver</i>, se mostrará un documento en PDF con toda la información que se realizó durante el LEDA</p>
Baja de información	<p>Eliminar registros</p> <p>Cancelar Baja de datos</p>	<p>El sistema desplegará una lista de todos aquellos registros existentes en la base de datos, además de mandar un mensaje de confirmación de la baja del registro y realizará la baja</p> <p>El sistema cancelará la acción de baja de registro</p>
Salir	<p>Cerrar sesión</p>	<p>El sistema mostrará un mensaje diciendo que ha salido del sistema y después se redireccionará a la página principal del sistema.</p>

6.2 Mantenimiento del DBMS

Se han insertado en la base de datos todos los registros que ya se tenían, se han actualizado gran parte de ellos y se han dado de alta algunos registros más.

Una vez realizado el Levantamiento de Datos (LEDA) en la base de datos, se realizaron nuevamente las pruebas de la tabla. Logrando ver un buen funcionamiento y desempeño comparando la información contenida en la base de datos y efectuando de manera manual las consultas sql.

Sin embargo la mejor práctica de mantenimiento para comprobar que el sistema funciona de manera correcta, es utilizándolo, lo cual se ha hecho desde su momento de instalación, ya que a diario se realizan altas, bajas, actualizaciones y consultas, es decir, el sistema se encuentra trabajando las veinticuatro horas los 7 días de la semana (24x7), como se había planeado desde un principio y no ha presentado ningún tipo de anomalías, por lo que se puede concluir que la etapa de mantenimiento ha sido exitosa.

En el momento en el que se registra el defecto, se procede a diagnosticar de qué tipo de mantenimiento se trata. Atendiendo a los fines, podemos establecer los siguientes tipos de mantenimiento [28]:

- **Correctivo:** son aquellos cambios precisos para corregir errores del producto software.
- **Evolutivo:** son las incorporaciones, modificaciones y eliminaciones necesarias en un producto software para cubrir la expansión o cambio en las necesidades del usuario.
- **Adaptativo:** son las modificaciones que afectan a los entornos en los que el sistema opera, por ejemplo, cambios de configuración del hardware, software de base, gestores de base de datos, comunicaciones, etc.
- **Perfectivo:** son las acciones llevadas a cabo para mejorar la calidad interna de los sistemas en cualquiera de sus aspectos: reestructuración del código, definición más clara del sistema y optimización del rendimiento y eficiencia.

Una vez elegido el tipo de mantenimiento que se efectuará, se realiza la práctica de mejoras posibles al sistema para el buen funcionamiento del mismo:

- Corrigiendo errores surgidos durante las pruebas al sistema, estos errores surgieron en base al valor de los atributos de la base de datos. Al realizar el alta y consulta de información, la base de datos arrojaba *información nula*, es decir, espacios en blanco. Fue entonces que se revisó la estructura de la base de datos y se cambió el tipo de atributos y su longitud.
- Se aplicó un juego de datos de prueba al sistema, con esto se pueden descubrir defectos impredecibles y no deseables que influirán sobre el funcionamiento y el rendimiento global. Es por esto que antes de realizar algún cambio de versión de software, se ejecutan los datos y se prueban en línea, para así poder diagnosticar cual es la falla y en donde se aloja.

- La información es actualizada cada vez que se ingresa al sistema, así se evita algún error de captura.

El sistema es adaptable a cualquier necesidad que surja. El administrador de la base debe analizar las nuevas necesidades y volver a las fases adecuadas del análisis del diseño y a la implementación del sistema.

6.2.1 Políticas de mantenimiento de la Base de Datos

El monitoreo permanente del sistema necesita ser sistematizado para asegurar que las necesidades de mantenimiento sean identificadas y satisfechas cuando resulte necesario. Es necesario establecer un mecanismo para recibir retroalimentación de los usuarios como otra forma de determinar las necesidades de mantenimiento y modificación.

Cuando se realicen modificaciones al equipo, programa o comunicaciones como resultado de programas de mantenimiento o actualización, puede ser necesario promover rondas adicionales de verificación y prueba del sistema para asegurarse que sigue cumpliendo las normas exigidas.

Los pasos que se requieren para efectuar el mantenimiento del sistema consta de cinco pasos [28]:

1) Analizar las solicitudes de mejora.

El propósito de esta actividad es determinar el curso apropiado de acciones para tratar nuevos problemas de ideas de mejoras, problemas o limitaciones técnicas (resultante de otras actividades de soporte). Esta fase de soporte, en general, no sirve en realidad para mejorar el sistema, no estudia la documentación existente para determinar el curso apropiado de acciones. Sobre la base análisis de los modelos del sistema actual, estas acciones pueden incluir:

- Nuevas necesidades de empresa y volver al Análisis de Sistemas.
- Nuevas necesidades técnicas y volver al Diseño de Sistema.
- Nuevas necesidades de programas y proceder a la tarea 2.

2) Escribir nuevos programas sencillos.

Esto quiere decir que estos programas pueden conseguirse rápidamente mediante la estructura de nuevos programas sencillos. Los programas sencillos son aquellos que no actualizan datos existentes y no introducen nuevos datos (por motivo de almacenamiento de datos). Las necesidades de nuevos programas conforman la mayoría de las mejoras que se requieren hoy en día.

3) Reestructurar archivos o bases de datos.

El administrador del sistema colabora en la reingeniería de archivos y bases de datos. La tecnología actual de base de datos más idónea es la base de datos relacionadas con SQL (que

almacenan los datos en tablas integradas por medio de campos redundantes que actúan como punteros).

4) **Analizar la biblioteca de programas y los costos de mantenimiento:**

La primera actividad requerida para lograr este objetivo es analizar la biblioteca de programas y los costos de mantenimiento. Esta actividad casi siempre requiere de software capaz de llevar a cabo el análisis.

La métrica de Software: Es un conjunto de medidas matemáticamente probadas sobre la calidad y la productividad del software.

Ejemplos de métricas de software aplicables al mantenimiento son:

- **Nudo de flujo de control**, o número de veces que se cruzan entre sí los caminos lógicos. En términos ideales, un programa debería tener cero nudos de flujo de control.
- **Complejidad de los Ciclos**, o números de caminos únicos a través de un programa. En términos ideales, cuántos menos sean mejor.

5) **Hacer reingeniería y pruebas de los programas.**

Existen tres tipos de reingeniería que pueden aplicarse sobre dicho sistema:

» **La reorganización de código:** Reestructurar la organización modular y/o lógica del sistema. La lógica puede reestructurarse para eliminar nudos de flujos de control y reducir la complejidad de los ciclos.

» **La Conversión de código:** Traducir el código de un lenguaje a otro. Típicamente, esta traducción se realiza de una a otra versión de un mismo lenguaje.

El programa candidato para reingeniería se copia desde la biblioteca de programas. La reingeniería se hace por medio del empleo de uno o más de los métodos anteriores. Los nuevos modelos de datos, procesos y/o redes se actualizan en el diccionario.

6.2.2 Políticas de respaldo y recuperación de información

Los sistemas de información, se enfrentan a varios riesgos, uno de los más comunes son las interrupciones. Las interrupciones se presentan de formas muy variadas: virus informáticos, fallos de electricidad, errores de hardware y software, caídas de red, hackers, errores humanos, incendios, inundaciones, etc. Y aunque no se pueda prevenir cada una de estas interrupciones, un administrador de sistemas sí puede prepararse para evitar las consecuencias que éstas puedan tener sobre su proyecto. Del tiempo que tarde en reaccionar una empresa dependerá la gravedad de sus consecuencias.

Cuando hablamos de respaldar la información significa copiar el contenido lógico de nuestro sistema informático a un medio que cumpla con los siguientes requerimientos:

1. Ser confiable: Minimizar las probabilidades de error. El respaldo del sistema podría ser copiado en otro servidor ya que muchos medios magnéticos como las cintas de respaldo, memorias USB, o discos duros tienen probabilidades de error o son particularmente sensibles a campos magnéticos y podrían ser fatales para lograr nuestro objetivo.

2. Estar fuera de línea, en un lugar seguro: Tan pronto se realiza el respaldo de información, el soporte que almacena este respaldo debe ser desconectado de la computadora y almacenado en un lugar seguro tanto desde el punto de vista de sus requerimientos técnicos como humedad, temperatura, campos magnéticos, como de su seguridad física y lógica.

3. La forma de recuperación sea rápida y eficiente: Es necesario probar la confiabilidad del sistema de respaldo no sólo para respaldar sino que también para recuperar.

Esto nos lleva a que un sistema de respaldo y recuperación de información tiene que ser probado y eficiente.

Una vez definidas y establecidas las características fundamentales de la implementación del sistema y establecidos los planes de copias de seguridad y recuperación de la información, aseguramos que nuestro sistema está protegido y podremos recuperar la información del respaldo Full que se realizó.

El sistema se basa en el lenguaje sql usando el administrador PhpMyAdmin, entonces para realizar el respaldo de la base de datos del sistema SisFIUnamAhorroDeEnergía, simplemente se exporta la base de datos, el servidor exporta la base de datos a un archivo de texto con extensión .sql que el navegador recibirá para poder guardarla en cualquier equipo de cómputo o incluso en una memoria de almacenamiento masivo, siempre y cuando este sea confiable.

Para la recuperación de la información, se importa el archivo de texto guardado con extensión .sql en el administrador PhpMyAdmin.

También es posible respaldar y recuperar la información del sistema y de la base de datos desde Shell:

Respaldo y Restauración MySQL de Manera Local.

Para hacer un respaldo de una base de datos MySQL desde una consola o mediante comandos shell se puede usar el comando mysqldump como se ejemplifica en la siguiente liga.

Comando: `mysqldump -u "usuario" -p"contraseña" nombre-de-la-base-de-datos > nombre-del-respaldo.sql`

NOTA: Las comillas deben omitirse tanto en el usuario como en la contraseña.

Para restaurar un respaldo de una base de datos MySQL se usa el siguiente comando

Comando: `mysql -u "usuario" -p"contraseña" nombre-de-la-base-de-datos < nombre-del-respaldo.sql`

NOTA: Al igual que en el ejemplo anterior las comillas deben omitirse tanto en el usuario como en la contraseña.

Respaldo y Restauración MySQL de Manera Remota.

Para Respaldo o Restaurar una Base de datos remota se utilizan los mismos comandos que de manera local, con la única diferencia de agregar la opción "-h" con la cual se especifica el nombre o dirección del host en donde se encuentra la base.

Para Respaldo usamos:

Comando: `mysqldump -u "usuario" -p"contraseña" -h"nombre-o-dirección-del-host" nombre-de-la-base-de-datos > nombre-del-respaldo.sql`

Para restaurar utilizamos:

Comando: `mysql -u "usuario" -p"contraseña" -h"nombre-o-dirección-del-host" nombre-de-la-base-de-datos < nombre-del-respaldo.sql`

6.2.3 Políticas de roles a usuarios

El alcance de estas políticas incluye a todo usuario del sistema que tenga un rol, cuyas actividades sean de administración del Sistema FIUnamAhorroDeEnergía, o cualquier otro acceso que sí esté permitido.

Políticas de información para el acceso

1. El uso de la cuenta de usuario es responsabilidad de la persona a la que está asignada. La cuenta es para uso personal e intransferible.
2. La cuenta de usuario se protegerá mediante una contraseña.
3. Las cuentas de usuario (usuario y contraseña) son sensibles a mayúsculas y minúsculas, es decir que estas deben ser tecleadas como están.
4. No compartir la cuenta de usuario con otras personas: compañeros de trabajo, amigos, familiares, etc.
5. Si se detecta o sospecha que las actividades de una cuenta de usuario puede comprometer la integridad y seguridad de la información, el acceso a dicha cuenta será suspendido temporalmente y será reactivada sólo después de haber tomado las medidas necesarias a consideración del Administrador del Sistema.
6. Tipos de Cuentas de Usuario

Para efectos de las presentes políticas, se definen dos tipos de cuentas de usuario:

a) Cuenta de Encuestador de Sistema de Información: todas aquellas cuentas que sean utilizadas por los encuestadores para acceder al sistema y realizar el Levantamiento de Datos (LEDA). Estas cuentas permiten el acceso para altas, consulta y actualización de información, con respectivas restricciones, las cuales se encuentran reguladas por los roles de usuario del Sistema.

b) Cuenta de Administración de Sistema de Información: corresponde a la cuenta de usuario que permite al administrador del Sistema realizar tareas específicas de usuario a nivel directivo, como por ejemplo: altas/actualizar/bajas de cuentas de usuario del sistema, además de las respectivas a las encuestas.

7. Todas las contraseñas para acceso al Sistema FI-UNAM Ahorro de Energía con carácter administrativo deberán ser cambiadas al menos cada 6 meses.
8. Todas las contraseñas para acceso al Sistema FI-UNAM Ahorro de Energía de nivel usuario deberán ser cambiadas al menos cada 12 meses.
9. Todas las contraseñas deberán ser tratadas con carácter confidencial.
10. Se evitará el activar o hacer uso de la utilidad de "Recordar Contraseña" o "Recordar Password" de las aplicaciones.

Políticas de información para el levantamiento de datos

1. Una vez ingresando al sistema se captura toda la información que solicita cada formato.
2. Ningún campo de la encuesta deberá quedar en blanco.
3. Los formatos de encuesta tienen un seguimiento que deberá de conservarse, pues de lo contrario el sistema cierra sesión del encuestador.
4. La captura de información debe de ser verídica y de ser posible verificar valores del consumo energético en los equipos que se almacenan en cada recinto.
5. Se deberá seguir la nomenclatura de cada identificador de encuesta, estos datos se detallan en el capítulo IV, tema 4.1
6. Al momento de terminar la encuesta, es imprescindible cerrar sesión.

6.3 Manual sistema FI-UNAM Ahorro de Energía

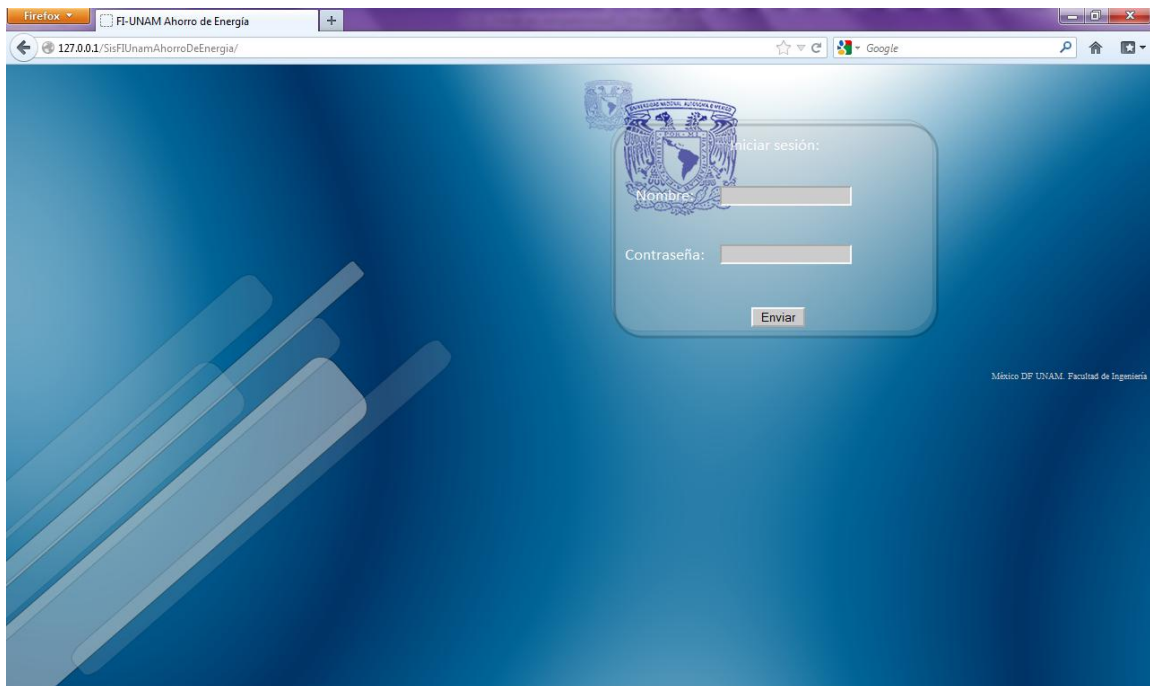
El Sistema FI-UNAM Ahorro de Energía ha sido desarrollado para llevar a cabo el levantamiento de datos para el análisis de consumo de energía en los edificios de Ciudad Universitaria.

Ingreso al sistema

El ingreso al Sistema FI-UNAM Ahorro de Energía lo podemos realizar a través de la web, en la dirección

<http://www.SisFIUnamAhorroDeEnergia>

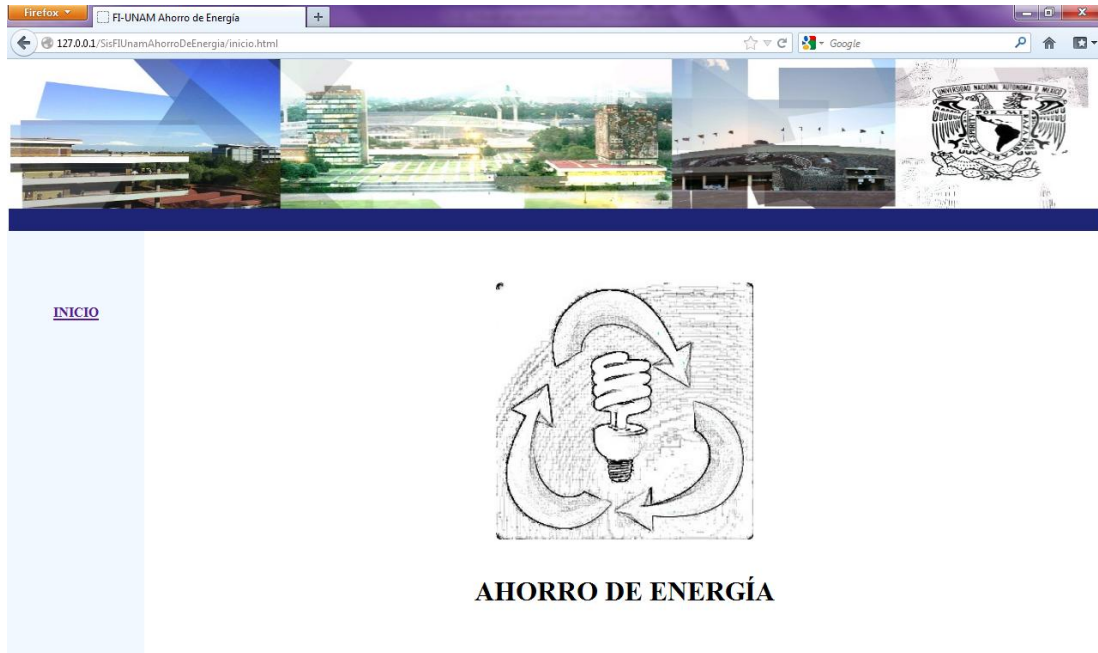
Para poder entrar al Sistema es necesario que el usuario ingrese en el formulario que se muestra en la página de acceso el nombre y contraseña que se le fue asignado por el administrador. Para continuar es necesario presionar el botón “Enviar”.



Al momento de enviar los datos, se visualiza la pantalla de autenticación de usuario:



Una vez autenticando usuario, se visualiza la pantalla de inicio



Cuando nos direccionamos al inicio de nuestro sistema nos muestra la siguiente pantalla



“MENÚ” nos despliega el menú inicial de nuestro sistema, donde veremos las funciones que podemos realizar al sistema.

“SALIR” cierra nuestra sesión, y nos regresa al inicio de nuestro sistema.

Facultad de Ingeniería

FI-UNAM Ahorro de Energía

Altas

Bajas

Actualizar

Consultas

Menú de las opciones

“Altas” nos despliega el listado de datos que podemos ingresar al sistema, el botón “regresar” nos direcciona al menú del sistema.

“Bajas” nos despliega el listado de datos que podemos dar de baja del sistema

“Actualizar” nos despliega el listado de datos que podemos actualizar al sistema

“Consultas” nos despliega el listado de datos que podemos consultar del sistema



Altas

Es necesario conocer el tipo de formato que lleva cada uno de las opciones que nos muestra el sistema para introducir identificador correcto.

El identificador de encuestador y/o usuario será su número de cuenta y/o clave de académico.

Para el caso del identificador de la dependencia deberá llevar el siguiente formato:

FI-A-001

iniciales de la dependencia – nombre del edificio – número del local.

Para el caso del identificador del edificio deberá llevar el siguiente formato:

A-001

nombre del edificio – número del local.

Para el caso del identificador de la encuesta deberá llevar el siguiente formato:

001-FI

número de encuesta (3digitos)- iniciales de la dependencia

Para el caso del identificador de la iluminación del recinto deberá llevar el siguiente formato:

001-ilu

número de encuesta (3digitos)- iluminación

Para el caso del identificador de la iluminación artificial del recinto deberá llevar el siguiente formato:

001-artificial

número de encuesta (3digitos)- artificial

Para el caso del identificador del aire acondicionado, taller, calentadores y misceláneos, deberá llevar el siguiente formato:

001-aire/taller/cal/misc

número de encuesta (3digitos)- tipo de dato

Para realizar un alta, es necesario introducir todos los campos correspondientes y oprimir el botón "ALTA".

ALTA DE DATOS PARA EL ENCUESTADOR

IDENTIFICADOR DEL ENCUESTADOR:

APELLIDO PATERNO:

APELLIDO MATERNO:

NOMBRE(S):

TELÉFONO:

EMAIL:

LOGIN:

CONTRASEÑA:

TIPO:

TIPO DE PRESTADOR:

Si el alta fue procesada sin problemas, se mostrará en pantalla la siguiente pantalla de confirmación.

EL USUARIO FUE DADO DE ALTA CORRECTAMENTE

Consultas

Para poder visualizar nuestro nuevo registro nos direccionamos al botón de "consultas"

CONSULTAR POR:

[IDENTIFICADOR DEL ENCUESTADOR](#)

[TIPO DE USUARIO](#)

[TODOS](#)

Donde podemos consultar al encuestador por medio de su identificador o por tipo de usuario. Seleccionando el botón *actualizar*, realizamos la función hacia la base de datos.

CONSULTA DE DATOS PARA EL ENCUESTADOR

IDENTIFICADOR DEL ENCUESTADOR:

Por identificador de encuestador nos traerá solo la información que contenga ese registro

Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	e-mail	Teléfono	Login	Contraseña	Tipo de Prestador
302166239	Pozas	Montoya	Karla Jossavet	USUARIO	joss_kar@hotmail.com	5518188499	joss	JOSS	SERVICIO SOCIAL

CONSULTA DE DATOS PARA EL ENCUESTADOR

TIPO DE USUARIO:

La información que nos mostrará será todo aquel que este dado de alta como *usuario*

Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	e-mail	Teléfono	Login	Contraseña	Tipo de Prestador
123456789	Pozas	Montoya	Alejandro	USUARIO	mapm_pumas@hotmail.com	5549281506	alex	ALEX	SERVICIO SOCIAL
302166239	Pozas	Montoya	Karla Jossavet	USUARIO	joss_kar@hotmail.com	5518188499	joss	JOSS	SERVICIO SOCIAL

Cuando queremos consultar la información de todos los encuestadores que se encuentran en la base de datos se elige el botón *todos*.

Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	e-mail	Teléfono	Login	Contraseña	Tipo de Prestador
123456789	Pozas	Montoya	Alejandro	USUARIO	mapm_pumas@hotmail.com	5549281506	alex	ALEX	SERVICIO SOCIAL
302166239	Pozas	Montoya	Karla Jossavet	USUARIO	joss_kar@hotmail.com	5518188499	joss	JOSS	SERVICIO SOCIAL
405023822	Rendon	Salgado	Axel	ADMINISTRADOR	axe12424_2@hotmail.com	5523254209	ax	AX	TRABAJADOR

Actualizar

Para actualizar la información del encuestador basta con ingresar al botón actualizar del menú inicial, y nos mostrará una pantalla con los datos existentes en la base de datos

ACTUALIZACIÓN DE DATOS PARA EL ENCUESTADOR										
Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	E-Mail	Numero de Teléfono	Login	Contraseña	Tipo de Prestador	Actualizar
123456789	Pozas	Montoya	Alejandro	USUARIO	mapm_pumas@hotmail.com	5549281506	alex	ALEX	SERVICIO SOCIAL	ACTUALIZA
302166239	Pozas	Montoya	Karla Jossavet	USUARIO	joss_kar@hotmail.com	5518188499	joss	JOSS	SERVICIO SOCIAL	ACTUALIZA
405023822	Rendon	Salgado	Axel	ADMINISTRADOR	axe12424_2@hotmail.com	5523254209	ax	AX	TRABAJADOR	ACTUALIZA

Al elegir actualizar en alguno de los registros, el sistema muestra la siguiente pantalla

ACTUALIZAR DATOS DEL ENCUESTADOR

Cambiamos el registro de *nombre(s)*

Manuel Alejandro

IDENTIFICADOR DEL ENCUESTADOR: 123456789
APELLIDO PATERNO: Pozas
APELLIDO MATERNO: Montoya
NOMBRE(S): Alejandro
TELÉFONO: 5549281506
EMAIL: mapm_pumas@hotmail.com
LOGIN: alex
CONTRASEÑA: ALEX
TIPO: USUARIO
TIPO DE PRESTADOR: SERVICIO SOCIAL

ACTUALIZAR

LIMPIAR

Muestra toda la información del usuario y así podremos solo actualizar la información que se requiere, una vez que la información este actualizada, seleccionamos el botón *actualizar* para que la nueva información se refleje en la base de datos.

El sistema nos envía un mensaje de confirmación.

LA INFORMACIÓN DEL ENCUESTADOR FUE ACTUALIZADA CORRECTAMENTE

Aceptar

Realizamos nuevamente nuestra consulta por identificador para verificar los datos, ya que solo deseamos observar este registro.

CONSULTA DE DATOS PARA EL ENCUESTADOR

IDENTIFICADOR DEL ENCUESTADOR: 123456789

ENVIAR

LIMPIAR

Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	e-mail	Teléfono	Login	Contraseña	Tipo de Prestador
123456789	Pozas	Montoya	Manuel Alejandro	USUARIO	mapm_pumas@hotmail.com	5549281506	alex	ALEX	SERVICIO SOCIAL

Bajas

Para solicitar la baja de algún registro, es necesario ingresar al botón de *bajas* que se encuentra en el *menú* de nuestro sistema y elegir el registro que queremos dar de baja.

BAJA DE DATOS PARA EL ENCUESTADOR					
Cuenta del Encuestador	Apellido Paterno	Apellido Materno	Nombre(s)	Tipo de Usuario	Baja
123456789	Pozas	Montoya	Manuel Alejandro	USUARIO	BORRAR
302166239	Pozas	Montoya	Karla Jossavet	USUARIO	BORRAR
405023822	Rendon	Salgado	Axel	ADMINISTRADOR	BORRAR

Nos envía un mensaje de confirmación

¿Realmente deseas dar de baja a este Encuestador?

Al elegir el botón *aceptar*, confirmamos la petición, la baja se procesa con éxito y se eliminan automáticamente los datos dentro de la tabla que contiene el registro.

EL ENCUESTADOR FUE DADO DE BAJA CORRECTAMENTE

Y nos envía un mensaje de confirmación.

Podemos realizar una consulta por identificador de encuestador para verificar la baja del registro.

CONSULTA DE DATOS PARA EL ENCUESTADOR

IDENTIFICADOR DEL ENCUESTADOR:

Muestra los campos del registro vacíos, es decir, ya no se encuentra ningún usuario con ese identificador único.



Cuenta del Encuestador Apellido Paterno Apellido Materno Nombre(s) Tipo de Usuario e-mail Teléfono Login Contraseña Tipo de Prestador

[INICIO](#)

Cada uno de los registros a llenar durante el Levantamiento de Datos es muy similar a este proceso de encuestador.

En esta ocasión se ingresó con autenticación de administrador, ya que únicamente el administrador del sistema puede ingresar datos de otro encuestador y dar de alta a otros usuarios, a diferencia del administrador, el encuestador no puede dar de baja ningún registro como a continuación se muestra.

Al ingresar como encuestador la primer pantalla que nos muestra el sistema es:

The screenshot shows a dark blue header with the text "Facultad de Ingeniería" and "FI-UNAM Ahorro de Energía". Below this is a light blue banner with the text "Favor de ingresar tus datos para iniciar con la encuesta". In the center, there is a button labeled "Encuestador" which is highlighted with a red rectangular box. In the bottom right corner, there is a "SALIR" link and a power icon.

La primera acción del encuestador es registrarse, seleccionando el botón *encuestador*

The screenshot shows a registration form titled "ALTA DE DATOS PARA EL ENCUESTADOR". The form contains the following fields and values:

IDENTIFICADOR DEL ENCUESTADOR:	302166239
APELLIDO PATERNO:	Pozas
APELLIDO MATERNO:	Montoya
NOMBRE(S):	Karla Jossavet
TELÉFONO	5518188499
EMAIL:	joss_kar@hotmail.com
LOGIN:	joss
CONTRASEÑA:	••••
TIPO:	USUARIO
TIPO DE PRESTADOR:	SERVICIO SOCIAL

At the bottom of the form, there are two buttons: "ALTA" and "LIMPIAR".

Se llenan los campos de igual manera y al momento de seleccionar el botón *alta*, ingresamos al menú inicial de nuestro sistema, donde nos elimina la función *bajas* y elimina la opciones de encuestador y usuarios.

CAPÍTULO 7 CONCLUSIONES

El Sistema de Administración de Bases de Datos sobre el Comportamiento del Consumo de Energía en Edificios de C.U. (SisFI-UNAM Ahorro de Energía) cumple en su totalidad con los requerimientos establecidos en un principio. Este sistema permitirá el proceso del Levantamiento de datos (LEDA) de encuestas, para después realizar el análisis del consumo energético en Ciudad Universitaria, dicho proceso podrá realizarse vía internet a través de una página web desde cualquier explorador de forma segura y en tiempo real, es decir, que se accede a la base de datos permitiendo una actualización de manera inmediata de la información, así como la consulta de la misma.

El sistema se encuentra trabajando las veinticuatro horas los 7 días de la semana (24x7), sin reporte de anomalías.

La interfaz que presenta el SisFI-UNAM Ahorro de Energía, ha resultado ser bastante amigable para los usuarios, tanto para quienes administrarán el sistema, como para quienes realizarán la encuesta. Dichos usuarios pueden realizar las funciones para dar de alta, baja, actualizar y consultar la información de la base de datos de manera fácil, rápida y segura.

La estructura interna de la base de datos es totalmente portable, por lo que resulta de suma importancia, ya que el SisFI-UNAM Ahorro de Energía pretende ser usado no nada más en Ciudad Universitaria, si no fuera de ella, por la información tan general que maneja puede ser utilizado en cualquier otra dependencia, ya sea gubernamental o privada.

Debido a que la aplicación se basa en el patrón MVC, podemos separar los módulos de modelo, vista y de controlador, de tal forma que las modificaciones correspondientes se realizan de manera independiente, lo cual es benéfico para el sistema ya que las modificaciones no alteran ninguna parte del mismo.

El diseño del sistema se realizó pasando por pruebas para la detección de errores en diseño y codificación, con ello se garantiza un buen funcionamiento y así obtener los resultados esperados.

7.1 Dificultades en el desarrollo

Para lograr los objetivos establecidos con anterioridad, se desarrolló un proceso de análisis de los requerimientos y necesidades del sistema tanto de hardware como de software, el cual debería complementarse en cada módulo de programación, más adelante se elaboró un diseño adecuado que cumpla con las necesidades de los usuarios finales para el proceso del LEDA brindando seguridad y disponibilidad.

Durante el proceso de análisis de requerimientos se tuvieron que detallar muchos de los conceptos que se utilizan en el área de electricidad, para así entender qué es lo que se pretende de un diagnóstico energético y poder brindar un mayor desempeño del sistema.

7.2 Conocimientos adquiridos

Se necesitó de la aplicación de conocimientos teóricos así como de experiencia práctica. Fue una experiencia bastante gratificante ya que al momento de realizar este proyecto, en cada paso del desarrollo e implementación, reafirmé los conocimientos adquiridos durante la carrera y pude también aplicarlos, así como también obtener ganancia del mismo, pues se hizo mucho trabajo de investigación sobre las diferentes áreas posibles de aplicación del sistema.

En este proyecto se trabajó con software libre y fue necesario investigar para poder usarlo y sacar provecho al máximo todas las ventajas que el software libre nos ofrece, esto requiere de disposición y empeño para aprender de todas las herramientas que se utilizaron para la implementación del sistema.

Trabajar en este tema de tesis ha sido una gran experiencia que me ha dejado una gran satisfacción y muchos conocimientos tanto teóricos como prácticos que han contribuido a mi desarrollo y crecimiento tanto profesional como personal.

7.3 Tiempo

El planteamiento de la situación actual del problema fue crucial y una de las etapas en las que se invirtió más tiempo fue en el análisis de requerimientos, ya que no se entendía el proceso de un análisis de diagnóstico energético, pues se ingresaba a un área distinta y no tenía conocimiento de ciertas variables que se manejan para tal proceso, por lo que se tuvieron que realizar varias citas con el cliente para interrogar y hacer notables las inquietudes surgidas, sin embargo siempre surgieron inconvenientes, pues en cada cita concretada, el cliente sumaba más requisitos a los solicitados. Al final hubo un común acuerdo para poder seguir adelante con el diseño e implementación del sistema.

7.4 Costo

En la actualidad, este tipo de sistemas se utilizan para el control y la administración de la información, ya que haciendo uso de la tecnología, internet y las bases de datos, se pueden construir sistemas y aplicaciones que respondan a las necesidades específicas de un cliente.

Es por esta razón que el costo de un sistema de administración de bases de datos depende de varios factores, tanto de calidad, confiabilidad, seguridad y funcionalidad que ofrece. Por ejemplo, un SGBD para un ordenador personal puede costar \$8,000, mientras que un SGBD para un sistema multiusuario que dé servicio a cientos de usuarios puede costar entre \$150,000 y \$1,600,000. Además, hay que pagar una cuota anual de mantenimiento que suele ser un porcentaje del precio del SGBD.

En caso de este sistema contamos con todos estos factores y la funcionalidad que ofrece es mayor, ya que hoy en día la situación en la que se vive para realizar el LEDA, nos es muy confiable, pues se realiza en papel, y no cuenta con mucha seguridad, este sistema apoya mucho ya que se genera de forma más rápida, cuenta con la consistencia de datos, ahorra al usuario final detalles acerca del almacenamiento físico de los datos y además de contar con la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

Por lo que este sistema no tiene costo para nuestra máxima casa de estudios, sin embargo sí se planea vender este sistema al sector gubernamental o al sector privado, el costo del sistema se encontrará en un rango entre \$20,000 y \$80,000 dependiendo el sector de venta y la necesidad del cliente. Este precio se justifica, el software utilizado para el sistema es libre, es decir no tiene costo y se puede descargar de la red, sin embargo, el costo que se le da al proyecto es por la necesidad de los distintos sectores que pueden costear un sistema como este para conocer cuanta energía están generando, es decir, este sistema apoya con la realización de los diagnósticos energéticos de dependencias, y con esto se puede reducir el gran consumo de energía basándose en el análisis de sus locales, aulas, recintos, etc.

Sin embargo, el hecho de implementar un proyecto que se ha diseñado y desarrollado con tanto interés y esfuerzo es la mejor recompensa que se obtiene ya que todo aquello que se puso en papel ahora es un hecho y puede ser utilizado por muchas personas, además, con este sistema se cumple la misión y visión de la Facultad de Ingeniería, donde la misión es formar de manera integral recursos humanos en Ingeniería, realizar investigación acorde con las necesidades de la sociedad, y difundir ampliamente la cultura nacional y universal, y la visión ha sido y deberá ser la institución líder en la formación de profesionales en ingeniería del país; semillero fundamental donde se generan nuevos conocimientos al realizar investigación que impacte en el óptimo desarrollo nacional, con aportaciones a la cultura y al desarrollo de capacidades con sentido humanista, social y ecológico; por ello, sus profesionales deberán estar permanentemente actualizados gracias a la sólida oferta brindada a través de una educación continua y a distancia.

ANEXOS

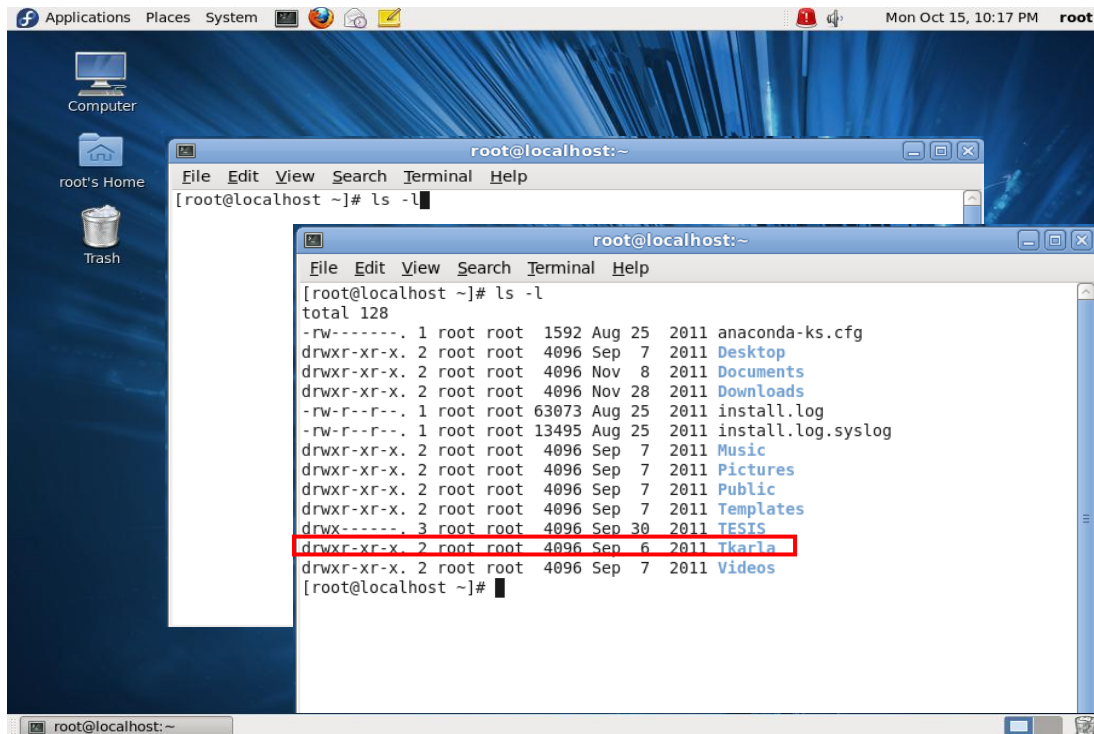
Implementación del sistema

Este proyecto se instaló en un sistema LAMP como ya se mencionó en el capítulo III, para este sistema es importante conocer la política de permisos en Linux. Ya que la administración de usuarios en Linux es la referida a la creación y administración de las cuentas de usuarios, grupos de usuarios, establecimiento de permisos y relaciones entre ambos. Linux es un sistema multiusuario, por lo que necesita de una política de permisos segura y planificada para mantener el sistema seguro

1.1.1 Política de permisos en Linux

Este tipo de administración la llevaremos a cabo cuando queramos establecer políticas de seguridad en un equipo o en una LAN, o para administrar servidores del tipo NFS, FTP o WEB.

Podemos ver como se administran los permisos de usuarios con el comando *ls-l* desde nuestro intérprete de comandos.



Cada línea tiene un formato del estilo:

{T} {rwx} {rwx} {rwx} {N} {usuario} {grupo} {tamaño} {fecha de creación}{nombre}

1er campo T: Nos indica que tipo de archivo es:

- Si es un fichero normal	d Si es un directorio	p Pipe
l Enlace simbólico	c Especial de modo carácter (Dispositivo tty, impresora...)	

2º Campo {rwx}: Nos indica los permisos que tiene el propietario del archivo.

3º Campo {rwx}: Nos indica los permisos que tiene el grupo al que pertenece el archivo.

4º Campo {rwx}: Nos indica los permisos del resto de usuarios. (Otros)

5º Campo {N}: Es el número de archivos/directorios que contiene. Si es un fichero aparecerá 1, si se trata de un directorio aparecerá como mínimo 2 (los directorios '.' y '..' más los que contenga).

6º Campo {usuario}: Indica el nombre del usuario al que pertenece el archivo o directorio.

7º Campo {grupo}: Indica el nombre del grupo al que pertenece el archivo o directorio.

8º Campo {tamaño}: Indica el tamaño.

9º Campo {fecha}: Indica la fecha de creación.

10º Campo {nombre}: Indica su nombre.

1.1.2 Administración de grupos

Los grupos de permisos se agrupan de 3 en 3: rwx

(r) significa permiso de lectura	(w) permiso de escritura	(x) permisos de ejecución
---	---------------------------------	----------------------------------

Por lo tanto un fichero con esta apariencia:

rwx r-x —

significa que tendrá permisos de lectura, escritura y ejecución para el propietario, permisos de lectura y ejecución para el grupo, y ningún permiso para el resto.

PERMISOS	SIGNIFICADO
rwx rwx rwx	Permiso para todos (Muy poco seguro para archivos sin importancia)
rwx r- —	Todos los permisos para el propietario y solo de lectura para el grupo.

r-x — —	Sólo lectura y ejecución para el propietario.
---------	---

Podemos cambiar permisos con el comando **chmod**, para administrar permisos, del siguiente modo:

[chmod] [modo] [permisos] [fichero/s]

Para añadir o quitar permisos, será utilizando estos modos:

<p>a Indica que se aplicará a todos.(<u>a</u>ll) u Indica que se aplicará al usuario.(<u>u</u>ser) g Indica que se aplicará al grupo.(<u>g</u>roup) o Indica que se aplicará a otros.(<u>o</u>ther) + Indica que se añade el permiso. - Indica que se quita el permiso. r Indica permiso de lectura. w Indica permiso de escritura. x Indica permiso de ejecución.</p>

La manera de aplicar este nuevo método será:

(A quién se aplica) +/- (Qué permisos aplica)

Aplicando lo anterior, resultarían estas posibles combinaciones:

<p>a+r Permisos de lectura para todos. +r Igual que antes, si no se indica nada se supone 'a'. og-x Quita permiso de ejecución a todos menos al usuario. u+rwx Da todos los permisos al usuario. o-rwx Quita los permisos a los otros.</p>

1.1.3 Administración de usuarios

Una vez comprendido cómo cambiar los permisos a los usuarios del sistema, es necesario complementarlo con la administración de dichos usuarios.

En Linux existe un archivo editable que contiene toda la información acerca de los usuarios creados en nuestro sistema. Lógicamente, su configuración se encontrará en el directorio de configuraciones: `/etc`

El archivo en cuestión se llama `passwd` y se localiza en `/etc/passwd`.

```

root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# vi /etc/passwd

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:./:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:./:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
dbus:x:81:81:System message bus:./:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/home/oprofile:/sbi
n/nologin
abrt:x:499:498:./etc/abrt:/sbin/nologin
"/etc/passwd" 42L, 2128C

```

En este archivo, existe una línea que contiene toda la información propia por cada usuario existente en la máquina, con una sintaxis parecida a esta:

nombre: clave_encryptada:UID:GID:GECOS:directorio_inicial:intérprete

root:x:0:0:root:/root:/bin/bash

Cada campo está separado por ‘:’

El campo nombre indica el alias con el que se logea el usuario en el sistema.
La clave encryptada clave con la que se logea el usuario, aparecerá como una ‘x’.
UID es el identificador de usuario para el sistema
GID es el identificador de grupo primario al que el usuario pertenece, puesto que un usuario puede pertenecer a muchos grupos
GECOS es un campo opcional para guardar información. Normalmente contiene el nombre completo del usuario. GECOS significa General Electric Comprehensive Operating System. En este campo podemos poner lo que queramos ya que el sistema no lo usa para ninguna administración.
directorio inicial es donde empezará el usuario una vez logueado,
intérprete de comandos usará (bash, sh, tcsh...).

Para agregar usuarios, utilizamos el comando **intérprete de comandos adduser** o **useradd** desde nuestro intérprete de comandos

Para modificar usuarios, utilizamos el comando **usermod**

Para eliminar usuarios utilizamos el comando **deluser** o **userdel**

1.1.4 Administración de bitácoras

❖ En la bitácora de usuario

Usamos el comando **su** – para ingresar con permisos de Root

Con el comando **df – h** mostramos la información sobre la utilización del espacio en disco en los diferentes sistemas de archivos montados en el sistema y se imprimen en forma legible para el usuario.

Con **ls –la** listamos absolutamente todos los archivos (incluyendo directorios) del directorio donde el usuario está posicionado.

Con **cp –Rf** copiamos recursivamente todo lo que se encuentra en SRVBASIC a \$HOME

Indicamos con **pwd** el directorio donde el usuario está trabajando

Con **history** mostramos todas las líneas de comando que se han ejecutado hasta el momento

Con **id** mostramos el identificador actual y real de usuarios y grupos

Con **cd** nos cambiamos de directorio a **/usr/local/mysql/** y listamos los archivos que están dentro de mysql

Con **more INSTALL-BINARY** leemos todo lo que está en el manual de instalación para mysql

Indicamos el directorio donde nos encontramos con **pwd**

Nos cambiamos nuevamente de directorio a **/usr/local/** con **cd**

Listamos los archivos con **ls –la**

La instalación de APACHE se realiza como usuario común, a diferencia del manejador de base de datos MySQL que se realiza como Root.

Nos cambiamos con **cd** a la carpeta **SRVBASIC** para descomprimir ahora el archivo de http

Con el comando **make** realizamos una compilación de archivos para ver cuales necesitan ser recompilados y así poder instalar.

❖ En la bitácora de root

Es aquí donde realizamos la instalación y levantamiento del manejador de la base de datos MySQL

Nos cambiamos de ubicación con **cd /etc/init.d/**

Listamos archivos y preguntamos qué usuarios están en sesión y quienes no lo están

```
>> ls
```

```
>> who –r
```

Listamos los archivos existentes **ls -la**

```
../rc5.d/
```

```
vi /etc/pam.d/
```

El paquete GDM contiene el demonio de administración de pantalla de GNOME. Es útil para permitir un acceso configurable al sistema en modo gráfico.

vi gdm

vi gdm-autologin

vi gdm-fingerprint

vi gdm-password

Reiniciamos el fedora 14 y hacemos ping para ver si tenemos salida a internet

reboot 14 ifconfig

ping 192.168.111.254

Con **yum update** actualizamos nuestra lista de repositorios y los paquetes que tengamos instalados si hay alguna actualización disponible.

Con **rpm -qa | grep mysql**, **rpm -qa | grep php** y **rpm -qa | grep http** nos damos cuenta si tenemos algún tipo de instalación en mysql php y apache.

Si tenemos algo instalado lo borramos con **rpm -e**, en nuestro caso nos encontramos instalaciones en el servidor entonces borramos lo que se encuentre dentro:

rpm -e httpd-tools-2.2.16-1.fc14.i686

rpm -e httpd-2.2.16-1.fc14.i686

rpm -e python-httplib2-0.6.0-3.fc14.noarch

Volvemos a verificar **rpm -qa | grep mysql**, **rpm -qa | grep php** y **rpm -qa | grep http**

Nos cambiamos de directorio **cd /home/jossavet-pc/SRVBASIC/** para descompactar la carpeta del manejador de bd MySQL

tar -zxvf mysql-5.1.40-linux-i686-glibc23.tar.gz

Copiamos y le indicamos la ruta

cp -Rf mysql-5.1.40-linux-i686-glibc23 /usr/local/mysql

Nos cambiamos de directorio **cd /usr/local/mysql/**

Agregamos un nuevo grupo y usuario con contraseña

groupadd mysql

useradd -g mysql mysql

passwd mysql

Listamos los archivos con **ls -la**

Cambiamos de usuario y de grupo **chown -Rf mysql:mysql .**

Levantamos el manejador de base de datos MySQL **scripts/mysql_install_db --user=mysql**

1.2 Programación de la Base de Datos

Para la programación de la base de datos de nuestro sistema ocupamos el lenguaje SQL, apoyándonos con el manejador de MySQL.

El nombre de la base de datos se llama cenergia, la cual fue creada con la siguiente sintaxis:

```
CREATE DATABASE 'cenergia';
```

Y entonces creamos las tablas que necesitamos, para nuestro proyecto se programaron 11 tablas, sin embargo presentamos la programación para crear la tabla encuestador, insertar datos en la tabla, actualizar datos de la tabla, consultar y dar de baja datos de la tabla.

Creando la tabla

```
CREATE TABLE 'encuestador' {  
  
'id_encuestador'      varchar (10) NOT NULL;  
'appat'              char(20) NULL;  
'ape_mat'            char(20) NULL;  
'noms'               char(30) NULL;  
'noms_2'             char(30) NULL;  
'tipo'               char(17) NULL;  
'correo_e'           varchar(30) NULL;  
'num_tel'            int(20) NULL;  
'login'              varchar(10) NOT NULL;  
'contrasenia'        varchar(10) NOT NULL;  
PRIMARY KEY {' id_encuestador'};
```

Creando campos de la tabla

```
ALTER TABLE 'encuestador' ADD 'tipo_prestador' VARCHAR(20) NOT NULL;
```

Modificando el nombre de un campo

```
ALTER TABLE 'encuestador' CHANGE 'contrasenia' 'contraseña' VARCHAR(10)  
CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL
```

Eliminando un campo

```
ALTER TABLE 'encuestador' DROP 'noms_2'
```

Insertando datos

```
INSERT INTO 'cenergia' . 'encuestador' {  
  
'id_encuestador',
```

```

'appat',
'ape_mat',
'noms',
'tipo',
'correo_e',
'num_tel',
'login',
'contraseña',
'tipo_prestador'
VALUES { '302166239','Pozas','Montoya','Karla
Jossavet','ADMINISTRADOR','joss_kar@hotmail.com','46337955','joss','JOSS','Trabajador'};

```

Actualizando datos

```
UPDATE 'cenergia' . 'encuestador' SET 'noms'='KARLA' WHERE 'id_encuestador'='302166239'
```

Consultando todos los datos

```
SELECT *FROM 'encuestador'
```

Eliminando datos

```
DELETE FROM 'encuestador' WHERE 'id_encuestador'='302166239'
```

Si lo que queremos es eliminar la base de datos, entonces utilizaremos la sentencia:

```
DROP DATABASE 'cenergia';
```

Este es un procedimiento similar que se aplica para cada una de las tablas del sistema y de las opciones que podemos realizar dentro del sistema.

1.3 Programación del sitio Web

Se realizará una descripción breve de la programación que utilizamos para el *menú* del sistema.

Estructura básica de un documento HTML

<HTML>	Indica el inicio del documento.
<HEAD>	Inicio de la cabecera.
<TITLE>	Inicio del título del documento.
</TITLE>	Final del título del documento.
</HEAD>	Final de la cabecera del documento.
<BODY>	Inicio del cuerpo del documento.
</BODY>	Final del cuerpo del documento.
</HTML>	Final del documento.

Cabecera del documento del SisFI-UNAM Ahorro de Energía

```
<html> //inicio de nuestro documento
<?php session_start ( ); ? //se inicia sesión dependiendo del tipo de usuario que ingreso al sistema
<head> //inicio de cabecera
<title>FI- UNAM AHORRO DE ENERGIA</title> //titulo de inicio de nuestro sistema
```

Cuerpo del documento

```
<body> // Indica el inicio del contenido de nuestra página, textos, gráficos, enlaces, códigos de color e
hipervínculos
<div class="Estilo4" id="Layer1"> // formato del texto
    <div align="center"> //alineación centrada para el título de la página inicial
        <h3>FACULTAD DE INGENIERIA</h3> // título de la página inicial
        <p>FI- UNAM AHORRO DE ENERGIA </p>
    </div> //cerramos alineación
</div> //cerramos formato
```

// aquí iniciamos con la barra de menú, que nos indica las opciones iniciales que podemos realizar dependiendo del tipo de usuario de acceso

```
<td width="136" height="40"> // ancho y altura de la barra de menú
<a href="formulario_alta.php" //hipervínculo que nos permitirá realizar la acción para dar de alta
onmouseover="mm_showmenu(window.mm_menu_0216213603_0,0,41,null,'image1')"
onmouseout="mm_starttimeout();" </a></td> //acción que realizará el mouse al momento de dar click a la imagen en
formato png
<?php if ($_SESSION ['tipo']=="ADMINISTRADOR") { ? // solo sí tenemos permiso como administrador,
podemos ingresar al botón de Baja
<td width="137"><a href="formulario_baja.php" //hipervínculo que nos permitirá realizar la acción para
dar de baja
onmouseover="mm_showmenu(window.mm_menu_0216220925_0,0,39,null,'image2')"
onmouseout="mm_starttimeout();" </a></td> //acción que realizará el mouse al momento de dar click a la imagen en
formato png
<?php } ? //cierro sesión para el permiso de usuario
</body> //cierro cuerpo de la página
</html> //cierro documento
```

BIBLIOGRAFÍA

- [1] CONAE, Diagnósticos Energéticos. México: Secretaría de Energía, 1995.
- [2] CONAE, Administración y Ahorro de Energía. México: Secretaría de Energía, 1997.
- [3] Metodologías ágiles para el desarrollo de software, disponible en:
<http://www.willydev.net/descargas/masyxp.pdf>
- [4] Modelo Vista-Controlador; Definición y características, disponible en:
<http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>
- [5] VACCA, John R. Los Secretos de la Seguridad en Internet. España: Ediciones ANAYA multimedia, 1997.
- [6] Sistemas de Información, disponible en: <http://www.econlink.com.ar/sistemas-informacion/definicion>
- [7] Historia de la Informática/ Historia de las Bases de Datos, disponible en:
<http://histinf.blogs.upv.es/2011/01/04/historia-de-las-bases-de-datos/>
- [8] SILBERSCHATZ, Abraham. Sistemas Operativos. México: Pearson Educación, 1999.
- [9] TANENBAUM, Andrew S. Sistemas Operativos: Diseño e Implementación. México: Prentice Hall Hispanoamericana, 1998.
- [10] Aprendiendo a usar Linux / ¿Qué es un sistema operativo?, disponible en:
<http://aprendiendoausarlinux.wordpress.com/2012/01/15/que-es-un-sistema-operativo/>
- [11] Breve historia de los sistemas operativos, disponible en:
<http://platea.pntic.mec.es/~jdelucas/sistemasoperativos.htm>
- [12] Información sobre Linux, disponible en: <http://www.linux-es.org/>
- [13] BLANCO, Vicente J. Linux, Instalación y Uso del sistema. México: Alfaomega Grupo Editor, 1997.
- [14] ROMO, Proaño Marcelo, Informática Básica. Ecuador: Escuela Politécnica del Ejército, 2005.
- [15] Información sobre Windows, disponible en: <http://www.microsoft.com>

- [16] Información sobre los lenguajes de programación, disponible en:
http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_terminados/PolilibroFC/Unidad_III/Unidad%20III_4.htm
- [17] DATE, C.J. Introducción a los Sistemas de Bases de Datos. Única edición en español. Estados Unidos de Norteamérica: Adison – Wesley Iberoamericana, S.A., 1986. 648 p. ISBN 0-201-14440-9.
- [18] Historia de las Bases de datos, disponible en: <http://www.slideshare.net/da4equipo3/historia-de-las-bases-de-datos>
- [19] Breve historia de Ascher Opler, documento disponible en:
<http://webcache.googleusercontent.com/search?q=cache:v4swoUW68HEJ:corphist.computerhistory.org/corphist/documents/doc-437a4a92bbf21.doc+&cd=13&hl=es&ct=clnk&gl=mx>
- [20] TSAI, Alice Y.H. Sistemas de Bases de Datos: Administración y Uso. México: Prentice Hall Hispanoamericana, 1990.
- [21] SILBERSCHATZ, Abraham; Korth, Henry F.; Sudarshan, S. Fundamentos de Bases de Datos. Quinta Edición, España: McGraw-Hill, 2006.
- [22] Información sobre el Lenguaje de consulta estructurado SQL, disponible en:
http://www.slideshare.net/alexandrita_da85/lenguaje-sql
- [23] BARCELÓ, José M. et. al. Redes de Computadores. 1a. ed. España: Eureka Media, 2004. 351 p.
- [24] GONZÁLEZ SAINZ, Néstor. Comunicaciones y Redes de Procesamiento de Datos. 1a. ed. México: McGRAW – HILL, 1987. 396 p. ISBN 968-422-147-9
- [25] Diferencias entre MySQL y Postgresql, disponible en:
<http://www.bisente.com/documentos/mysql-postgres.html>
- [26] TANENBAUM, Andrew S. Redes de Computadoras. 4a. ed. México: Pearson Educación, 2003. ISBN 970-26-0162-2.
- [27] Definición de un Diagnóstico energético, disponible en:
<http://www.cubasolar.cu/biblioteca/energia/Energia25/HTML/articulo09.htm>
- [28] Información sobre la Verificación, Prueba y Mantenimiento de los Sistemas, disponible en:
<http://aceproject.org/main/espanol/et/ete05.htm>