



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA DE ADMINISTRACIÓN DE PROYECTOS,
ACTIVIDADES Y RECURSOS HUMANOS
ASIGNADOS**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTAN:

**JAIME AVENDAÑO VARGAS
SANDRA LUZ PAREDES ASBELL
JOSE LUIS GARCIA ROJAS
PEDRO BLANCAS GILES
OLIVA SANCHEZ VEGA**

DIRECTORA DE TESIS:

ING. LUCILA P. ARELLANO MENDOZA



MÉXICO, D.F., ABRIL, 2004



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A Dios por darme la oportunidad de vivir.

A mis padres por su apoyo incondicional en toda mi vida.

A mis hermanos que me acompañaron en todos mis estudios.

A mi esposa y compañera que siempre ha estado a mi lado.

Jaime Avendaño Vargas

Agradecimientos

A ti Señor porque en los momentos de debilidad y cansancio tu presencia me fortalece y me anima a continuar.

A ti papá (†) porque la fuerza de tu carácter me ayuda a alcanzar las metas propuestas día con día. Te quiero y te extraño.

Mamá gracias por tu cariño y tu apoyo en todo momento. Te quiero mucho.

A mis hermanos y sobrinos por su presencia y formar la familia que somos.

A ti Jaime por estar siempre conmigo sosteniéndome en cada caída y por ser mi vida misma. Te amo.

Padre Ricardo Lemus (†) gracias por ser mi guía espiritual y un gran amigo.

A Miss Cristy por ser un gran ejemplo y alentarme siempre a dar este paso.

A mis maestros de la Facultad de Ingeniería, por compartir sus conocimientos y sabiduría. Tio Dick, gracias por tus consejos.

A la Universidad Nacional Autónoma de México, por todo lo que como Institución me ha brindado, por ser el alma mater y el hogar de tantas personas que como yo, soñamos un día estar titulados y poder decir
“soy ... Orgullosamente UNAM”

Gracias Ing. Lucila por ayudarnos a dar este gran paso que nos costaba tanto darlo, gracias por su labor y entrega a la Facultad y a los alumnos.

Sandra Luz Paredes Asbell

Agradecimientos

A mi padre José Luis, gracias por haberme enseñado el valor de la responsabilidad y el esfuerzo, gracias a ello alcanzo una meta muy importante en mi vida.

A mi madre Guadalupe, porque siempre has sido ejemplo de amor y valor, tu ejemplo me ha motivado siempre a no rendirme y seguir siempre para alcanzar mis objetivos.

A mis hermanos porque cada uno de ustedes ha contribuido para llegar hasta aquí. Martín gracias por tus regaños y apoyo. Claudia siempre has estado conmigo cuando lo he necesitado. Pita gracias por tu comprensión, ternura y el cariño, a todos los quiero y respeto mucho.

Para mi esposa Claudia, porque siempre has tenido la palabra que me reconforta y anima a seguir aún cuando me equivoco. Por tu paciencia y cariño, tu fortaleza y amor, gracias, te quiero.

Juan Enrique, Albertina, Delia, Mayte, Rocío, Diego, Atziri, Sally, Gracias por confiar en mí.

A toda mi familia, sería interminable mencionarlos a todos, pero todos están presentes en mi mente y corazón.

Margarita, Rogelio, Julio haber convivido con ustedes me motivó siempre para dar mi mejor esfuerzo y a confiar más en mí y en lo que puedo lograr.

Para todas aquellas personas que me alentaron y apoyaron, en la escuela, en el trabajo, gracias a todos.

José Luis García Rojas

SISTEMA DE ADMINISTRACIÓN DE PROYECTOS, ACTIVIDADES Y RECURSOS HUMANOS ASIGNADOS

Temario	Página
Capítulo 1. Introducción	
1.1 Antecedentes	2
1.2 Definición del Problema	3
1.2.1 Planteamiento del Problema	4
1.2.2 Entorno actual y tendencias	4
1.2.3 Sistemas en sitios Web	5
1.3 Objetivos	6
Capítulo 2. Fundamentos Teóricos y Tecnológicos	
2.1 Ingeniería de Software	8
2.1.1 Metodologías de desarrollo de sistemas	8
2.2 Bases de Datos	16
2.2.1 Conceptos	16
2.2.2 Modelo Entidad-Relación	20
2.2.3 Modelo Relacional	23
2.2.4 Modelo Cliente-Servidor	30
2.2.5 Normalización	38
2.3 Internet	42
2.3.1 Sistemas operativos de servidores Web	43
2.3.2 Servidores Web	46
2.3.3 Arquitecturas de Hardware	55
2.3.4 Manejadores de Bases de Datos	63
2.3.5 Herramientas de Programación	68
Capítulo 3. Análisis del Sistema	
3.1 Requerimientos	75
3.2 Propuesta de Solución	76
3.2.1 Justificación	77
3.3 Modelado de Datos	77
3.3.1 Diagrama Entidad-Relación	77
3.3.2 Diagrama de Flujo de Datos	80
3.3.2.1 Diagrama de Procesos	80
3.4 Análisis de la Interfaz Gráfica	86
Capítulo 4. Diseño del Sistema	
4.1 Diseño de la Base de Datos	88
4.1.1 Modelo Relacional	89
4.1.2 Normalización	90
4.1.3 Diccionario de Datos	92
4.2 Selección de Elementos y Herramientas de desarrollo	98
4.2.1 Manejador de Bases de Datos	99
4.2.2 Servidor de Web	102
4.2.3 Herramienta de Programación	105
4.2.4 Hardware y Arquitectura de desarrollo	108
Capítulo 5. Implementación y Pruebas	
5.1 Implementación del Sistema	110
5.1.1 Desarrollo de la Base de Datos	110
5.1.2 Implementación de la Interfaz Gráfica	111
5.2 Pruebas realizadas al Sistema	114
Conclusiones	118

CAPÍTULO 1. INTRODUCCIÓN

1.1 Antecedentes.

Uno de los problemas que se presentan actualmente en las empresas dedicadas al desarrollo de software es la asignación de personal a puestos y tareas específicas sin tener un control real de sus actividades. Esto ha llevado a tener personal en actividades sobre las que no se tiene control de tiempos ni de costos los cuales se ven reflejados en pérdidas para la empresa, tanto económicas como en recursos humanos.

Dentro de las tareas más importantes en una empresa de desarrollo de software, es colocar el personal adecuado en el puesto adecuado, tarea que en muchos de los casos nunca ocurre, lo que conlleva a tener personas que rinden menos de lo que su capacidad puede dar, obteniendo poco rendimiento en su desempeño.

Es necesario buscar formas de asignar a cada persona una tarea en la cual no sólo se sienta identificado y pueda rendir al cien por ciento, sino que además, proporcione a la empresa un beneficio tanto en costo como en la mejora de sus productos.

La asignación de tareas en el desarrollo de proyectos es asignada a través de los roles de cada persona, de esta forma tenemos personas encargadas de la venta y contacto con los clientes, personas líderes de los proyectos, personas dedicadas al análisis del proyecto o sistema, personas dedicadas al desarrollo o programación de las interfaces y procesos de los sistemas, así como personal asignado a la entrega y mantenimiento de los sistemas o desarrollos de software.

Dependiendo de las tareas que se tengan ya establecidas al inicio de un proyecto se les asignan de acuerdo a los roles que a cada empleado le correspondan, de esta forma tendremos al inicio de un proyecto tareas asignadas solo al personal encargado de realizar la venta y contacto con el cliente, teniendo como primer tarea el obtener el proyecto y estimar los tiempos y recursos que deben asignarse al mismo.

Una vez autorizado el proyecto las tareas son asignadas de acuerdo al rol de cada persona, en donde se especifican los tiempos en los cuales deben ser iniciadas y terminadas cada una de ellas.

Actualmente los proyectos son llevados a cabo principalmente poniendo metas en tiempos de entrega; estos tiempos son repartidos en diferentes actividades principalmente en las cinco fases que constituyen el desarrollo de software los cuales son: el análisis, diseño, modelado, programación y puesta en producción de los sistemas. Para cada una de estas tareas un líder o responsable de proyecto, se dedica una gran parte de su tiempo a estimar y planear las diferentes tareas por realizar asignando los recursos humanos y el tiempo necesarios para cada una de ellas, esta es una tarea de uso común en todo desarrollo, sin embargo no existe una forma directa de medir en forma precisa si el personal asignado a cada tarea está rindiendo y haciendo las tareas que le han sido asignadas; la forma actual para hacer estas medición es

verbal y a través de preguntas directas a las personas, las cuales en la mayor cantidad de ocasiones, estima su avance a su conveniencia, siendo en muchos casos poco real.

Esto se refiere sólo a la parte que corresponde a su avance, sin embargo, si quisiéramos conocer el tiempo real que ha dedicado a cada tarea, no tenemos los elementos necesarios para saberlo, a menos que tengamos un contacto directo con las personas y estemos a su lado para saber si realmente ha dedicado el tiempo que dice haber asignado al mismo. Para estimar el tiempo enfocado a cada tarea es necesario confiar en lo que la persona a cargo de dicha tarea nos diga, siendo normalmente fuera de realidad y con ventajas a sus intereses.

Es por ello que se hace necesario conocer de forma más precisa, el tiempo que una persona está dedicando a sus tareas asignadas, de forma al menos diaria, lo cual nos permitirá tomar decisiones en momentos precisos adelantando cualquier demora en el desarrollo de software.

También nos permitirá llevar un control y una medición más precisa de los avances que se tienen en las tareas, lo que nos proporcionará en desarrollos futuros tener los elementos necesarios para estimar tiempos y recursos que se deban asignar tomando como base las medidas ya realizadas en proyectos de software realizados.

Un beneficio más que nos proporciona el realizar estas mediciones, es conocer en que actividades una persona se desenvuelve de una mejor forma, obteniendo mayores satisfacciones en su desempeño logrando con esto, realizar trabajos de mayor calidad, lo que beneficia en mejores productos finales.

Durante la planeación de un proyecto asignamos personas a cada actividad, en un principio dependiendo sólo del personal con el que se cuenta y sin considerar alguna especialidad. El controlar los tiempos asignados por cada persona a una actividad en particular, nos va a servir también para especializar a cada participante del desarrollo, así por ejemplo, si tenemos que realizar tres tareas en un proyecto tales como análisis, diseño y programación, a partir de los tiempos que cada persona le asigna a los mismos, podremos saber quienes se han dedicado más tiempo ya sea al análisis, al diseño o a la programación y con ello tendremos los elementos necesarios para especializar a cada uno en las tareas que más le agraden y en las que se desempeña con mayor calidad.

1.2 Definición del Problema.

Debido a la necesidad de llevar un control más preciso de los tiempos asignados por cada persona a sus tareas en el desarrollo de software, así como de su desempeño en las mismas, se vuelve de suma importancia la creación de mecanismos para obtener resultados que nos permitan tener acceso a la información necesaria para colocar a las personas en lugares en donde se desenvuelva de una mejor forma, además de tener la información necesaria para tomar decisiones rápidas y precisas sobre el camino que se está tomando en el desarrollo del software.

1.2.1 Planteamiento del Problema.

En este momento, muchas de las personas con preparación y capacidad para atender tareas específicas se encuentran atendiendo actividades que pueden ser realizadas por personas con menor preparación, provocando que tareas con bajos costos le resulten a la empresa costos muy elevados al tener personal especializado realizando tareas que podrían ser realizadas por personas con otra preparación.

Otro de los problemas que se presentan actualmente, es que la dirección de la empresa no cuenta con herramientas para valorar el desempeño de cada uno de sus empleados, tanto para las actividades que realizan como los tiempos que ocupan en ellas.

La falta de organización en la elaboración de tareas provoca descuidos en el proceso de tareas que le competen a la persona y para las que el puesto está diseñado.

No hay forma de estimar tiempos en proyectos o tareas lo que provoca incertidumbre en la entrega de resultados, o bien, en la estimación de tiempos para nuevos proyectos.

Esta problemática genera una mala utilización de los recursos tanto humanos como materiales, ya que eleva los costos en sueldos al tener una persona con un sueldo desempeñando actividades que pueden ser desarrolladas por una persona contratada con un sueldo diferente o menor.

Es necesario conocer si la asignación real del personal al proyecto es la adecuada para evitar que se tengan tiempos muertos en las actividades o en caso contrario, un trabajo excesivo.

Un problema más que se presenta en la asignación de un proyecto, es no poder consultar desde el punto donde se encuentren las personas encargadas de la toma de decisiones, la información suficiente para poder llevar el manejo del desarrollo de los proyectos, costos, personal asignado y recursos disponibles con los que cuenta la empresa.

1.2.2 Entorno actual y tendencias.

Actualmente existen pocas herramientas con las cuales se tenga el control real de un proyecto contabilizando las actividades en las que se desempeña cada persona durante un día de trabajo. Una de las herramientas de mayor uso se enfoca al control de proyectos pero de forma general abarcando costos y actividades de cada persona pero sin tener la certeza que el personal está realmente realizando dichas actividades.

La tendencia es automatizar la contabilidad de tiempo asignado a las tareas día por día obteniendo con ello, un valor más real de las actividades que desarrolla una persona

durante su día de trabajo sin llegar a ser específico y tedioso, tendiendo a una estimación general por día.

En la actualidad y debido a que los desarrollos de sistemas se realizan en muchos casos en las instalaciones de los mismos clientes o bien en fábricas de software ubicadas en lugares diferentes de donde se encuentran los directivos de la empresa, se vuelve importante tener acceso a la información y al registro de actividades desde cualquier punto en donde se encuentre el personal asignado a los proyectos, esto ha llevado al desarrollo de software que se presenta con una interfaz montadas sobre servidores Web, presentando así la ventaja de que en cualquier punto en donde se encuentre una conexión a Internet se puedan registrar día a día las actividades.

Con este tipo de interfaces es posible recabar la información de todo el personal involucrado en un desarrollo de software el cual registra sus actividades, mismas que pueden ser revisadas por parte de los líderes o responsables de los proyectos en cualquier parte que se encuentren y que además cuenten con una conexión a Internet.

Hoy en día la evolución en las comunicaciones vía Internet están tomando un mayor auge debido a las ventajas que presenta el obtener información en cualquier sitio en donde se cuente con una conexión y un browser, de esta forma se tienen diferentes sistemas que permiten a una empresa extender sus alcances y el control de actividades, ventas y administración a cualquier parte del mundo.

1.2.3 Sistemas en sitios Web.

El mercado actual de software dedicado a llevar el control de actividades de cada persona, aún es reducido, enfocándose principalmente a la estimación en tiempos y personas relacionadas a cada tarea de un proyecto. En cuanto a la incursión en un ambiente Web es aún más reducida existiendo en el mercado muy pocas herramientas con propósitos de llevar el control de las actividades individuales de personas, como ejemplo nombramos a cuatro, las cuales son enfocadas principalmente al control del proyecto y no de los tiempos asignados a cada tarea de forma real y diaria:

- Time Panic.- Este software nos permite llevar un control diario de nuestras actividades tal como lo requerimos, sin embargo trabaja de forma independiente para cada usuario y no puede interactuar con más personas. Presenta también la desventaja que no esta diseñado para una plataforma Web¹.
- Alexys Team.- Software desarrollado para el control de proyectos con un enfoque multiusuario, en donde incluso utiliza una interfase con Outlook para

¹ <http://www.timepanic.com>

realizar reuniones. Incluso es un software capaz de llevar el control de nuevas requisiciones de trabajo o modificaciones a alguna tarea ya asignada. También se tiene en un formato de Web para salida de reportes; sin embargo el objetivo que nos hemos planteado, no sólo es el de llevar el control de un proyecto y sus actividades si no el de controlar diariamente a cada persona, obteniendo la información de sus actividades. Este software no incluye interacción con la información vía Web.¹

- TimeSheet Express.- Es una de las herramientas más completas en ambiente Web. Se pueden registrar cada una de las actividades que se desarrollan en los proyectos de software. En este software se incluyen todos los proyectos así como sus tareas brindando la posibilidad de asignarlas a cualquier persona participante del mismo. A pesar de ser una de las herramientas mas completas, aun le hace falta el registro de asistencia de las personas que participan en el proyecto, la cual nos permite de forma exacta conocer el inicio de actividades de cada participante.²
- Microsoft Project.- Herramienta de las más utilizadas para la planeación de proyectos. Su principal función es la de crear proyectos y tareas asignando a las mismas recursos humanos y estimando tiempos para cada una de las tareas. Actualmente ya cuenta con consultas vía Internet sin embargo este producto no cuenta con el registro de actividades par parte de cada uno de los involucrados en el proyecto.³

1.3 Objetivos.

Crear una herramienta que nos permita llevar el control de las actividades del personal que labora en una empresa de desarrollo de software. Este control está enfocado a recabar la información diaria de las actividades asignadas a cada proyecto.

El análisis de las actividades diarias, nos permitirá conocer con un grado mínimo de incertidumbre la óptima utilización de los recursos humanos en cada actividad, así como de sus avances, permitiéndonos con esto tomar decisiones al colocar personas de apoyo en caso de tareas ajenas a las destinadas para cada persona o bien hacer una reestructuración de personas para obtener soluciones a las diferentes tareas que se desarrollan en la creación de software. También nos permitirá hacer ahorros al mantener al personal en actividades en donde mejor se desempeñen.

¹ <http://www.alexcorp.com/>

² <http://www.timesheetelite.com>

³ <http://office.microsoft.com/home/default.aspx>

La herramienta pretende estar en una plataforma Web con la finalidad de obtener información de actividades desde cualquier punto en donde se encuentre una conexión a Internet. Esto nos presenta la ventaja de que todo el personal que tenga una conexión pueda brindar la información necesaria de sus avances en donde quiera que se encuentre pues sólo necesitará entrar a través de un browser e Internet al sitio en donde podrá consultar o modificar todas las actividades que tiene asignadas, colocando a cada una un porcentaje aproximado del tiempo dedicado durante el día a cada una de ellas.

También teniendo el desarrollo en una plataforma Web, permitirá dar a los directivos y líderes o responsables de los proyectos, una herramienta en donde pueden obtener de forma rápida y en línea el estado actual de los proyectos y los tiempos asignados hasta ese momento a cada una de las actividades proyectadas en el desarrollo de software.

Otro propósito a cubrir es el poder consultar cuántas personas y quiénes son los asignados a cada actividad dentro del proyecto de desarrollo de software.

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS TECNOLÓGICOS

2.1 Ingeniería del Software.

Para la IEEE "software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según el mismo autor, "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software (SE del inglés *Software Engineering*) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", que en palabras más llanas, se considera que "la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones económicamente efectivas a los problemas de desarrollo de software".

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad". El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo".

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

2.1.1 Metodologías de desarrollo de sistemas.

Existen varios métodos tanto formales clásicos como orientados a objetos los cuales se presentan de forma resumida a continuación:

Métodos Orientados a Objetos.

En cuanto a las metodologías OO, diremos que actualmente hay un gran número de métodos orientado a objetos. Muchos de los métodos pueden ser clasificados como orientados a objetos porque soportan de manera central los conceptos de la orientación a objetos. Algunas de las metodologías más conocidas y estudiadas hasta antes del UML según Jacobson 1992 son:

- Object-Oriented Design (OOD), Booch.
- Object Modeling Technique (OMT), Rumbaugh.

- Object Oriented Analysis (OOA), Coad/Yourdon.
- Hierarchical Object Oriented Design (HOOD), ESA.
- Object Oriented Structured Design (OOSD), Wasserman.
- Object Oriented Systems Analysis (OOSA), Shaler y Mellor.
- Responsibility Driven Design (RDD), Wirfs-Brock, entre otros.

Actualmente las metodologías más importantes de análisis y diseño de sistemas han concluido en lo que es el UML, bajo el respaldo del Object Management Group.

Metodología de Booch.

La metodología de Booch usa los siguientes tipos de diagramas para describir las decisiones de análisis y diseño, tácticas y estratégicas, que deben ser hechas en la creación de un sistema orientado por objetos utilizando:

- Diagrama de Clases.
- Diagrama de Categorías.
- Diagramas de transición de estados.
- Diagramas de Objetos.
- Diagramas de Tiempo.
- Diagramas de módulos.
- Diagramas de procesos.

Object Modeling Technique, por James Rumbaugh et al.

OMT hace un cubrimiento de las etapas de análisis, diseño e implementación definidas por la OMG, dejando sin cubrir el modelamiento estratégico. Modelo de Objetos se define como un diagrama de objetos más un diccionario de datos. El diagrama de objetos muestra las clases y sus relaciones (generalización, agregación, asociación, instanciación). El diccionario de datos es el detalle de las clases en el diagrama de objetos incluyendo:

- Modelo dinámico.
- Modelo funcional.

Métodos Clásicos Estructurados.

Un modelo de proceso de software es una representación abstracta de un proceso. Presenta una descripción de un proceso desde una perspectiva particular.

El modelo de cascada.

También conocido como ciclo de vida clásico, es una extensión al modelo de etapas pero con un reconocimiento de ciclos de retroalimentación entre etapas sucesivas. Tal y como se muestra en la figura 2.1.

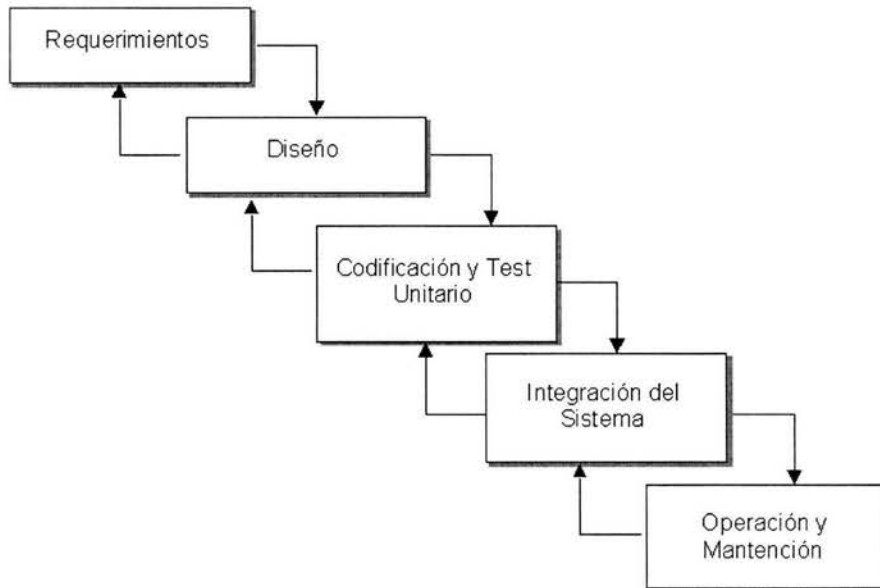


Figura 2.1 Modelo en Cascada

Las fases del modelo de cascada.

- El análisis y definición de los requerimientos.
- El diseño del sistema y software.
- Implementación y pruebas de unidad.
- Integración y pruebas del sistema.
- Operación y mantenimiento.

La desventaja del modelo de cascada es la dificultad de permitir cambio después de que el proceso haya iniciado.

Los problemas con el modelo de cascada

- División inflexible del proyecto en fases distintas
- Esto hace difícil responder a los requerimientos cambiantes del cliente
- Por eso, este modelo solo es apropiado cuando los requerimientos son bien comprendidos.

Desarrollo evolutivo.

El desarrollo evolutivo es una metodología de desarrollo de software muy relacionada con desarrollo por prototipos. El énfasis está puesto sobre la importancia de obtener un sistema de producción flexible y expandible. Así, si los requerimientos cambian durante el desarrollo del sistema, entonces con un mínimo de esfuerzo y tiempo se puede desarrollar un sistema de trabajo flexible. Dicho método se representa gráficamente en la figura 2.2.

- Desarrollo exploratorio.
- El objetivo es trabajar con el cliente y evolucionar un sistema final desde una especificación inicial a grandes rasgos. Se debe empezar con los especificaciones bien comprendidas.
- Haciendo prototipos desechables.
- El objetivo es entender los requerimientos del sistema.

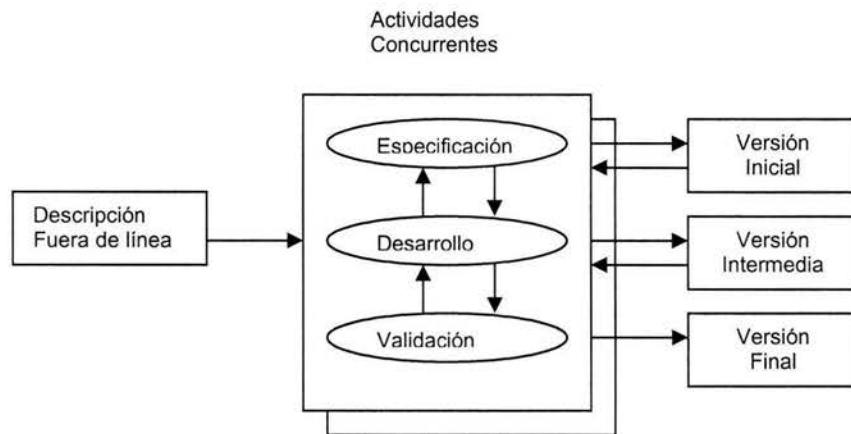


Figura 2.2 Desarrollo Evolutivo

- Actividades concurrentes.
- Descripción a grandes rasgos.
- Especificación.
- Desarrollo.
- Validación.
- Versión inicial.
- Versiones intermedias.
- Versión final.

Problemas.

- Una falta de visibilidad del proceso.

- Los sistemas a menudo son mal estructurados.
- Habilidades especiales (Ej., en lenguajes para hacer rápidamente los prototipos).

Aplicabilidad.

- Para sistemas interactivos pequeños o de tamaño medio.
- Para partes de sistemas grandes (Ej., la interfaz de usuario).
- Para sistemas con vidas cortas.

Desarrollo de sistemas formales.

Basado en la transformación de una especificación matemática a través de diferentes representaciones hasta un programa ejecutable, las transformaciones preservan la correctividad, por eso es sencillo mostrar que el programa se ajuste a su especificación, es encarnado en el enfoque "Cuarto Limpio" (Cleanroom) al desarrollo de software, El proceso de este desarrollo se puede observar en la figura 2.3.

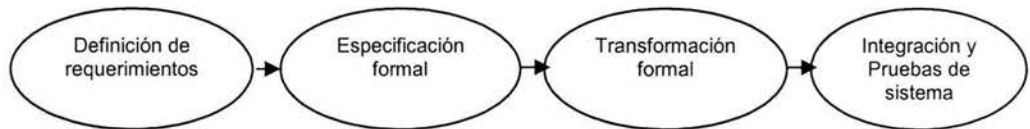


Figura 2.3 Proceso de Desarrollo de Sistemas Formales

- Definición de requerimientos.
- Especificación formal.
- Transformación formal.
- Integración y pruebas del sistema.

Transformaciones formales.

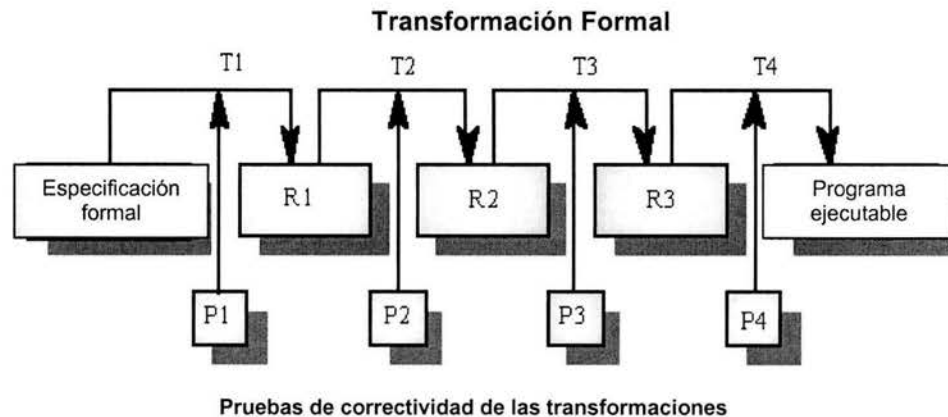


Figura 2.4 Transformaciones formales

- Transformaciones formales.
- Pruebas de correctividad de las transformaciones.
- Especificación formal.
- Programa ejecutable.

Problemas.

- La necesidad de habilidades especiales y entrenamiento para aplicar la técnica.
- Es difícil especificar formalmente algunos aspectos del sistema como la interfaz de usuario.

Aplicabilidad.

- Sistemas críticos, especialmente esos donde se necesita demostrar la seguridad antes de que el sistema sea lanzado a producción.

Desarrollo orientado a rehusos.

Basado en el rehusos sistemático donde los sistemas son integrados desde componentes existentes o sistemas COTS (Commercial off-the-shelf) (software no customizado para un mercado general).

Las fases del proceso.

- Análisis de los componentes.
- Modificación de los requerimientos.
- Diseño del sistema.
- Desarrollo e integración.

Las fases de este proceso se observan gráficamente en la figura 2.5. Este enfoque está volviéndose más importante, pero todavía hay experiencia limitada con él.

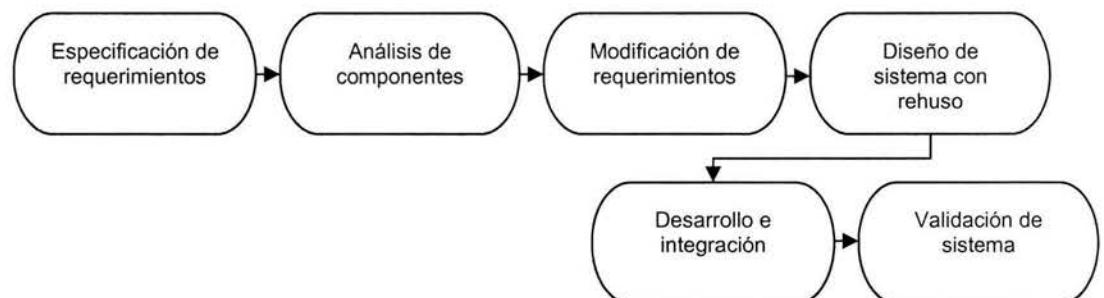


Figura 2.5 Fases del proceso de desarrollo orientado a rehusos

- Especificación de requerimientos.
- Análisis de componentes.
- Modificación de requerimientos.
- Diseño del sistema.

- Desarrollo e integración.
- Validación de sistema.

Iteración del proceso.

Los requerimientos del sistema siempre evolucionan durante el curso del proyecto, por eso la iteración del proceso donde las fases iniciales son vueltas a trabajar es siempre parte del proceso para sistemas grandes.

La iteración puede ser aplicada a cualquier modelo genérico de proceso. Dos enfoques relacionados:

- Desarrollo incremental.
- Desarrollo en espiral.

Desarrollo incremental.

En vez de entregar el sistema como una única entrega, el desarrollo y la entrega son descompuestos en incrementos, con cada incremento se entrega una parte de la funcionalidad requerida, como se puede observar en la figura 2.6.

A los requerimientos de usuario se les da prioridad y los requerimientos de alta prioridad son incluidos en los primeros incrementos, una vez que el desarrollo ha empezado, los requerimientos son congelados aunque los requerimientos de incrementos futuros pueden continuar evolucionando.

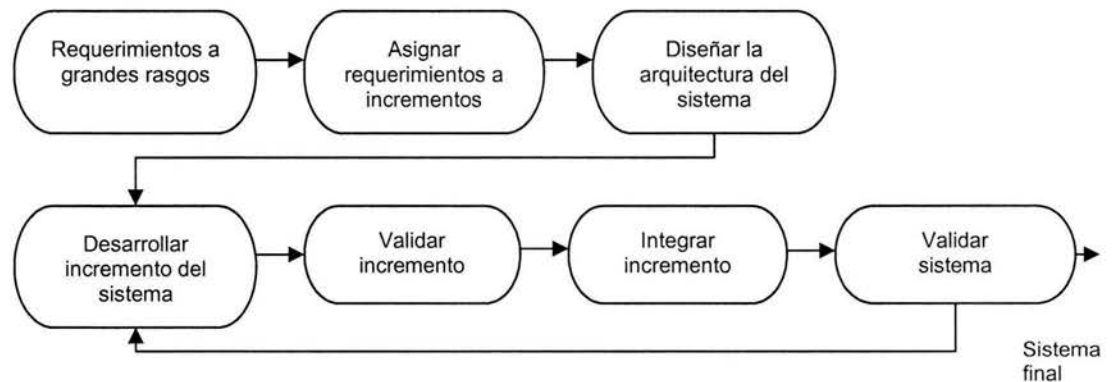


Figura 2.6 Fase del Desarrollo incremental

- Definir requerimientos a grandes rasgos.
- Asignar requerimientos a incrementos.
- Diseñar la arquitectura del sistema.
- Desarrollar incremento del sistema.
- Validar incremento.
- Integrar incremento.
- Validar sistema.
- Sistema incompleto.

- Sistema final.

Ventajas de desarrollo incremental

- Cosas que tienen valor para el cliente pueden ser entregadas con cada incremento para que la funcionalidad del sistema esté disponible más pronto
- Incrementos iniciales funcionan como un prototipo para ayudar a obtener los requerimientos de incrementos futuros
- Más bajo riesgo de fracaso del sistema entero
- Los servicios del sistema con la más alta prioridad tienden a recibir más pruebas

Programación extrema

- Enfoque nuevo al desarrollo basado en el desarrollo y la entrega de incrementos de funcionalidad muy pequeños.
- Depende en la mejora constante de código, participación de los usuarios en el equipo de desarrollo y programación en parejas.

Desarrollo en espiral.

El proceso es representado como una espiral en vez de una secuencia de actividades con vueltas hacia atrás, cada vuelta en la espiral represente una fase en el proceso, no hay fases fijas como especificación o diseño, las vueltas en la espiral son escogidas dependiendo en lo que sea requerido. Observe el proceso gráficamente en la figura 2.7.

Los riesgos son valorados explícitamente y resueltos durante todo el proceso

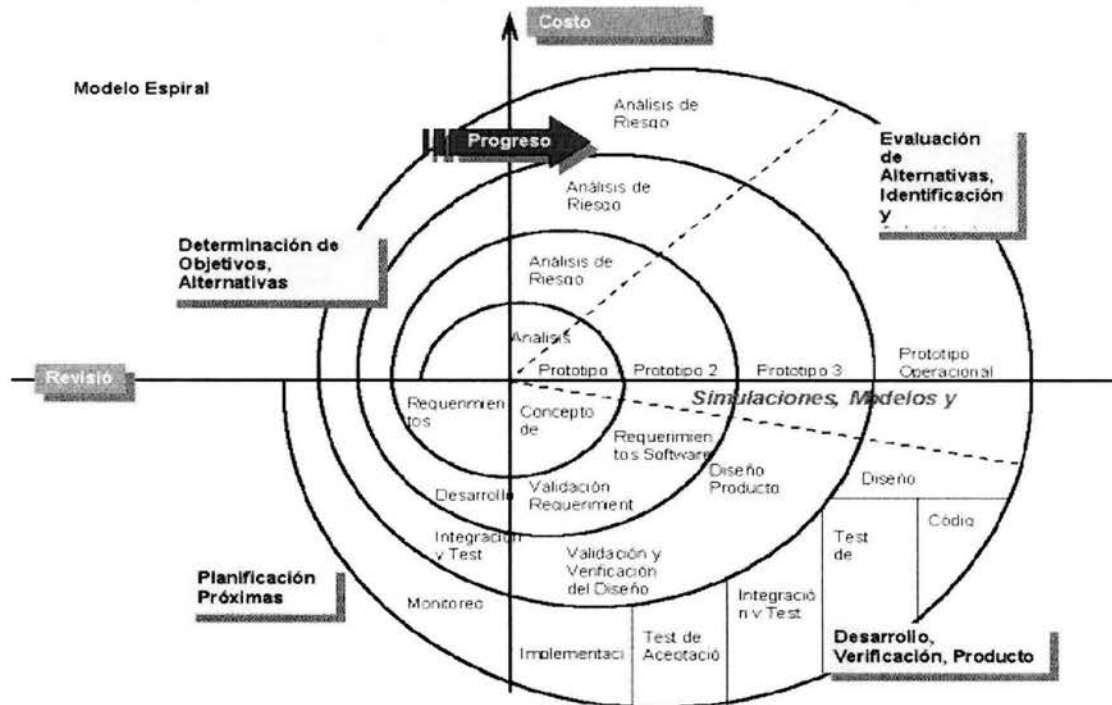


Figura 2.7 Proceso del modelo en espiral

Las fases del proceso en espiral.

- Determinar objetivos, alternativas y restricciones.
- Evaluar alternativas identificadas, resolver riesgos.
- Desarrollar, verificar producto del próximo nivel.
- Planificar próxima fase.

Los sectores del modelo en espiral.

- Determinar los objetivos.
- Los objetivos específicos para cada fase son identificados.
- Valorar y reducir los riesgos.
- Los riesgos son valorados y actividades son puestas en vigor para reducir los riesgos claves.
- Desarrollar y validar.
- Un modelo de desarrollo del sistema es escogido que puede ser cualquiera de los modelos genéricos.
- Planificar.
- El proyecto es repasado y la próxima fase de la espiral es planificada.

2.2 Bases de Datos.

Un sistema administrador de bases de datos (DBMS database management system) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos. La colección de datos se denomina base de datos (BD). El objetivo de un DBMS es controlar el acceso concurrente, evitar redundancia, que se cumplan las restricciones y reglas de integridad, usar elementos que aceleren el acceso físico a los datos, distribuir los bloques del modo más adecuado para el crecimiento y uso de los datos, controlar el acceso y privilegios de los usuarios, recuperar ante fallos entre otras cosas.

2.2.1 Conceptos.

Base de datos es un conjunto exhaustivo no redundante de datos estructurados organizados en máquina accesibles en tiempo real y compatible con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de una organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas. Por este motivo es importante conocer la forma en que están estructuradas las bases de datos y su manejo.

Las bases de datos manejan conceptos básicos para su tratamiento de información, como son:

- Registro: Es una colección de campos (atributos). Un registro, es el conjunto de información referida a una misma persona u objeto. Un registro vendría a ser algo así como una ficha.
- Campo: Unidad básica de una base de datos.
- Tabla: Conjunto de registros contenidos en campos.

Una base de datos esta conformada de diferentes componentes, estos son:

- Datos. Los datos son la base de datos propiamente dicha.
- Hardware. El hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.
- Software. Está constituido por un conjunto de programas que se conoce como sistema manejador de base de datos. Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.
- Usuarios. Existen tres clases de usuarios relacionados con una base de datos:
 1. El programador de aplicaciones, quien crea programas de aplicación que utilizan la base de datos.
 2. El usuario final, quien accesa la base de datos por medio de un lenguaje de consulta o de programas de aplicación.
 3. El administrador de la Base de Datos (DBA: Data Base Administrator), quien se encarga del control general del sistema de base de datos.

Modelos de datos

Tenemos dos modelos de datos:

Lógicos basados en registros

Existen fundamentalmente tres alternativas disponibles de este modelo para diseñar las bases de datos: el modelo jerárquico, el modelo de red y el modelo relacional.

a) El modelo jerárquico

La forma de esquematizar la información se realiza a través de representaciones jerárquicas o relaciones de padre/hijo, de manera similar a la estructura de un árbol. Así, el modelo jerárquico puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.

b) El modelo de red

El modelo de red evita redundancia en la información, a través de la incorporación de un tipo de registro denominado conector. La dificultad surge al manejar las conexiones o ligas entre los registros y sus correspondientes registros conectores.

c) El modelo relacional

Se está empleando con más frecuencia en la práctica, debido el rápido entendimiento por parte de los usuarios que no tienen conocimientos profundos sobre sistemas de bases de datos y a las ventajas que ofrece sobre los dos modelos anteriores.

En este modelo toda la información se representa a través de arreglos bidimensionales o tablas. Las operaciones básicas son:

- Seleccionar renglones de alguna tabla (SELECT)
- Seleccionar columnas de alguna tabla (PROJECT)
- Unir o juntar información de varias tablas (JOIN)

Es importante mencionar que la mayoría de los paquetes que manejan bases de datos disponibles en el mercado poseen las instrucciones SELECT, PROJECT Y JOIN con diferentes nombres y modalidades

Lógicos basados en objetos

a) El modelo entidad-relación (E-R)

Se enfoca primordialmente a los niveles conceptual y de visión. Una de las características de este modelo es que permite representar con claridad las limitantes de los datos. El modelo Entidad-Relación es en esencia una herramienta para representar el mundo real por medio de simbologías y expresiones determinadas.

Una entidad es un objeto que se distingue de otros por medio de un conjunto específico de atributos.

Una relación es una asociación entre varias entidades.

Además de las entidades y relaciones, este modelo representa ciertas restricciones a las que deben ajustarse los contenidos de una base de datos.

Una restricción importante es la cardinalidad de asignación, que expresa el número de entidades a las que puede asociarse otra entidad mediante un conjunto de relación.

La estructura lógica global de una base de datos puede expresarse gráficamente por medio de un diagrama E-R, que consta de:

- Rectángulos - representan conjuntos de entidades
- Elipses - representan atributos
- Rombos - representan relaciones entre conjuntos de entidades
- Líneas - conectan atributos a conjuntos de entidades y conjuntos de entidades a relaciones

b) El modelo Orientado a objetos

Al igual que el modelo E-R se basa en una colección de objetos.

Un objeto contiene valores almacenados en instancias dentro del objeto. Estos valores son objetos por sí mismos, esto es, los objetos contienen objetos a un nivel de anidamiento de profundidad arbitraria.

Un objeto también contiene partes de código que operan sobre el objeto. Estas partes se llaman métodos. Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan en clases. Una clase puede verse como definición de tipo para objetos.

En este modelo hay dos niveles de abstracción de datos, una que es visible externamente, que ocurre en la interfase de llamada de los métodos de un objeto y otro nivel que ocurre en la parte interna del objeto en la instancia y el código del método.

La interfaz externa del objeto permanece sin cambios, la diferencia de las entidades en el modelo E-R, cada objeto tiene su propia identidad única independiente de los valores que contiene. Así, dos objetos que contienen los mismos valores son, sin embargo, distintos.

La distinción entre objetos individuales se mantiene en el nivel físico por medio de identificadores de objeto.

Sistema Administrador de Bases de Datos (DMBS)

Un sistema administrador de base de datos es una colección de programas que permite a los usuarios crear y mantener una base de datos.

Los sistemas de administrador de base de datos pueden ser estudiados desde 3 niveles distintos:

1.- Nivel físico. Es el nivel real de los datos almacenados. Es decir como se almacenan los datos, ya sea en registros, o como sea. Este nivel lleva asociada una representación de los datos, que es lo que denominamos esquema físico.

2.- Nivel conceptual. Es el correspondiente a una visión de la base de datos desde el punto de visto del mundo real. Es decir tratamos con la entidad u objeto representado, sin

importarnos como está representado o almacenado. Este nivel lleva asociado el esquema conceptual.

3.- Nivel visión. Son partes del esquema conceptual. El nivel conceptual presenta toda la base de datos, mientras que los usuarios por lo general sólo tienen acceso a pequeñas parcelas de ésta. El nivel visión es el encargado de dividir estas parcelas. El esquema asociado a éste nivel es el Esquema de visión.

2.2.2 Modelo Entidad-Relación

El modelo de datos entidad-relación (ER) se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones. Se desarrolla para facilitar el diseño de bases de datos permitiendo la especificación de un esquema empresarial, que representa la estructura lógica global de la base de datos.

Entidad

El principal concepto del modelo ER es la *entidad*, que es una "cosa" en el mundo real con existencia independiente. Una entidad puede ser un objeto físico (una persona, un auto, una casa o un empleado) o un objeto conceptual (una compañía, un puesto de trabajo o un curso universitario).

Cada entidad tiene propiedades específicas, llamadas *atributos*, que la describen. Por ejemplo, una sala de clases tiene un nombre (19S, F20), una ubicación, un cupo máximo, etc. Una entidad particular tiene un valor para cada uno de sus atributos.

Cada uno de los atributos de una entidad posee un dominio, el que corresponde al tipo del atributo. Por ejemplo, "matrícula" tiene como dominio al conjunto de los enteros positivos y "nombre" tiene como dominio al conjunto de caracteres.

Para todo conjunto de valores de una entidad, debe existir un atributo o combinación de atributos, que identifique a cada entidad en forma única. Este atributo o combinación de atributos se denomina llave (primaria).

El concepto de un conjunto de entidades corresponde a la noción de definición de tipo en un lenguaje de programación, por tanto, una base de datos incluye una colección de conjuntos de entidades cada uno de los cuales contiene un número cualquiera de entidades del mismo tipo.

Relación

Una *relación* se puede definir como una asociación entre entidades. Una relación también puede tener atributos.

Un conjunto de relaciones es un grupo de relaciones del mismo tipo. Formalmente es una relación de $n > 2$ conjuntos de entidades (posiblemente no distintos). Si E_1, E_2, \dots, E_n son conjuntos de entidades, entonces un conjunto de relaciones R es un subconjunto de

$$\{(e_1, e_2, \dots, e_n) | e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

donde (e_1, e_2, \dots, e_n) es una relación.

La mayoría de los conjuntos de relaciones en un sistema de base de datos son binarios, aunque ocasionalmente hay conjuntos de relaciones que implican más de dos conjuntos de entidades, siempre es posible sustituir un conjunto de relaciones no binario por varios conjuntos de relaciones binarias distintos. Así, conceptualmente, podemos restringir el modelo E-R para incluir sólo conjuntos binarios de relaciones, aunque no siempre es deseable.

La función que juega una entidad en una relación se llama papel, y normalmente son implícitos y no se suelen especificar. Sin embargo son útiles cuando el significado de una relación necesita ser clarificado. Tal es el caso cuando los conjuntos de entidades de un conjunto de relaciones no son distintos. Una relación también puede tener atributos descriptivos.

Tipos de relaciones

Según el número de entidades relacionadas (o razón de cardinalidad), se pueden definir tres tipos de relaciones:

1. Relaciones Uno a Uno (1:1). Una entidad A está asociada a lo más con una entidad B, y una entidad B a lo más con una entidad A. en la figura 2.8 se muestra este tipo de relación.

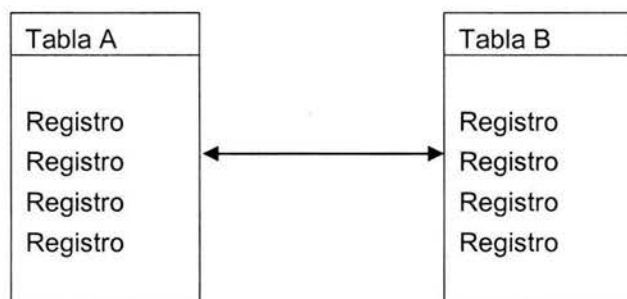


Figura 2.8

2. Relaciones Uno a Muchos (1:n). Una entidad A está asociada con una o varias entidades B. Una entidad B, sin embargo, puede estar a lo más asociada con una entidad A. tal como se muestra en la figura 2.9:

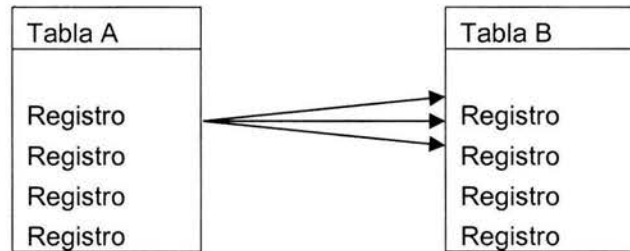


Figura 2.9

3. Relaciones Muchos a Muchos (n:m). Una entidad A está asociada con una o varias entidades B y una entidad B está asociada con una o varias entidades A. La figura 2.10 muestra este tipo de relación.

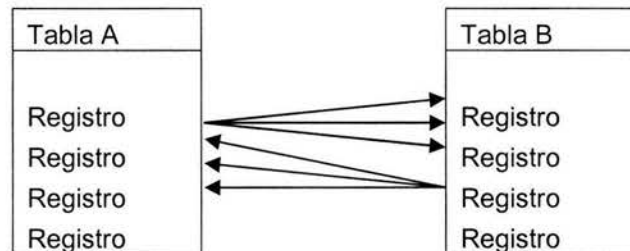


Figura 2.10

Claves

Es importante poder especificar cómo se distinguen las entidades y las relaciones. Conceptualmente, las entidades individuales y las relaciones son distintas, pero, desde la perspectiva de una base de datos, la diferencia entre ellas debe expresarse en términos de sus atributos. El concepto de superclave nos permite hacer dicha distinción.

Una superclave (clave candidata) es un conjunto de uno o más atributos que, considerados conjuntamente, nos permiten identificar de forma única a una entidad dentro del conjunto de entidades.

Usaremos el término clave primaria para denotar una clave candidata que elige el diseñador de la base de datos como el medio principal de identificar entidades dentro de un conjunto de entidades. Es posible que un conjunto de entidades no tenga atributos suficientes para formar una clave primaria. Un conjunto de entidades de este tipo se denomina conjunto de entidades débil, mientras que se denominará conjunto de entidades fuerte si tiene una clave primaria.

Los conceptos de entidades fuertes y débiles están relacionados con las dependencias por existencia. Un miembro de un conjunto de entidades fuerte es, por definición una entidad dominante, mientras que un miembro de un conjunto de entidades débil es una

entidad subordinada por definición. Aunque un conjunto de entidades débil no tiene clave primaria debe tener un medio de distinguir entre todas aquellas entidades en el conjunto de entidades que dependen de una entidad fuerte determinada.

Diagramas Entidad/Relación.

Es una forma gráfica de representar una base de datos.

Desde hace mucho tiempo, los diseñadores de sistemas, los analistas y los programadores han buscado un sistema que les permita "visualizar" una base de datos, sus tablas, el contenido de éstas, las claves y las interrelaciones con otras tablas.

Los diagramas entidad/relación permiten efectuar esta representación de forma sencilla usando pocos símbolos:

- Un *rectángulo* representa una entidad o aquello de lo que queremos guardar información.
- Un *rombo* representa la interrelación de una entidad con otra.
- Un *círculo pequeño*, unido mediante una línea a una entidad, indica un atributo (o unidad básica de información) de ésta.

Las entidades pueden ser FUERTES y DÉBILES. Una entidad FUERTE es la que existe por sí misma y la DÉBIL es la que precisa de una fuerte para su existencia. Las entidades débiles se representan con un *rectángulo dentro de otro*.

Los atributos pueden ser:

- Simples: No se pueden descomponer.
- Compuestos: Pueden ser descompuestos (por ejemplo, nombre se puede descomponer en Nombre, Primer Apellido y Segundo Apellido).
- Univalorados: Solo pueden tener un valor único.
- Multivalorados: Pueden contener varios valores (por ejemplo teléfonos).
- Nulos: Pueden tener valores nulos.
- Derivados: Se obtienen a partir de otros. (por ejemplo el importe de una línea de factura, se obtiene de Cantidad * Precio) Las interrelaciones entre entidades expresan mediante un verbo el lazo de unión existente.

2.2.3 Modelo relacional

Una base de datos relacional es un conjunto de relaciones normalizadas.

El modelo relacional representa la segunda generación de los SMDB. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Pero detrás de esa

simple estructura hay un fundamento teórico importante del que carecen los SMBD de la primera generación, lo que constituye otro punto a su favor.

El modelo relacional, como todo modelo de datos, tiene que ver con tres aspectos de los datos:

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

Tras la introducción del modelo relacional se ha desarrollado una teoría esencial para las bases de datos relacionales, que ayuda al diseño de bases de datos relacionales y al procesamiento eficiente de solicitudes de información.

Estructura de las bases de datos relacionales.

Una base de datos relacional consiste en una colección de tablas, a cada una de las cuales se les asigna un nombre único. Una fila de la tabla representa una relación entre un conjunto de valores, puesto que una tabla es una colección de dichas relaciones, hay una estrecha correspondencia entre el concepto de tabla y el concepto matemático de relación, del cual toma su nombre el modelo de datos relacional.

Relaciones.

El modelo relacional se basa en el concepto matemático de relación, que gráficamente se representa mediante una tabla. Codd, que era un experto matemático, utilizó una terminología perteneciente a las matemáticas, en concreto de la teoría de conjuntos y de la lógica de predicados.

Una relación es una tabla con columnas y filas. Un SMBD sólo necesita que el usuario pueda percibir la base de datos como un conjunto de tablas. Esta percepción sólo se aplica a la estructura lógica de la base de datos (en el nivel externo y conceptual de la arquitectura de tres niveles ANSI-SPARC). No se aplica a la estructura física de la base de datos, que se puede implementar con distintas estructuras de almacenamiento.

Un atributo es el nombre de una columna de una relación. En el modelo relacional, las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos. Una relación se representa gráficamente como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas corresponden a los campos o atributos de esos registros. Los atributos pueden aparecer en la relación en cualquier orden.

Un dominio es el conjunto de valores legales de uno o varios atributos. Los dominios constituyen una poderosa característica del modelo relacional. Cada atributo de una base

de datos relacional se define sobre un dominio, pudiendo haber varios atributos definidos sobre el mismo dominio.

El concepto de dominio es importante porque permite que el usuario defina, en un lugar común, el significado y la fuente de los valores que los atributos pueden tomar. Esto hace que haya más información disponible para el sistema cuando éste va a ejecutar una operación relacional, de modo que las operaciones que son semánticamente incorrectas, se pueden evitar. Por ejemplo, no tiene sentido comparar el nombre de una calle con un número de teléfono, aunque los dos atributos sean cadenas de caracteres. Sin embargo, el importe mensual del alquiler de un inmueble no estará definido sobre el mismo dominio que el número de meses que dura el alquiler, pero sí tiene sentido multiplicar los valores de ambos dominios para averiguar el importe total al que asciende el alquiler. Los SMDB relacionales no ofrecen un soporte completo de los dominios ya que su implementación es extremadamente compleja.

Una tupla es una fila de una relación. Los elementos de una relación son las tuplas o filas de la tabla. Las tuplas de una relación no siguen ningún orden.

El grado de una relación es el número de atributos que contiene. Esto quiere decir que cada fila de la tabla es una tupla con n valores. El grado de una relación no cambia con frecuencia.

La cardinalidad de una relación es el número de tuplas que contiene. Ya que en las relaciones se van insertando y borrando tuplas a menudo, la cardinalidad de las mismas varía constantemente.

Gráficamente se suelen representar las relaciones mediante tablas. Los nombres de las columnas corresponden a los nombres de los atributos y las filas son cada una de las tuplas de la relación. Los valores que aparecen en cada una de las columnas pertenecen al conjunto de valores del dominio sobre el que está definido el atributo correspondiente.

Propiedades de las relaciones.

Las relaciones tienen ciertas características que le permiten mantener la integridad del modelo, estas son las siguientes:

- Cada relación tiene un nombre y éste es distinto del nombre de todas las demás.
- Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están normalizadas.
- No hay dos atributos que se llamen igual.
- El orden de los atributos no importa: los atributos no están ordenados.
- Cada tupla es distinta de las demás: no hay tuplas duplicadas.
- El orden de las tuplas no importa: las tuplas no están ordenadas.

Tipos de relaciones

En un SMDB relacional pueden existir varios tipos de relaciones, aunque no todos manejan todos los tipos.

- Relaciones base. Son relaciones reales que tienen nombre y forman parte directa de la base de datos almacenada (son autónomas).
- Vistas. También denominadas relaciones virtuales, son relaciones con nombre y derivadas. Se representan mediante su definición en términos de otras relaciones con nombre, no poseen datos almacenados propios.
- Instantáneas. Son relaciones con nombre y derivadas. Pero a diferencia de las vistas, son reales, no virtuales. Están representadas no sólo por su definición en términos de otras relaciones con nombre, sino también por sus propios datos almacenados. Son relaciones de sólo lectura y se refrescan periódicamente.
- Resultados de consultas. Son las relaciones resultantes de alguna consulta especificada. Pueden o no tener nombre y no persisten en la base de datos.
- Resultados intermedios. Son las relaciones que contienen los resultados de las subconsultas. Normalmente no tienen nombre y tampoco persisten en la base de datos.
- Resultados temporales. Son relaciones con nombre, similares a las relaciones base o a las instantáneas, pero la diferencia es que se destruyen automáticamente en algún momento apropiado.

Reglas de integridad.

Los conceptos básicos de integridad en el modelo relacional son el de llave primaria, llave foránea, valores nulos y un par de reglas de integridad.

Ya que en una relación no hay tuplas repetidas, éstas se pueden distinguir unas de otras, es decir, se pueden identificar de modo único. La forma de identificarlas es mediante los valores de sus atributos.

Una *llave primaria* es uno o un conjunto de atributos que permiten identificar a las n-adas de manera única en cualquier momento.

Una *llave foránea* de una relación es un atributo que hace referencia a una llave primaria de otra relación; esto da pie a que una relación pueda tener varias llaves foráneas.

Un *valor nulo* es un valor que está fuera de la definición de cualquier dominio el cual permite dejar el valor del atributo "latente", su uso es frecuente en las siguientes situaciones:

- a) Cuando se crea una n-ada y no se conocen todos los valores de cada uno de los atributos.
- b) Cuando se agrega un atributo a una relación ya existente.
- c) Para no tomarse en cuenta al hacer cálculos numéricos.

Las dos reglas de integridad tienen que ver precisamente con los conceptos antes mencionados y son:

- a) Integridad de Relaciones. Ningún atributo que forme parte de una llave primaria puede aceptar valores nulos.
- b) Integridad Referencial. Al tener una relación con llave primaria y otra relación con atributo que no es llave primaria, entonces cualquier valor en el atributo debe ser nulo o un valor que esté en el atributo de la llave primaria.

Lenguajes relacionales.

Otra parte muy importante de un modelo de datos es la manipulación. Son varios los lenguajes utilizados por los SDBD relacionales para manejar las relaciones. Algunos de ellos son *procedurales*, lo que quiere decir que el usuario dice al sistema exactamente cómo debe manipular los datos. Otros son *no procedurales*, que significa que el usuario dice qué datos necesita, en lugar de decir cómo deben obtenerse.

A continuación se presentará el álgebra relacional y el cálculo relacional, definidos por Codd como la base de los lenguajes relacionales. Se puede decir que el álgebra es un lenguaje procedural (de alto nivel), mientras que el cálculo relacional es un lenguaje no procedural. Sin embargo, ambos lenguajes son equivalentes: para cada expresión del álgebra, se puede encontrar una expresión equivalente en el cálculo y viceversa.

El álgebra relacional (o el cálculo relacional) se utilizan para medir la potencia de los lenguajes relacionales. Si un lenguaje permite obtener cualquier relación que se pueda derivar mediante el álgebra relacional, se dice que es *relacionalmente completo*. La mayoría de los lenguajes relacionales son relacionalmente completos, pero tienen más potencia que el álgebra o el cálculo porque se les han añadido operadores especiales. Tanto el álgebra como el cálculo son lenguajes formales no muy "amigables". Pero se deben estudiar porque sirven para ilustrar las operaciones básicas que todo lenguaje de manejo de datos debe ofrecer. Además, han sido la base para otros lenguajes relacionales de manejo de datos de más alto nivel.

Álgebra relacional

Las operaciones de álgebra relacional manipulan relaciones. Esto significa que estas operaciones usan uno o dos relaciones existentes para crear una nueva relación. Esta nueva relación puede entonces usarse como entrada para una nueva operación. Este poderoso concepto - la creación de una nueva relación a partir de relaciones existentes - hace considerablemente más fácil la solución de las consultas, debido a que se puede experimentar con soluciones parciales hasta encontrar la proposición con la que se trabajará.

Unión

La operación de unión permite combinar datos de varias relaciones. No siempre es posible realizar consultas de unión entre varias tablas, para poder realizar esta operación es necesario e imprescindible que las tablas a unir tengan las mismas estructuras y que sus campos sean iguales.

Intersección

La operación de intersección permite identificar filas que son comunes en dos relaciones.

Diferencia

La operación diferencia permite identificar filas que están en una relación y no en otra.

Producto

La operación producto consiste en la realización de un producto cartesiano entre dos tablas dando como resultado todas las posibles combinaciones entre los registros de la primera y los registros de la segunda.

Selección

La operación selección consiste en recuperar un conjunto de registros de una tabla o de una relación indicando las condiciones que deben cumplir los registros recuperados, de tal forma que los registros devueltos por la selección han de satisfacer todas las condiciones que se hayan establecido. Esta operación es la que normalmente se conoce como consulta.

En este tipo de consulta se emplean los diferentes operadores de comparación (=, >, <, >=, <=, <>), los operadores lógicos (and, or, xor) o la negación lógica (not).

Proyección

Una proyección es un caso concreto de la operación selección, esta última devuelve todos los campos de aquellos registros que cumplen la condición que he establecido. Una proyección es una selección en la que seleccionamos aquellos campos que deseamos recuperar.

Reunión

La reunión se utiliza para recuperar datos a través de varias tablas conectadas unas con otras mediante cláusulas JOIN, en cualquiera de sus tres variantes INNER, LEFT, RIGHT. La operación reunión se puede combinar con las operaciones selección y proyección.

División

La operación división es la contraria a la operación producto.

Asignación

Esta operación algebraica consiste en asignar un valor a uno o varios campos de una tabla.

Cálculo relacional

El cálculo relacional usa un enfoque completamente diferente al álgebra relacional. No obstante, los dos lenguajes son lógicamente equivalentes. Esto significa que cualquier consulta que pueda resolverse en un lenguaje puede resolverse en el otro. La solución para toda consulta en este tipo de cálculo se define por:

- a) Una lista de resultados
- b) Una sentencia de cualificación

La lista de resultados está clara, son aquellos registros que cumplen las condiciones que deseamos. La sentencia de cualificación contiene las condiciones que deseamos que cumplan los registros de la lista de resultados. La diferencia entre el cálculo y el álgebra radica en que el cálculo realiza la operación en un único paso, sin necesidad de tener que obtener tablas intermedias, el álgebra realiza las operaciones paso a paso.

Normalmente el cálculo relacional se apoya en algún lenguaje de interrogación de bases de datos como puede ser el SQL.

El cálculo relacional incluye un concepto nuevo denominado cuantificador, los cuantificadores tratan de averiguar el número de registros afectados por una determinada operación, incluso antes de realizarla. Según su naturaleza los podemos dividir en dos grupos:

- a) Cuantificadores existenciales.- Son aquellos que tratan de averiguar el número de registros que devolvería un tipo de consulta.
- b) Cuantificadores universales.- Son aquellos que indican que una condición se aplica a todas las filas de algún tipo. Se usa para brindar la misma capacidad que la operación división del álgebra relacional.

-
- FUNDAMENTOS DE BASES DE DATOS, Henry F. Korth, Abraham Silberschatz, McGraw-Hill, México 1993
 - FUNDAMENTOS Y MODELOS DE BASES DE DATOS, Adoración de Miguel, Mario Piattini, Alfaomega Grupo Editor, Colombia 1998
 - DISEÑO Y GESTIÓN DE SISTEMAS DE BASES DE DATOS, Angel Lucas Gomez, Paloma Romera García, Editorial Paraninfo sa, España 1993

2.2.4 Modelo Cliente-Servidor

Los ordenadores personales y los paquetes de software de aplicaciones proliferan comercialmente. Estos ordenadores, también conocidos como estaciones de trabajo programables, están conectados a las Redes de Área Local (LAN), mediante las cuales, los grupos de usuarios y profesionales comparten aplicaciones y datos. Las nuevas tecnologías de distribución de funciones y datos en una red, permiten desarrollar aplicaciones distribuidas de una manera transparente, de forma que múltiples procesadores de diferentes tipos (ordenadores personales de gama baja, media y alta, estaciones de trabajo, mini computadoras o incluso mainframes), puedan ejecutar partes distintas de una aplicación.

Años atrás, aún cuando no se hablaba mucho y se hacía muy poco sobre el tema, se tenían razones para establecer que el desarrollo de aplicaciones cliente/servidor era inevitable, algunas de estas razones son:

- En muchas situaciones es más eficiente que el procesamiento centralizado, dado que éste experimenta una reducción en costos económicos cuando aumenta mucho la cantidad de usuarios.
- Existían ya en ese momento servidores razonablemente eficientes y confiables.
- Se había establecido un estándar para una interfase cliente/servidor (el ODBC para SQL, adoptado por todos los fabricantes importantes de servidores).
- Era imprescindible, para apoyar con información de manera veraz y ágil a la creciente cantidad de empresas e instituciones que necesitan tomar decisiones.

En los últimos años los servidores se han mostrado sólidos y eficientes, además de que sus optimizadores probaron, en general, ser excelentes. Una cantidad muy importante de empresas, en todo el mundo, ha encarado aplicaciones cliente/servidor y quienes lo están haciendo con los planes necesarios y con las herramientas adecuadas, están obteniendo éxitos muy importantes, mientras los que lo hicieron desaprensivamente, han cosechado fracasos.

Hoy en día, el modelo cliente/servidor se considera clave para abordar las necesidades de las empresas. El proceso distribuido se reconoce actualmente como el nuevo paradigma de sistemas de información, en contraste con los sistemas independientes. Este cambio fundamental ha surgido como consecuencia de importantes factores (negocio, tecnología, proveedores), y se apoya en la existencia de una gran variedad de aplicaciones estándar y herramientas de desarrollo, fáciles de usar que soportan un entorno informático distribuido.

Modelo cliente-servidor

El concepto de cliente/servidor proporciona una forma eficiente de utilizar todos los recursos de una máquina, de tal forma que la seguridad y fiabilidad que proporcionan los

entornos mainframe se traspassa a la red de área local. A esto hay que añadir la ventaja de la potencia y simplicidad de los ordenadores personales.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina *cliente* al proceso que inicia el diálogo o solicita los recursos y *servidor*, al proceso que responde a las solicitudes.

Es el modelo de interacción más común entre aplicaciones en una red. No forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo. Los principales componentes del esquema cliente/servidor son entonces los Clientes, los Servidores y la infraestructura de comunicaciones, en este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfase del usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Los Servidores proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloqueos, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores, la plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Por esta razón se utilizan PC'S poderosas, estaciones de trabajo, mini computadores o sistemas grandes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema, auditoria, recuperación y contabilidad. Usualmente en los servidores existe algún tipo de servicio de bases de datos. En ciertas circunstancias, este término designará a una máquina. Este será el caso si dicha máquina está dedicada a un servicio particular, por ejemplo: servidores de impresión, servidor de archivos, servidor de correo electrónico, etc.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.
- Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le responde proporcionándolo normalmente, pero no necesariamente.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas cliente/servidor actuales, se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Entre las principales características de la arquitectura cliente/servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfase única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfase externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Componentes esenciales de la infraestructura Cliente/Servidor.

Una infraestructura cliente/servidor consta de tres componentes esenciales, todos ellos estrechamente ligados y con igualdad de importancia:

Plataforma operativa. La plataforma deberá soportar todos los modelos de distribución cliente/servidor, todos los servicios de comunicación, y deberá utilizar, preferentemente, componentes estándar de la industria para los servicios de distribución. Los desarrollos propios deben coexistir con las aplicaciones estándar y su integración deberá ser imperceptible para el usuario. Igualmente, podrán acomodarse programas escritos utilizando diferentes tecnologías y herramientas.

Entorno de desarrollo de aplicaciones. Este se elige después de la plataforma operativa. Aunque es conveniente evitar la proliferación de herramientas de desarrollo, se garantizará que el enlace entre éstas y el middleware no sea excesivamente rígido. Será posible utilizar diferentes herramientas para desarrollar partes de una aplicación. Un entorno de aplicación incremental, debe posibilitar la coexistencia de procesos cliente y servidor desarrollados con distintos lenguajes de programación y/o herramientas, así como utilizar distintas tecnologías (por ejemplo, lenguaje procedural, lenguaje orientado a objetos, multimedia), y que han sido puestas en explotación en distintos momentos del tiempo.

Gestión de sistemas. Estas funciones aumentan considerablemente el costo de una solución. Siempre deben adaptarse a las necesidades de la organización, y al decidir la plataforma operativa y el entorno de desarrollo, es decir, en las primeras fases de la definición de la solución.

No es en absoluto recomendable el intento de una infraestructura completa desde el principio, ya que las tecnologías pueden no responder a tiempo a las necesidades. El

enfoque más adecuado está en un sistema y una plataforma de aplicación conceptual, y una arquitectura construida incrementalmente y ampliada a medida que se desarrollan nuevas aplicaciones.

Características funcionales.

Esta arquitectura se puede clasificar en cinco niveles, según las funciones que asumen el cliente y el servidor, en la figura 2.11 se pueden observar estos cinco niveles:

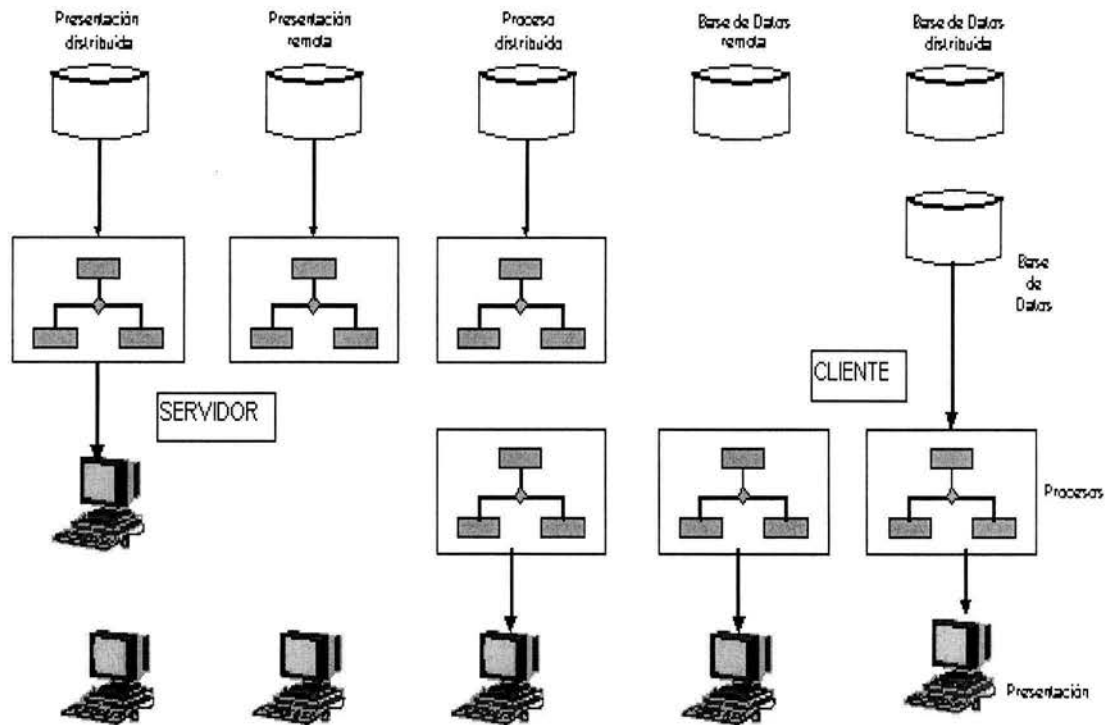


Figura 2.11 Niveles funcionales de la arquitectura cliente – servidor

En el primer nivel el cliente asume parte de las funciones de presentación de la aplicación, ya que siguen existiendo programas en el servidor, dedicados a esta tarea. Dicha distribución se realiza mediante el uso de productos para el "maquillaje" de las pantallas del mainframe. Esta técnica no exige el cambio en las aplicaciones orientadas a terminales, pero dificulta su mantenimiento. Además, el servidor ejecuta todos los procesos y almacena la totalidad de los datos. En este caso se dice que hay una *presentación distribuida* o embellecimiento.

En el segundo nivel, la aplicación está soportada directamente por el servidor, excepto la presentación que es totalmente remota y reside en el cliente. Los terminales del cliente soportan la captura de datos, incluyendo una validación parcial de los mismos y una presentación de las consultas. En este caso se dice que hay una *presentación remota*.

En el tercer nivel, la lógica de los procesos se divide entre los distintos componentes del cliente y del servidor. El diseñador de la aplicación debe definir los servicios y las

interfaces del sistema de información, de forma que los papeles de cliente y servidor sean intercambiables, excepto en el control de los datos, que es responsabilidad exclusiva del servidor. En este tipo de situaciones se dice que hay un *proceso distribuido o cooperativo*.

En el cuarto nivel el cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada. En esta situación se dice que hay una *gestión de datos remota*.

En el quinto y último nivel, el reparto de tareas es como en el anterior y además el gestor de base de datos divide sus componentes entre el cliente y el servidor. Las interfaces entre ambos, están dentro de las funciones del gestor de datos y, por lo tanto, no tienen impacto en el desarrollo de las aplicaciones. En este nivel se da lo que se conoce como *bases de datos distribuidas*.

Características físicas.

La figura 2.12 proporciona una idea de la estructura física de conexión entre las distintas partes que componen una arquitectura cliente/servidor. La idea principal consiste en aprovechar la potencia de los ordenadores personales para realizar, sobre todo, los servicios de presentación y, según el nivel, algunos procesos o incluso algún acceso a datos locales. De esta forma se descarga al servidor de ciertas tareas para que pueda realizar otras más rápidamente.

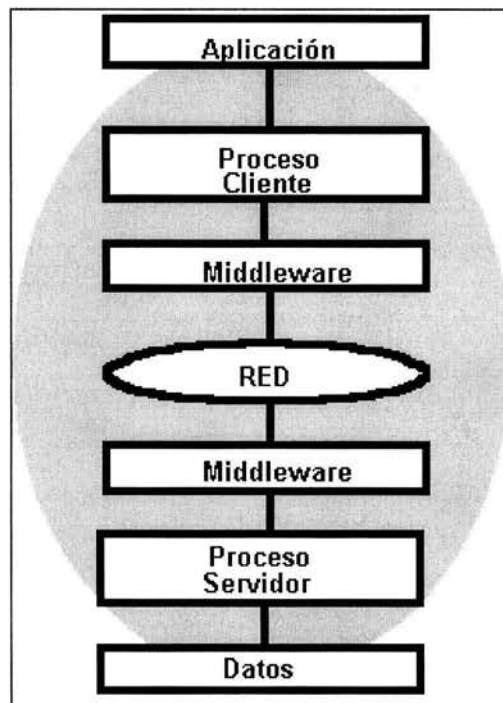


Figura 2.12 Estructura Física del modelo cliente/servidor

Existe también una plataforma de servidores que sustituye al ordenador central tradicional y que da servicio a los clientes autorizados. Incluso a veces el antiguo ordenador central se integra en dicha plataforma como un servidor más. Estos servidores suelen estar especializados por funciones (seguridad, cálculo, bases de datos, comunicaciones, etc.), aunque, dependiendo de las dimensiones de la instalación se pueden reunir en un servidor una o varias de estas funciones.

Las unidades de almacenamiento masivo en esta arquitectura, se caracterizan por incorporar elementos de protección que evitan la pérdida de datos y permiten multitud de accesos simultáneos.

Para la comunicación de todos estos elementos se emplea un sistema de red que se encarga de transmitir la información entre clientes y servidores. Físicamente consiste en un cableado (coaxial, par trenzado, fibra óptica, etc.) o en conexiones mediante señales de radio o infrarrojos, dependiendo de que la red sea local (LAN), metropolitana (MAN) o de área extensa (WAN).

Para la comunicación de los procesos con la red se emplea un tipo de equipo lógico denominado middleware que controla las conversaciones. Su función es independizar ambos procesos (cliente y servidor). La interfase que presenta es la estándar de los servicios de red, hace que los procesos "piensen" en todo momento que se están comunicando con una red.

Características lógicas.

Una de las principales aportaciones de esta arquitectura a los sistemas de información, es la interfaz gráfica de usuario. Gracias a ella se dispone de un manejo más fácil e intuitivo de las aplicaciones. En esta arquitectura los datos se presentan, editan y validan en la parte de la aplicación cliente.

En cuanto a los datos, cabe señalar que en la arquitectura cliente/servidor se evitan las duplicidades (copias y comparaciones de datos), teniendo siempre una imagen única y correcta de los mismos, disponible en línea para su uso inmediato.

Todo esto tiene como fin que el usuario de un sistema de información soportado por una arquitectura cliente/servidor, trabaje desde su estación de trabajo con distintos datos y aplicaciones, sin importarle dónde están o dónde se ejecuta cada uno de ellos.

Fases de implantación.

Una arquitectura cliente / servidor debe mostrar los sistemas de información no como un cliente que accede a un servidor corporativo, sino como un entorno que ofrece acceso a una colección de servicios.

Para llegar a este estado pueden distinguirse las siguientes fases de evolución del sistema:

Fase de Iniciación.- Esta etapa se centra sobre todo en la distribución física de los componentes entre plataformas. Los dos tipos de plataforma son:

- Una plataforma cliente para la presentación (generalmente un ordenador personal de sobremesa).
- Una plataforma servidora (como por ejemplo el servidor de una base de datos relacional) para la ejecución de procesos y la gestión de los datos.

En esta fase pueden surgir los siguientes problemas:

- Cómo repartir la lógica de la aplicación entre las plataformas cliente y servidor de la forma más conveniente.
- Cómo gestionar la arquitectura para que permita que cualquier cliente se conecte con cualquier servidor.

Fase de Proliferación.- La segunda etapa de una arquitectura cliente/servidor se caracteriza por la proliferación de plataformas clientes y servidoras. Ahora, el entorno para la interacción entre clientes y servidores se hace mucho más complejo. Puede hacerse una distinción entre:

- Datos de servidores a los que se accede a través de una red de área extensa (conocida como WAN) y
- Datos a los que se accede a través de una red de área local (conocida como LAN).

Los mecanismos de conexión son muy variados y suelen ser incompatibles. En esta fase los problemas que se pueden plantear son:

- La gestión de accesos se convierte en crítica y compleja, debido a la estructura del organismo donde se está implantando la arquitectura. El mercado ofrece algunas soluciones que mejoran la interoperabilidad y que se basan en conexiones modulares que utilizan, entre otros:
 - Drivers en la parte cliente.
 - Pasarelas (gateways) a bases de datos.
 - Especificaciones de protocolos cliente / servidor, etc.
- Los requisitos de actualización de datos pasan a formar parte de los requisitos solicitados al sistema cliente/servidor. Ahora no sólo se consultan datos, sino que se envían peticiones para actualizar, insertar y borrar datos.

Fase de Control.- En esta fase se consolidan los caminos de acceso desde una plataforma cliente particular, a una plataforma servidora particular. Los conceptos en los que se debe poner especial énfasis son los siguientes:

- **Transparencia en la localización.** Significa que la aplicación cliente no necesita saber nada acerca de la localización (física o lógica) de los datos o los procesos. La localización de los recursos debe estar gestionada por servidores y estar representada en las plataformas adecuadas, de forma que se facilite su uso por parte de las plataformas cliente.
- **Gestión de copias.** El sistema se debe configurar de forma que se permita copiar la información (datos o procesos) de los servidores.
- **Especialización de los equipos servidores,** en servidores de bases de datos o en servidores de aplicaciones. Los servidores de bases de datos continúan ofreciendo servicios orientados a datos a través de llamadas SQL o a través de procedimientos almacenados. En cualquier caso, los servicios se orientan a mantener la integridad de los datos. Por otro lado, los servidores de aplicaciones se centran en los procesos, implementando partes de la lógica de la aplicación en la parte servidora.

Fase de Integración.- Esta etapa se caracteriza por el papel conjunto que juegan la gestión de accesos, la gestión de copias y la gestión de recursos. La gestión de la información se debe realizar de forma que se pueda entregar la información controlada por los servidores que contienen los datos a las plataformas clientes que los requieran. El concepto en que se basa este tipo de gestión es la distinción entre dos tipos de datos: datos de operación y datos de información. Para ajustarse a los posibles cambios en los procesos, los datos de operación varían continuamente, mientras que los datos de información son invariables porque son de naturaleza histórica y se obtienen tomando muestras en el tiempo, de los datos de operación.

Fase de Madurez.- Esta es la etapa final de una arquitectura cliente/servidor. Se caracteriza por una visión más flexible de las plataformas físicas del sistema que se contemplan como una única unidad lógica. Este estado también se caracteriza porque la tecnología cliente/servidor se ha generalizado en la empresa. Ya no es un problema saber qué componentes se distribuyen en qué plataformas, porque los recursos se pueden redistribuir para equilibrar la carga de trabajo y para compartir los recursos de información. Lo fundamental aquí, es saber quién ofrece qué servicios. Para ello es necesario distinguir qué tipo de servicios y recursos se demandan y conocer las características de esta arquitectura basada en servicios.

En la fase de integración veíamos que se establecía una distinción entre datos de operación y datos de información histórica. Por contra, en un entorno de operación cliente/servidor que se encuentre en la fase de madurez, lo interesante es distinguir entre dos nuevos términos: organismo y grupo de trabajo. Esta distinción se establece basándose en sus diferencias organizativas. El grupo de trabajo es el entorno en el que grupos organizados de personas se centran en tareas específicas de la actividad del

organismo al que pertenecen. Estos equipos de personas requieren una información propia y unas reglas de trabajo particulares, que pueden ser diferentes de las del organismo en su globalidad.

Una arquitectura basada en servicios es la que se contempla como una colección de consumidores de servicios poco relacionados entre sí y los productores de dichos servicios. La utilización de este tipo de arquitectura permite pensar en nuevos retos de diseño:

- Desarrollo de componentes reutilizables entre distintas aplicaciones y distintos grupos de trabajo
- Desarrollo de aplicaciones distribuidas
- Gestión del desarrollo de aplicaciones entre distintos equipos, etc.

2.2.5 Normalización

El proceso de normalización es un estándar que consiste, básicamente, en un proceso de conversión de las relaciones entre las entidades, evitando:

- La redundancia de los datos: repetición de datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

El proceso de normalización nos conduce hasta el modelo físico de datos y consta de varias fases denominadas formas normales, estas formas se detallan a continuación.

Primera forma normal (1NF).

Se dice que una tabla se encuentra en primera forma normal (1NF) si y solo si cada uno de los campos contiene un único valor para un registro determinado. Por ejemplo, supongamos que deseamos crear una tabla en la que deseamos almacenar los productos pertenecientes a cada marca, podríamos considerar el siguiente diseño de la tabla 2.1:

Código	Marca	Productos
1	Muebles 1	Mesas
2	Muebles 2	Sillas
3	Muebles 3	Escritorios

Tabla 2.1

Podemos observar que cada registro de código 1, 2, etc. cumple la primera forma normal, cada campo del registro contiene un único dato. Una vez normalizada la tabla se puede pasar a comprobar la segunda forma normal.

Segunda forma normal (2NF).

La segunda forma normal compara todos y cada uno de los campos de la tabla con la clave definida. Si todos los campos dependen directamente de la clave se dice que la tabla está es segunda forma normal (2NF).

Supongamos que construimos una tabla con los años que cada alumno ha estado en una escuela (tabla 2.2):

Código Alumno	Código Escuela.	Nombre	Escuela	Años
1	6	Juan	Primaria	6
2	3	Pedro	Secundaria	3
3	2	Sonia	Preparatoria	1
4	3	Verónica	Secundaria	3
2	6	Pedro	Primaria	6

Tabla 2.2

Tomando como punto de partida que la clave de la tabla 2.2 está formada por los campos código de alumno y código escuela, aseguramos que la tabla se encuentra en primera forma normal, por tanto vamos a estudiar la segunda:

El campo nombre no depende funcionalmente de toda la clave, sólo depende del código del alumno. El campo escuela solo depende del código del escuela, pero el campo años si que depende funcionalmente de la clave ya que depende del código del alumno y del código del escuela.

Por tanto, al no depender todos los campos de la totalidad de la clave la tabla no está en segunda forma normal, la solución es la siguiente:

Código Alumno	Nombre
1	Juan
2	Pedro
3	Sonia
4	Verónica

Tabla 2.3

Código Escuela	Escuela
2	Preparatoria
3	Secundaria
6	Primaria

Tabla 2.4

Código Alumno	Código Escuela	Años
1	6	6
2	3	3
3	2	1
4	3	10
2	6	5

Tabla 2.5

Ahora podemos observar que la tabla 2.5 si se encuentra en segunda forma normal, considerando que la tabla 2.3 tiene como índice el campo Código Alumno, la tabla 2.4 Código Escuela y la tabla 2.5 una clave compuesta por estos dos.

Tercera forma normal (3NF).

Se dice que una tabla está en tercera forma normal si y solo si los campos de la tabla son dependientes de la clave, dicho en otras palabras los campos de las tablas no dependen unos de otros. Por ejemplo supongamos que cada alumno sólo puede realizar un único curso a la vez y que deseamos guardar en que aula se imparte el curso. La estructura podría ser la siguiente:

Código	Nombre	Curso	Aula
1	Marcos	Informática	Aula A
2	Lucas	Inglés	Aula B
3	Marta	Contabilidad	Aula C

Tabla 2.6

Ahora veamos la dependencia de cada campo con respecto a la clave (código):

Nombre es dependiente directo del código del alumno.

Curso depende de igual forma del código del alumno.

El aula esta más ligado al curso que el alumno está realizando.

Por esta última razón la tabla no cumple con la 3NF. Para solucionar este problema se realizaría lo siguiente:

Código	Nombre	Curso
1	Marcos	Informática
2	Lucas	Inglés
3	Marta	Contabilidad

Tabla 2.7

Curso	Aula
Informática	Aula A
Inglés	Aula B
Contabilidad	Aula C

Tabla 2.8

Como podemos observar ahora si tanto el campo nombre como aula, son dependientes directos de la clave de la tabla a la cual pertenecen.

Cuarta forma normal (4NF).

Una tabla está en cuarta forma normal si y sólo si para cualquier combinación clave - campo no existen valores duplicados.

Vestidura		
Vestimenta	Color	Tamaño
camisa	Rojo	Grande
camisa	Azul	Grande
camisa	Azul	Mediano
pantalón	Blanco	Mediano
pantalón	Azul	Pequeño
pantalón	Azul	Mediano

Tabla 2.8

Como podemos ver, a simple vista podemos detectar las distintas repeticiones de información existente, por ejemplo se observa que camisa grande está repetido; de igual forma pasa con pantalón azul. Estas repeticiones son las que se evitan que una tabla cumpla con la 4NF.

Para lograr que esta tabla cumpla con la 4NF, la solución sería la siguiente:

Tamaño	
Figura	Tamaño
camisa	Grande
camisa	Pequeño
pantalón	Mediano
pantalón	Pequeño

Tabla 2.9

Color	
Figura	Color
camisa	Rojo
camisa	Azul
pantalón	Blanco
pantalón	Azul

Tabla 2.10

Con los cambios realizados, observamos que ahora si no existe la duplicidad de información en ninguna de las tablas, por lo tanto ahora si hay una base en 4NF.¹

2.3 INTERNET

El término Internet procede de las palabras inglesas interconnection y network; esto significa que Internet son miles de redes interconectadas. Estas redes están formadas por diferentes tipos de ordenadores y de diferentes tecnologías.

Técnicamente Internet es una red formada por la interconexión cooperativa de redes de ordenadores. Es decir: Internet es una red de redes.

Internet basa su utilidad básicamente en cuatro servicios: World Wide Web, correo electrónico, servicio de noticias y transferencia de ficheros.

El World Wide Web facilita a los usuarios el acceso a servidores de información. Esta información es presentada en formato multimedia, y puede contener textos, imágenes, sonidos, vídeo, animación, aplicaciones interactivas, formularios, etc. Son millones los servidores de información a los que nos podemos conectar. Mediante el Web se puede acceder a múltiples servicios de todo tipo: acceder a la programación de emisoras de radio o TV, establecer videoconferencias, acceder al banco para ver nuestros movimientos o hacer pagos o transferencias, comprar, vender, alquilar o subastar productos o servicios, etc., todo esto sin limitación geográfica.

¹ <http://programacion.com>
<http://es.tldp.org>

Con el correo electrónico, cada usuario puede ponerse en contacto con cualquier otro a lo largo de todo el mundo e intercambiar información con él, mensajes, ficheros, experiencia, etc.

Con el servicio de noticias, cada usuario puede suscribirse a los temas que más le interesen y diariamente tendrá acceso a multitud de información sobre esos temas. Las informaciones del servicio de noticias son generadas por los propios usuarios, y cada uno de ellos puede aportar sus propios ensayos sobre cualquier tema.

Por último, mediante el servicio de transferencia de ficheros, cualquier usuario puede recuperar ficheros desde cualquier ordenador conectado a la red. Con este procedimiento podemos tener acceso directo, tanto a aplicaciones informáticas gratuitas, como a fotografías, vídeos o documentos sobre los temas más interesantes.

2.3.1 Sistemas operativos de servidores Web.

Los dos sistemas operativos sobre los que basan la mayoría de las páginas Web visibles en Internet son Linux y Windows 2000 Server. Linux es un sistema operativo libre y gratuito, es muy estable y fiable. Windows 2000 Server es el sistema operativo diseñado por Microsoft específicamente para servidores, seguro, versátil y con tecnología muy avanzada.

Windows NT (Windows New Technology).

Este sistema operativo de 32 bits fue desarrollado originalmente para que sea OS/2.3.0 antes de que Microsoft e IBM se distanciaran. NT se diseñó para estaciones de trabajo avanzado (Windows NT) y para servers (Windows NT Advanced Server).

Windows NT fue lanzado en 1993. A diferencia de Windows 3.1 que corría con una interfase gráfica sobre MS-DOS, Windows NT es un sistema operativo por sí solo.

Esta basado en un MicroKernel con un direccionamiento de hasta 4gb de RAM, soporte para sistemas de archivos FAT, NTFS y HPFS, soporte de red incorporado, soporte para múltiple procesador y seguridad C2.

NT esta diseñado para ser independiente del Hardware. Una vez que la capa HAL (capa de abstracción de hardware) a sido llevada a una maquina en particular, el mismo sistema operativo debería compilarse sin problemas

Características:

- Robustez: es un SO estable y robusto, que impide a aplicaciones mal escritas estropear el resto del sistema.

- Seguridad: ha sido escrito para satisfacer las necesidades de seguridad de organismos oficiales y empresas cuyos datos deben estar bien protegidos contra personas no autorizadas
- Posee un sistema en donde cada usuario tiene privilegios sobre los datos a los cuales desee acceder.
- Portabilidad: el sistema permite que sea adaptable a otros tipos de arquitectura para las cuales no fue desarrollado. Soporta Intel X86, MIPS, ALPHA y POWERPC. Su diseño modular y su lenguaje fácil de entender como es el C le permite la rápida migración.
- Compatibilidad con las aplicaciones Windows: tiene la capacidad de correr aplicaciones Windows y DOS, que le permite obtener un rendimiento mayor. Las aplicaciones Win32 corren en modo nativo en las diferentes plataformas NT simplemente precompilándolas o incluso a través de emuladores o compiladores JIT (Just In Time) como son el X86.
- Velocidad: la plataforma NT permite hacerle frente a las aplicaciones que necesitan grandes requerimientos y altas velocidades de ejecución, típicas de servicios clientes/servidor, como lo puede ser un servidor de recursos de red o de base de datos.
- La interfase de Windows NT 3.x es parecida a la de Windows 3.x mientras que la interfase de Windows NT 4.0 se emplea la interfase de Windows 95/98

Versiones de Windows NT:

- NT WorkStation: está configurada para estaciones de trabajo, donde se ejecutaran las aplicaciones de usuarios. Es una plataforma que incluye todos los elementos para trabajar con archivos de Windows y en red, con una pila completa para TCP/IP.
- NT Server: como la palabra lo indica esta preparado para configurar servidores
- Es más robusto para tareas de servidor de red. Ofrece mayor seguridad en el almacenamiento de datos y el manejo de errores.

Ventajas de NT:

- Es multitarea y multiusuario (como todos los Windows a partir del 3.1)
- Apoya el uso de múltiples procesadores
- Soporta diferentes arquitecturas
- Soporta acceso remoto y ofrece mucha seguridad en los accesos.
- Soporta varios protocolos:
 - NetBEUI
 - TCP/IP
 - IPX/SPX
 - Banyan
 - DECnet
 - Apple Talk
- Trabaja con impresoras de estaciones remotas
- Brinda la posibilidad de dar diferentes permisos y privilegios a los usuarios
- No permite criptografía de llave publica ni privada

Desventajas de NT:

- Tiene ciertas limitaciones en cuanto a la RAM.
- No soporta archivos NFS.
- No ofrece bloqueo de intrusos.
- No soporta la ejecución de algunas aplicaciones de DOS.
- Requiere como mínimo 16 megas de RAM y un procesador Pentium 133 MHZ o superior.

Windows 2000.

Se basa en el poder de Windows NT, brinda mayor seguridad, mayor confiabilidad, y la habilidad de operar de manera integrada la red.

Características:

- Mayor grado de confiabilidad: mejora la seguridad de archivos más que en NT WorkStation.
- Elimina los 45 escenarios más comunes de reinicio
- Soluciona mas rápidamente y eficazmente los errores.
- Brinda 3 tipos de protección:
 - Local: El sistema de codificación de archivos (EFS) corre como un sistema integrado de seguridad, haciendo más difícil los ataques y más fácil su administración.
 - Empresarial: protege el acceso a la red a personas no autorizadas. Soporta el protocolo Kerberos v.5
 - Publico: protege las conexiones de la intranet. Soporta la seguridad por medio clave publica
- Soporta tarjetas inteligentes: útiles en cuanto a la privacidad, autenticación, registro, y correo electrónico.
- Capacidad de operar de manera integrada con otros ambientes: no importa que servidor se use ya que se puede manejar de manera mixta tanto servidores Unix, Novell NetWare o Windows NT.
- Tiene mejor desempeño: una memoria escalable y soporte de procesadores.
- Mejor desempeño en cuanto a la fuente de energía.
- Acceso más rápido a la información de su computadora e Internet.

LINUX.

Linux es una versión de Unix libremente distribuible e independiente, para plataformas con maquinas x86, Motorola 68k, Digital Alpha, Sparc, Mips y Motorola Power PC. Es una implantación de la especificación POSIX con la cual cumplen todas las verdaderas versiones de Unix. El núcleo de Linux no usa código de AT&T o de cualquier otra fuente propietaria, la mayoría de los programas disponible para Linux es desarrollado por el proyecto GNU de la Free Software Foundation (Fundación de programas Libres).

Éste soporta un amplio espectro de aplicaciones o paquetes de programación tales como X Windows, Emacs, redes de datos bajo protocolos TCP/IP (incluyendo SLIP, PPP, ISDN). Mucha gente ha corrido benchmarks en sistemas Linux 80486 consiguiendo a estos comparable con estaciones de rango-medio de Sun y Digital.

Linux esta disponible en Internet en cientos de servidores ftp y de varios distribuidores en discos flexibles o CD-rom, algunos de éstos son Caldera, Debian, Slakware, Red Hat, entre otros.

El núcleo de Linux esta legalmente protegido por la Licencia Publica GNU (GPL de sus siglas en ingles), es usualmente empaquetado con varios ejecutables que comprenden un sistema operativo Unix funcional.

Ambos sistemas operativos cuentan con partidarios y detractores, convirtiéndose a veces en una disputa ideológica lo que simplemente deberían ser diferencias técnicas o gustos personales.

2.3.2 Servidores Web.

Los Servidores Web son aquellos que permite a los clientes compartir datos, documentos y multimedia en formato Web.

Aunque parte de la tecnología Cliente-Servidor, el servidor Web aporta unas ventajas adicionales en aspectos muy importantes como son:

- El Web se crea normalmente como un sistema abierto al que cualquiera puede contribuir y acceder desde cualquier punto de la red de la empresa: no requieren logins ni password como los sistemas tradicionales cliente/servidor.
- Los Servidores Web vuelcan información con un simple clic del ratón a través de un proceso de hipervínculo.
- La información servida puede ser de cualquier tipo (datos, documentos, multimedia, etc.), gracias a la utilización de los estándares de Internet. Esa información es de *sólo lectura* pues, a diferencia de los sistemas normales cliente/servidor, el usuario no puede hacer cambios en el dispositivo original de los datos.
- Dado que el Servidor de Web es de tan fácil acceso, ello hace posible publicar información de forma instantánea en toda la compañía mediante un simple almacenamiento de la misma en el servidor.
- Un Servidor de Web en la Intranet puede servir para mostrar la misma copia de un archivo a toda la organización, de la misma forma que un único servidor Web en Internet puede servir al mundo entero. Así que ahora sólo hay una única copia del archivo a actualizar.

- La amplitud de la red suele ser mayor que otros sistemas cliente/servidor, pues la mayor parte de Intranet se construyen sobre Redes de Área Extensa (WAN), más que sobre redes LAN.

Apache.

Apache es uno de los servidores de páginas más utilizados, posiblemente porque ofrece instalaciones sencillas para sitios pequeños y si se requiere es posible expandirlo hasta el nivel de los mejores productos comerciales. Si se utiliza para un sitio pequeño que solo contenga archivos en HTML, esto es, no requiera de aplicaciones de servidor su funcionalidad es excelente, pero cuando se requiere una *aplicación de Servidor* implica lo que se muestra en la figura 2.13.

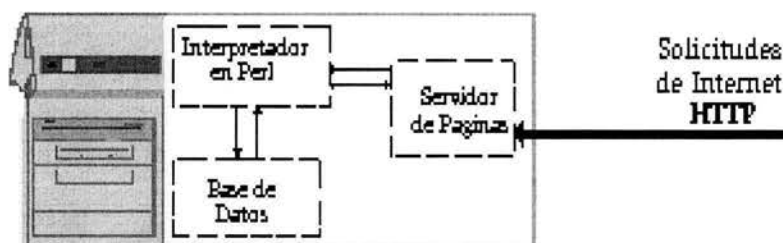


Figura 2.13

Cuando el servidor de páginas (Apache) recibe la requisición para "x" página éste reconoce cuando debe enviar un documento estático (HTML) o ejecutar algún tipo de aplicación, en el diagrama se puede observar que la solicitud de "x" pagina invoca (llama) un programa en Perl y este a su vez solicita información a una base de datos, por lo tanto para llevar a cabo esta operación debieron iniciarse 2 *procesos* nuevos, quizás esto no sea de gran importancia para un sitio de 100 visitas diarias, pero ¿que sucedería con uno de 2 visitas por segundo?

Si no se tienen los suficientes recursos en cuanto a memoria y procesadores se refiere, seguramente caerá el servidor de páginas o bien se queme el "Host" (computadora física) por la demanda excesiva. Apache tiene tanto tiempo de desarrollo que han sido desarrolladas diferentes soluciones para evitar estas ineficiencias, algunas:

- Es capaz de utilizar otros interpretadores y lenguajes como "Tcl", "PhP" y "Python".
- Puede conectarse directamente a una Base de Datos.

Cabe mencionar que muchos sitios de alto tráfico aún permanecen bajo este tipo de arquitectura, en ocasiones si se tienen los recursos suficientes continua siendo costeable esta metodología a migrar a otro tipo de desarrollo, sin embargo, siempre es conveniente conocer otras alternativas.

AOL Server.

AOLServer al igual que Apache es un producto open-source pero a diferencia de este último, fue diseñado conociendo varias deficiencias que existían en el modelo inicial utilizado por Apache.

AOLServer desde sus versiones iniciales fue desarrollado con "Threading" en mente, esto es, compartir la memoria del Proceso general en varios sub-procesos o "Threads".

Además de "Threading" AOLserver integra un Interpretador en su estructura interna evitando generar un proceso nuevo por *aplicación de servidor* y mantiene grupos de conexiones *latentes* ("pools") hacia bases de datos también para evitar generar procesos nuevos. Figura 2.14

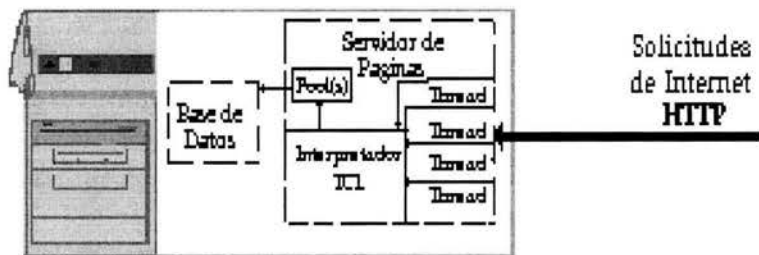


Figura 2.14

Java Application Servers.

"Java Application Servers" hoy en día ya denominados "Application Servers" ofrecen una manera de Integrar y ofrecer las funcionalidades requeridas por la gran mayoría de sistemas empresariales, una de las razones por las cuales el mercado ha sido inundado con estos "Application Servers" es que están diseñados alrededor de J2EE, que es *solo un grupo de especificaciones* definidas por Sun.

Estos "Application Servers" comúnmente llamados Middleware se encuentran compuestos como se muestra en la figura 2.15.

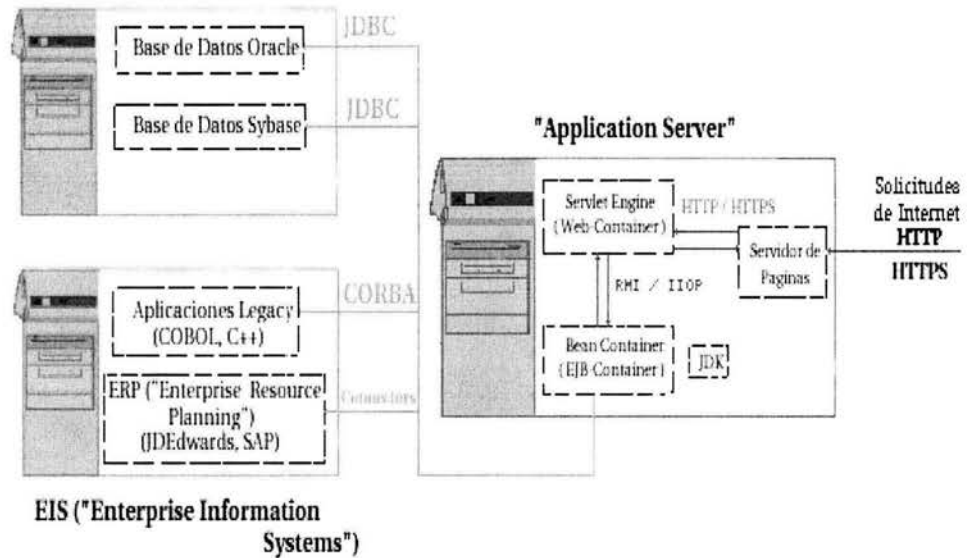


Figura 2.15

Como su denominación lo implica ("Middleware") se encuentran en la parte media de una arquitectura de sistema, su flexibilidad reside en la posibilidad de acceder información en sistemas empresariales (EIS) como SAP, JdEdwards, base de datos o alguna aplicación escrita en COBOL u otro lenguaje.

IIS (Information Server).

Las empresas buscan tecnologías de Web basadas en estándares para, de una manera más eficiente, intercambiar información con clientes, socios y empleados de todo el mundo. Además, intentan implementar soluciones que aprovechen al máximo las inversiones existentes en conocimientos, formación y tecnología. Microsoft Internet Information Server (IIS) está diseñado para cubrir estas necesidades a un amplio rango de usuarios, desde grupos de trabajo y departamentos de una Intranet corporativa hasta proveedores de servicios Internet que alojan sitios Web que reciben millones de visitas diarias. IIS 4.0 revoluciona las capacidades de Web del sistema operativo Microsoft Windows NT Server versión 4.0, proporcionando la forma más fácil de compartir información, crear y distribuir aplicaciones de negocio, y alojar y administrar sitios.

Ventajas:

- Las características innovadoras de publicación de Web, herramientas personalizables y tecnologías de nuevos asistentes, hacen de Windows NT Server con IIS 4.0 la forma más fácil de publicar y compartir información, de forma segura, a través de intranets corporativas y a través de Internet.
- Las aplicaciones Web pueden simplificar procesos como la creación de informes de gastos, administración de beneficios, compras, administración de inventario,

marketing y ventas. IIS 4.0 proporciona todo lo necesario para distribuir aplicaciones Web fiables y escalables en Windows NT Server.

- Las herramientas personalizables de administración, opciones flexibles de administración y herramientas de análisis hacen de Windows NT Server con IIS 4.0 el servidor de Web más fácil de administrar.

Características básicas:

- Protección contra errores graves. La protección contra errores graves permite a los clientes ejecutar varias aplicaciones en el servidor Web con fiabilidad. Si una aplicación produce un error, el servidor Web y las demás aplicaciones continuarán ejecutándose y la aplicación que falló se reinicia con la siguiente petición.
- Active Server Pages con transacciones. IIS 4.0 presenta Transactional Active Server Pages, que permiten a las aplicaciones con secuencias de comandos y componentes realizar múltiples acciones, aceptando todas las acciones simultáneamente, o sin aceptar ninguna. Esto es decisivo para las aplicaciones de bases de datos.
- Depuración de secuencias de comandos. IIS 4.0 ofrece una solución integrada para depurar secuencias de comandos y aplicaciones de Active Server Pages, lo que facilita la programación de aplicaciones ASP.
- Cola de mensajes integrada. Microsoft Message Queue Server (MSMQ) ofrece una forma fácil para que las aplicaciones envíen y reciban mensajes a través de una red con fiabilidad, incluso en el caso en que parte de la aplicación, o la red, no estén disponibles.
- Compatibilidad con Java. IIS 4.0 incluye un conjunto de clases de Java para crear componentes del lado del servidor y Java 1.1 Virtual Machine de Microsoft. Si utiliza Active Server Pages y Microsoft Transaction Server, puede crear aplicaciones Java y ejecutarlas fácilmente en entornos de servidor.
- Control total del contenido. Ahora los administradores pueden configurar el servidor Web en diferentes niveles: por servidor, por sitio, por directorio virtual, por subdirectorio o por archivo, lo que ofrece flexibilidad total en la mayoría de las configuraciones de servidor.
- Administración de contenido y análisis de sitios. IIS 4.0 ofrece la capacidad de crear mensajes de error personalizados y restricciones PIC. Esta versión también incluye Microsoft Site Server Express, que le permite observar patrones de uso de su sitio y visualizar la estructura de su sitio a través de un archivo de registro básico y de análisis de contenido.
- Múltiples sitios Web. IIS 4.0 admite múltiples sitios Web en una única dirección IP, operadores de sitios Web, ajuste de ancho de banda por cada sitio Web y administración mejorada y personalizable, que permite a las organizaciones alojar múltiples sitios de intranets de departamentos o múltiples sitios públicos de Internet.
- Estándares de Internet. IIS 4.0 incorpora los estándares existentes y los más recientes de la industria, incluidos: HTTP 1.1 para un mayor rendimiento de

Internet, administrando y emitiendo certificados digitales X.509 y enviando mensajes a servidores estándar de noticias NNTP y de correo SMTP.

- Compatibilidad con administración automatizada. Con los nuevos objetos de administración de IIS puede escribir secuencias de comandos y ejecutarlas desde la línea de comandos, automatizando diferentes tareas de administración. Esta versión incluye un conjunto de secuencias de comandos de ejemplo que están listas para su utilización o personalización, de manera que satisfagan sus necesidades de administración.

Características adicionales:

- Servidor Web más fácil de instalar y configurar. Si utiliza el asistente para descarga, puede seleccionar los componentes específicos que desea instalar. Además, la nueva capacidad de instalación desatendida permite instalar IIS en varios servidores sin tener que supervisar las instalaciones.
- Herramientas flexibles de administración. Las herramientas flexibles de administración del servidor Web IIS 4.0 incluyen una consola integrada de administración basada en Windows, administración mejorada basada en el explorador, secuencias de comandos ejecutables desde la línea de comandos, así como interfaces programables para crear herramientas de administración personalizadas.
- Seguridad. IIS 4.0 incluye un servidor de certificados que está estrechamente integrado con el modelo de seguridad de Windows NT Server, lo que permite a las organizaciones emitir y administrar certificados digitales X.509 estándar de Internet.
- Motor de búsqueda. IIS 4.0 incluye capacidades de búsqueda integradas, que permiten a los usuarios crear formularios de búsqueda personalizados con Active Server Pages, objetos de datos ActiveX y consultas SQL para buscar información en el servidor Web.

Web Server Para Windows.

KF Web Server 1.0.5.

Este es un servidor Web de pequeño tamaño, poco exigente en cuanto a recursos y muy sencillo de administrar. Soporta un número ilimitado de sitios Web, múltiples nombres de dominio y múltiples puertos, y está preparado para hacer uso de la última versión estándar de HTML, así como CGI, PHP, etc.

El servidor realiza logs de todos los eventos que tienen lugar, coloca un icono de monitoreo en la bandeja de sistema y se administra vía Web a través de un navegador.

KF Web Server cuenta asimismo con un completo sistema de seguridad y es totalmente personalizable.

WebSite Pro 3.1.13.0.

WebSite Pro es uno de los servidores Web más flexibles que existen hoy en día.

Soporta diversas tecnologías, como CGI, Perl, Pitón, ISAPI, WSAPI, SSI, Java, ColdFusion y ASP, proporcionando una amplia variedad de opciones a los desarrolladores Web respecto a qué plataforma elegir.

El servidor se controla desde una interfaz de uso sencillo, con un diseño dividido en varias pestañas que te dan acceso a las diferentes opciones de configuración y control.

Ironwall Servidor Web 5.08.

Ironwall HTTP Server es un interesante servidor de páginas web estándar que cumple con la norma RFC2616 (en las funciones esenciales) que sorprende por su sencillez y potencia.

Como datos interesantes destacan las limitaciones en conexiones simultáneas totales (512) y por equipo (4). El número de permisos es de un máximo de 10000 (como máximo simultáneamente, teniendo en cuenta que una máquina consume un permiso, sin que afecte el número de conexiones reales establecidas y contando con la vigencia del permiso).

Otras características remarcables son las descargas parciales, limitación de ancho de banda total y por equipo, administración de conexiones, lista de accesos, permisos y seguridad, envíos, etc.

Mejoras en la nueva versión:

Reajustada rutina de acceso al archivo "index.html" si el *Web browsing* está activo, soporte para múltiples idiomas (este paquete sólo incluye español), añadida la posibilidad

de editar el archivo inicial por defecto, el sistema de archivo *log* se acumula y no se borra al reiniciar el servidor, añadida mejora en el menú "Conexiones", el botón derecho permite mandar al usuario directamente a la lista de Klined y retoque en el menú de seguridad, se han quitado ciertos errores que afectaban a elementos de los menús, se ha revisado el "motor" Ironwall para buscar errores, etc.

Abyss Web Server 1.0.3.

Abyss Web Server es un servidor Web personal de pequeño tamaño pero con potentes funciones, válido tanto para sistemas basados en Windows como Linux.

Soporta HTTP/1.1, scripts CGI, alias de directorio, descargas parciales y archivos index. Además, añadir soporte para PHP y Perl es cuestión de minutos.

Incluye una interfaz gráfica que facilita enormemente la configuración y mantenimiento del servidor incluso remotamente, ya que funciona como una página Web.

Sambar Server 5.1 Beta 5.

Sambar Server es un servidor http. Entre otras muchas características, Sambar Server incluye:

- Completo motor de búsqueda.
- Generación de páginas en HTML dinámico.
- Servicios de HTTPS, SMTP, POP3, IMAP4 y proxy.
- Servidor FTP incorporado.
- Soporte para CGI.

Mejoras en la nueva versión:

Se han solucionado algunos bugs importantes

Limitaciones de la versión shareware:

No tienes acceso a todas las posibilidades de la aplicación

MyWebServer 1.0.2 Release Build 03.27.02.

MyWebServer es una aplicación de servidor Web de reducido tamaño pero potente funcionalidad.

Accede a la PC desde cualquier sitio mediante el uso de un navegador web estándar, como si fuera una página Web cualquiera.

MyWebServer ofrece un sistema de configuración paso a paso que será muy fácil de completar, e incluye multitud de opciones para personalizarlo. Se puede crear directorios

virtuales, establecer contraseñas de acceso a ciertos archivos o carpetas, utilizar cgi-bin, etc.

Una característica importante de este programa es que permite ejecutarlo en modo "dual", es decir, utilizando un doble servidor: uno puede dedicarlo a uso público mientras que el otro puede ser de acceso restringido sólo a ciertos usuarios.

Xitami Multithreaded Webserver 2.5b4.

Uno de los servidores Web más rápidos del mercado. Utiliza "multithreading" (múltiples hilos de ejecución) para soportar cientos de conexiones al mismo tiempo sin apenas esfuerzo.

Soporta las opciones de HTTP/1.0 estándar: CGI, registro de sucesos ("logging") NSCA/HTTP, "keep-alive", protección de páginas con contraseña, tipos MIME definidos por el usuario.

Ideal para Intranets. Ocupa muy poco en disco y en memoria. El código fuente del "kernel" está disponible en la Web del autor de forma totalmente gratuita

Viking Web Server 1.06.

Soporta todo lo que pueden necesitar usuarios privados o compañías para montar cualquier tipo de servidor.

Por ejemplo en HTTP/1.1 soporta: funcionalidad de caché proxy, SSL-tunneling, hosting de múltiples dominios/IP's, "mirroring" de Web's automático, SSI, CGI, Perl (a través un programa externo).

También actúa como servidor FTP, SMTP (con listas de distribución, redirector de e-mail), NNTP, servidor proxy (para conectar una LAN entera a Internet a través de una única conexión), Finger, y Firewall (cortafuegos). Se puede configurar remotamente.

Limitaciones de la versión shareware:

Totalmente funcional durante 90 días, luego deja de funcionar.¹

¹ www.shta.net/foros/index.php?s=ac40557e0f9afb9460f20401ca67d8a1&showtopic=17&st=0&#entry186

<http://www.ucm.es/info/Psyap/Prieto/alum9798/intranet01/server.htm>

www.osmosislatina.com/aplicaciones/servidor_web.htm

www.microsoft.com/catalog/display.asp?subid=44&site=534&pg=1

2.3.3 Arquitecturas de Hardware

El chip más importante de cualquier placa madre es el microprocesador o simplemente procesador. Sin él, un ordenador no podría funcionar. A menudo a este componente se le denomina CPU (Central Processing Unit, Unidad de procesamiento central), que describe a la perfección su papel dentro del sistema. El procesador es realmente el elemento central del proceso de tratamiento de datos.

La CPU gestiona cada paso en el proceso de los datos. Actúa como el conductor y el supervisor de los componentes de hardware del sistema. Asimismo, esta unidad, directa o indirectamente, con todos los demás componentes de la placa principal. Por lo tanto, muchos grupos de componentes reciben órdenes y son activados de forma directa por la CPU .

INTEL.

Fue fundado en 1968 con el propósito de crear soluciones para disminuir los costos de la tecnología de los semiconductores. Su primer chip fue el 1103, posteriormente en 1970 se lanzó el 4001. En el año de 1972 lanzó el 8008 con un bus de 8 bits y una cantidad de 3500 transistores. Con la aparición de la PC de IBM, entraron al mercado los microprocesadores 8086 y 8088, el 8086 contaba con un bus de 16 bits y velocidades de reloj de 5,8 y 10 Mhz, con 29000 transistores. En el año de 1982 hacen su aparición los primeros 80286, con un bus de 16 bits y 134000 transistores y velocidades de 8,10 y 12 Mhz. En 1985 apareció el 80386DX con arquitectura de 32 bits, unos 275000 transistores y una velocidad máxima de 33Mhz. El 80486DX aparece con mejoras tales como el co-procesador matemático, por primera vez se sobrepasaba el millón de transistores. En el año de 1992 aparecen la quinta generación de procesadores Intel denominada Pentium, con velocidades de 60 y 66 Mhz, 3 100 000 transistores, y bus de datos de 64 bits para una comunicación más rápida con la memoria externa. Para 1994 se introdujo la segunda generación de procesadores Pentium con velocidades de 90 y 100Mhz. La tercera generación apareció en 1997 con la tecnología MMX (Multimedia extensión) a velocidades de 166, 200 y 233 Mhz. A este microprocesador le sucedieron el Intel Pentium pro, el Pentium II, III y el Intel Pentium 4 el cual corre a velocidades de 1.4 y 1.5 Ghz, esta diseñado para aplicaciones de Internet y multimedia de alto nivel, como edición de video, codificación y carga de un video a Internet, codificación de archivos MP3 y aplicaciones tridimensionales y visualización.

AMD.

Fue fundada en 1969, inicialmente se dedicaba a rediseñar y mejorar productos de otras empresas ofreciendo más garantía en su producto final. De este modo AMD copiaba los diseños de Intel y los optimizaba, proporcionaba mayor Mhz con un precio inferior. Así es

como surgieron el Am386 y el Am486 Con la aparición de los Intel 486DX, AMD lanzó su propio procesador Am486 DX2.

La llegada de los microprocesadores Pentium de Intel provocó un rompimiento dentro de los esquemas de AMD, uno de ellos es que el Pentium estaba completamente patentado por lo que no era posible copiarlo. Es así como aparece el K5 que ofrecía ciertas mejoras con respecto al Pentium de Intel. El k6 surgió como mejora del NexGen Nx686 cuya característica principal era ser bastante más avanzado que el Pentium clásico y el MMX. La siguiente mejora se introdujo con el K6-2 , K6-3 competidores directos de el Intel Celeron. En 1999 se introdujo el Athlon, el cual obtuvo una gran aceptación, este microprocesador competía directamente con el Pentium III el cual llegó a superar en varios aspectos, entre ellos el ser el primero en correr a más de 1Ghz. El AMD duron se introdujo como una versión económica del Athlon con las mismas prestaciones pero con caché de menor capacidad.

La tabla 2.11 y 2.12 proporciona un comparativo de tecnología de microprocesadores Intel contra AMD.

PROCESADORES AMD	INTRO DU- CIDO	VELOCIDAD DE RELOJ (MHZ)	VELOCIDAD DE RELOJ (GHZ)	NIVEL 1 CACHE DE DATOS INSTRUCCIÓN/ DATOS	NIVEL 2 DE CACHE	VELOCIDAD DEL BUS DE SISTEMA MHZ	TECNOLOGÍA DE PROCESO MICRAS	TRANSISTORES INCLUYENDO CACHE EN MILLONES
Duron (Spitfire)	2000	600,650,700,750 800,850		64K/64K	64K	200	0.18	25
Duron (Morgan)	2001		1.0,1.1,1.2	64K/64K	64K	200	0.18	25.2
Athlon xp (Palomino)	2001		1.33,1.40, 1.47,1.53,160	64K/64K	256K	266	0.18	37.5
Athlon MP (Palomino)	2001		1.2,133,1.40, 1.53,1.60	64K/64K	256K	266	0.18	37.5
PROCESADORES INTEL	INTRO DU- CIDO	VELOCIDAD DE RELOJ (MHZ)	VELOCIDAD DE RELOJ (GHZ)	NIVEL 1 CACHE DE DATOS INSTRUCCIÓN/ DATOS	NIVEL 2 DE CACHE	VELOCIDAD DEL BUS DE SISTEMA MHZ	TECNOLOGÍA DE PROCESO MICRAS	TRANSISTORES INCLUYENDO CACHE EN MILLONES
Celeron (Tualantin)	2001		1.2	16K/16K	256K	100	0.13	44
Pentium III (Coppermine)	1999	533, 550, 600, 650, 667, 700, 733, 750, 800, 850, 866	1.0,1.13	16K/16K	256K	100,133	0.18	28
Pentium III (Tualantin)	2001		1.13,1.20, 1.33	16K/16K	256K	133	0.13	44
Pentium 4 (Willamete)	2000		1.3,1.4,1.5, 1.6,1.7,1.8, 1.9,2.0	12K/8K	256K	400	0.18	42
Pentium 4 (NorthWood)	2002		2.0,2.2	12K/8K	512K	400	0.13	55
Xeon (Foster)	2001		1.4,1.5,1.7, 2.0	12K/8K	256K	400	0.18	42
Xeon (Prestonia)	2002		2.2	12K/8K	512K	400	0.13	55

Tabla 2.11 Comparativo INTEL vs AMD

	Pentium III	Athlon
Velocidades Mínima Máxima	450 MHz 1,000 MHz	550 MHz 1,000 MHz
Desempeño	El SSE (Streaming SIMD Extensions), que permite procesar datos complejos para un mejor desempeño de gráficas, video y sonido	Permite ejecutar mas instrucciones x86 al mismo tiempo y cuenta con su sistema mejorado 3DNow! Para mejorar los gráficos, video y sonido
Cache L1 L2	32K Interno 256K, Externo 512 a 2 MB	128K 512K a 8 MB
Bus frontal	133 MHz	200 MHz expandible a 400 MHz
Arquitectura	Intel 820 Consiste en una serie de 3 hubs que enlaza el procesador a los demás componentes. Incluye un vía entre los controladores que proveen 266 MBps, el doble que el ancho de banda de los PCI.	AMD-750 Consiste en 2 controladores para conjugar los controladores del procesador, memoria, video y PCI – y el controlador bus de los periféricos para el disco duro, los puertos USB y los slots ISA.
AGP Video	El 820 soporta AGP 4X, que permita al controlador de gráficos acceder la memoria a velocidades de 1 GB por segundo	Soporta AGP 2X, así que el controlador de gráficos acceder a la memoria a 500 MB por segundo
RAM	Soporta RDRAM (Rambus Direct RAM), que corre a velocidades de hasta 400 MHz	Soporta el PC100 SDRAM a 100 MHz, no soporta RDRAM

Tabla 2.12 Características adicionales para el comparativo de Microprocesadores

Para comprender más estos datos, se complementa el análisis con otros resultados que son importantes para la correcta evaluación de los procesadores, estos son:

- El cache L2 del Athlon corre a un tercio de la velocidad del procesador, es decir a 333 MHz, mientras que el Pentium III corre a la velocidad máxima del procesador.
- El sistema SSE del Pentium III procesa el doble de datos (cuatro entradas de 32-bit por 2 del Athlon), que el sistema 3DNow! de Athlon.
- El voltaje del Pentium III es de 1.7 volts, mientras que el del Athlon es del 1.8 volts.
- El sistema Intel 820 soporta un tipo de memoria más veloz conocida como RDRAM, que permite velocidades hasta de 400 MHz. El sistema SDRAM opera sólo hasta los 100 MHz.

Servidores para Internet.

Internet es un conjunto de máquinas de diferentes manufacturas con arquitecturas internas también diferentes que se comunican una a otra a través del envío de datos.

Este propósito se logra enviando la información de manera electromagnética, por cable, satélite y otros. Figura 2.16

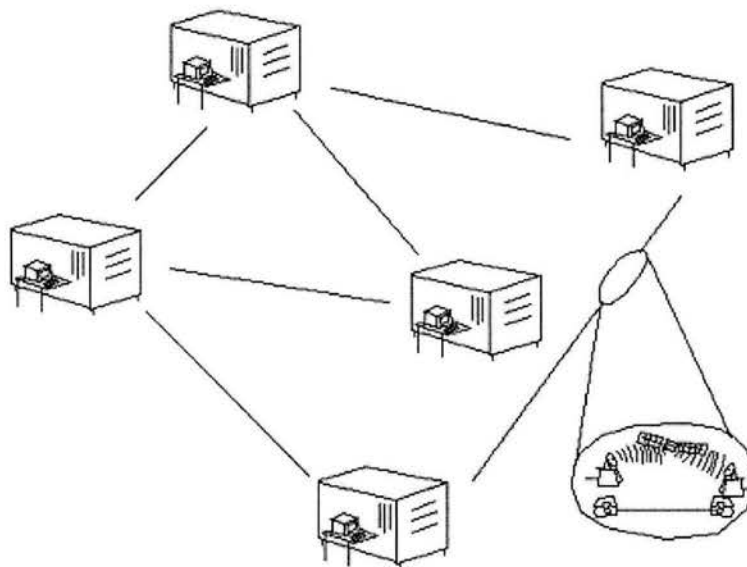


Figura 2.16 Conjunto de máquinas interconectadas en Internet

La base del Internet son los servidores, los servidores de Internet son máquinas más poderosas en el manejo de velocidad y espacio en comparación con los equipos personales.

De este modo para obtener una conexión de Internet es necesario conectarse a estos servidores.

SUN.

Los servidores SUN (Tablas 2.13 y 2.14) Fire V100, de SUN Microsystems, ofrecen una estupenda alternativa a las arquitecturas Intel/PC, con arquitectura UltraSparc, y con un rendimiento óptimo para muchas de las aplicaciones habituales de proyectos Internet. Además, al robusto sistema operativo Solaris 8, le acompañan aplicaciones de alta categoría como iPlanet WebServer y Chili!Soft ASP.

Características de algunos servidores comerciales de Sun

SUN Fire V100 (I)

Modelo:	SUN Fire V100
Procesador:	UltraSPARC® IIe 500-MHz
Memoria	256 MB RAM (max. 2GB)
Disco Duro:	40 GB IDE 7.200 r.p.m
S.O.	SUN Solaris 8

Tabla 2.13

SUN Fire V100 (III)

Modelo:	SUN Fire V100
Procesador:	UltraSPARC® IIe 500-MHz
Memoria	1 GB RAM (max. 2GB)
Disco Duro:	2x 40 GB IDE 7.200 r.p.m
S.O.	SUN Solaris 8

Tabla 2.14

IBM.

Hoy, los servidores basados en procesadores Intel pueden realizar muchas de las tareas que alguna vez estuvieron reservadas para servidores de tipo empresarial. De este modo IBM ofrece servidores para aplicaciones sencillas con pocos requerimientos de recursos hasta equipos poderosos que requieren grandes

accesos a disco (Tablas 2.15, 2.16, 2.17), como por ejemplo acceso a bases de datos.

Características de algunos servidores comerciales de IBM

<u>x300 PIII IDE (PIII Celeron 950 MHz / 256MB RAM / 20GB IDE)</u>	
Modelo:	IBM x300
Procesador:	PIII Celeron 950 MHz
Memoria	128 MB RAM (max. 1,5GB)
Disco Duro:	20 GB IDE 7.200 r.p.m.
S.O.	GNU/Linux

Tabla 2.15

Modelo:	IBM x305
Procesador:	P4 1.800 MHz (512 kb caché)
Memoria	256 MB RAM (max. 4 GB)
Disco Duro:	18 GB Ultra-160 SCSI 10.000 r.p.m.
S.O.	GNU/Linux ó FreeBSD (RedHat, Debian, SUSE, Mandrake...)
IPs	2

Tabla 2.16

Modelo:	IBM x345
Procesador:	Dual P4 XEON 2.0 GHz (512kb caché)
Memoria	1 GB RAM PC2100 ECC DDR (max. 4GB)
Disco Duro:	3x 36 GB Ultra-320 SCSI 10.000 r.p.m.
S.O.	GNU/Linux

Tabla 2.17

HEWLETT PACKARD.

La familia de servidores HP cubre el rango de desempeño más amplio del mercado, en los dos ambientes que dominan el universo de los sistemas operativos: Unix y Windows NT. La amplitud de la gama de servidores HP NetServers (procesadores Intel) y HP 9000 (procesadores PA-RISC de HP) permiten satisfacer la totalidad de los requerimiento de cómputo de la empresa.

La familia de Servidores Empresariales HP 9000 basados en la tecnología HP PA-RISC, es líder en la industria de servidores de negocios UNIX. Esta línea de

servidores soporta ambientes de cómputo que van desde pequeños grupos de trabajo hasta miles de usuarios en un centro de datos de administración centralizada.

La familia de HP NetServers, basados en procesadores Intel, ofrecen también niveles de disponibilidad, desempeño y administrabilidad superiores en el mundo de Windows NT.

Características de Algunas Servidores HP Comerciales (Tablas 2.18, 2.19).

HP SERVER rx2600

Modelo: IBM x305
Procesador: Intel Itanium
Memoria 12 Gb DDR
Disco Duro: 219 Gb
S.O. HP-UX, Windows, Linux

Tabla 2.18

HP SERVER rx5670

Modelo: IBM x305
Procesador: Intel Itanium
Memoria 48 Gb DDR
Disco Duro: 584 Gb
S.O. HP-UX, Windows, Linux

Tabla2.19

2.3.4 Manejadores de Bases de Datos

Sin duda alguna la infraestructura de Internet a dado cabida a nuevos tipos de aplicaciones y servicio a los usuarios desarrollados por empresas, organizaciones y gobiernos. Parte fundamental de estos servicios son la implementación de bases de datos accedidas a través de la World Wide Web. El alcance y el fácil acceso a ellas, así como la reducción de costos y la popularidad que ha cobrado la Web, son los principales atractivos que ofrece una aplicación de esta naturaleza.

La Web es un medio para localizar/enviar/recibir información de diversos tipos, aun con las bases de datos. En el ámbito competitivo, es esencial ver las ventajas que esta vía electrónica proporciona para presentar la información, reduciendo costos y el almacenamiento de la información, y aumentando la rapidez de difusión de la misma.

DB2 Versión 5.2.

- Plataformas.
DB2 Universal Database corre en modo nativo en casi todas las plataformas como Windows, NT, Sun Solaris, HP-UX, AIX, OS/400 y OS/2.
- Integridad.
Incluye características de integridad, asegurando la protección de datos; y de seguridad, permitiendo realizar respaldos en línea, sin que esto afecte la disponibilidad de acceso a los datos por parte de los usuarios.
- Múltiples usos.
Provee la capacidad de trabajar desde el procesamiento transaccional de misión crítica (OLTP), hasta análisis exhaustivo de los datos para el soporte a la toma de decisiones (OLAP).
- Trabajo en Web.
Permite generar aplicaciones en intranets y responder a las oportunidades de negocios en Internet, provee soporte a java.
- Facilidad de instalación y uso.
Cuenta con herramientas gráficas para facilitar el uso a usuarios, administradores y desarrolladores.
- Conectividad.
Permite acceder a los datos ya sea en mainframe o AS/400, Windows NT, Windows 95/98, OS/2 o cualquiera de los Unix soportados. Se puede

acceder de forma única y transparente a los datos residentes en ORACLE, Sybase, Informix, Microsoft SQL Server y otros.

Progress.

- Plataformas
Progress corre sobre todas las variantes principales de UNIX, incluyendo UNIX SVR4.2,SCO, AIX, HP-UX, ULTRIX Y Solaris, así como en los principales sistemas operativos propietarios de minicomputadoras, tales como VAX/VMS y OS/400. Progress también corre en los principales sistemas operativos de PC y LANs.
- Integridad.
Registro "Roll-back". Esta característica ofrece protección para la base de datos en caso de cualquier falla. La base de datos utiliza el registro "roll-back" durante la recuperación de caída para aplicar los cambios finalizados a la misma y deshacer todas las transacciones no finalizadas.
"Two-phase commit". Progress Enterprise RDBMS implementa un protocolo de dos fases como una característica automática para asegurar la integridad y consistencia de las transacciones que accesan múltiples bases de datos y múltiples sitios. Este protocolo asegura que una transacción puede ser deshecha si ocurre una falla en cualquier nodo de la red.
- Múltiples usos.
Progress Enterprise RDBMS está diseñado para grandes ambientes empresariales y el procesamiento de transacciones a través de las aplicaciones más demandantes de hoy en día, basadas en procesamiento de transacciones en línea (OLTP) utilizando SQL y Progress 4GL.
- Trabajo en web
Herramientas como Progress WebSpeed ofrece a los desarrolladores una solución perfecta y comprobada para construir de manera rápida y segura aplicaciones empresariales que pueden ser implementadas ya sea en Internet o en intranets y extranets corporativas.
- Facilidad de instalación y uso.
Sofisticadas capacidades autoajustables e interfaces gráficas simples para la administración del sistema, hacen de Progress Enterprise RDBMS un sistema más sencillo de instalar, afinar y administrar que otros productos.

- **Conectividad.**
Progress Enterprise RDBMS proporciona un acceso fácil y completo a datos almacenados en múltiples bases de datos a través de la red. Las aplicaciones pueden conectarse con 240 bases de datos simultáneamente.

Informix (informix dynamic server) (ids) 9.4.

- **Plataformas.**
IBM AIX, HP-UX, Sun Solaris, Linux sobre servidores compatibles con Intel, Linux para IBM, Relian UNIX, SGI IRIS, Compaq Tru64, Microsoft Windows 2000, XP, 2003-10-13.
- **Integridad.**
Presenta soporte de procesamiento de transacciones online con la seguridad de integridad de los datos.
- **Múltiples usos.**
Soporta requisitos de procesamiento de transacción online, complejos y rigurosos.
- **Trabajo en Web.**
Proporciona fiabilidad superior, atendiendo las necesidades de las exigentes prácticas actuales del e-business particularmente para aplicativos que requieran transacciones de alto desempeño.
- **Facilidad de instalación y uso.**
Las herramientas de administración de base de datos Informix proporciona una tecnología Web y Java, para una fácil administración de servidores IBM Informix. Esto elimina la necesidad de memorizar comandos en línea, instrucciones y opciones, simplifica tareas de configuración, respaldo y recuperación, monitoreo.
- **Conectividad.**
El Informix Enterprise Gateway Manager, permite interoperar de manera transparente con bases de datos que no son Informix por ejemplo Oracle y Sybase.

Oracle.

- **Plataformas.**
ORACLE es la base de datos más extensamente usada en el mundo. Su software funciona en más de 100 diferentes plataformas.

- **Integridad.**
Proporciona características de reinicio rápido ante caídas del sistema, garantiza que transacciones no guardadas en algún fallo, puedan ser “rolled back” al reinicio del sistema.
- **Múltiples usos.**
La tecnología de la base de datos Oracle, incluye desarrollos asociados y herramientas de administración, que permiten solventar las demandas de aplicaciones OLTP.
- **Trabajo en Web.**
Está diseñada para soportar las capacidades de Internet. Proporciona una amplia funcionalidad para soportar negocios que corren en el Web.
- **Facilidad de instalación y uso.**
Cuenta con un conjunto de herramientas gráficas para la administración de la base de datos. Provee la habilidad de crecer alrededor de plataformas y sistemas operativos, manejos de concurrencia permitiendo a la empresa crecer en el número de usuarios.

SQL SERVER.

- **Plataformas.**
Es la mejor base de datos para Windows NT Server. Posee los mejores registros de los benchmarks independientes, tanto en transacciones totales como en coste por transacción.
- **Integridad.**
Copias de seguridad online desatendidas garantizando la consistencia de datos para la más alta disponibilidad. Recuperación justo a tiempo para restaurar bases de datos en un intervalo de tiempo.
- **Múltiples usos**
Aporta una capacidad OLTP (online transaction processing) potente para gestionar las transacciones durante el día, y un rendimiento de soporte a decisiones fiable para actividades de cierre e informe. Además permite a los comercios enviar datos seleccionados automáticamente a los proveedores corporativos.

- Trabajo en Web.
Plataforma de desarrollo fácil y abierta; integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server. Diseñada para INTERNET, es el único gestor de bases de datos que contiene de forma integrada la posibilidad de generar contenido HTML de forma automática.
Mínimo costo de propiedad, la sencillez de la instalación, y la potencia de sus herramientas de gestión y el menor costo de la industria para entornos Internet.
Compatibilidad con Microsoft Internet Information Server y otros servidores de Web.

SQL Web Assistant, para el retorno automático de datos en formato HTML.

Facilidad de instalación y uso.

Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de base de datos.

MYSQL.

- Plataformas.
Está disponible para casi todas las plataformas hardware y sistemas operativo (incluidos *Windows NT/2000,98/95/ME*). Unix y OS/2 para las cuales es gratuita.
- Integridad.
No tiene integridad referencial, con lo que la base de datos puede llegar a tener datos inconsistentes. Realiza los bloqueos de escritura al nivel de tabla, castigando la concurrencia de inserciones en la base de datos.
- Trabajo en Web.
La velocidad de operaciones de lectura, permite a mysql ser una herramienta para Web; debido a que actualmente el trabajo en Web involucra grandes cantidades de operaciones de lectura.
- Facilidad de instalación y uso.
Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

.... Excelente Muy buena ... Buena .. Mala . Muy mala									
	N/D No Disponible								
	Ingeniería del servidor	Herramientas de administración	Diseño de la base de datos	Programa -bilidad	Seguri -dad	Inter- operabili -dad	Análisis de datos	Perfor -mance	total
DB2 Universal Database
Microsoft sql server
MySQL	N/D
Oracle
Sybase Adaptive Server Enterprise

Tabla 2.20 Tabla comparativa de Bases de Datos

2.3.5 Herramientas de Programación

Existe una multitud de lenguajes concebidos o no para Internet. Cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útiles para desarrollar distintas aplicaciones.

La versatilidad de un lenguaje está íntimamente relacionada con su complejidad. Un lenguaje complicado en su aprendizaje permite en general el realizar un espectro de tareas más amplio y más profundamente. Es por ello que a la hora de elegir el lenguaje a utilizar se tiene que saber claramente qué es lo que se quiere hacer y si el lenguaje en cuestión lo permite o no.

Podemos hablar de lenguajes de lado servidor como aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos

que pueden ser directamente "comprendidos " por el navegador y no necesitan un pretratamiento.

ASP.

Active Server Pages (ASP), es una tecnología propietaria de Microsoft. Se trata básicamente de un lenguaje de tratamiento de textos (scripts), basado en Basic, y que se denomina VBScript (Visual Basic Script). Se utiliza casi exclusivamente en los servidores Web de Microsoft (*Internet Information Server* y *Personal Web Server*). Los *scripts* ASP se ejecutan, por lo tanto, en el servidor y puede utilizarse conjuntamente con HTML y Javascript para realizar tareas interactivas y en tiempo real con el cliente.

Características.

Las principales características de la tecnología ASP y de sus páginas asociadas son:

Es gratuita. Viene implementada por defecto en los servidores web PWS (Windows 95-98-Millennium) e IIS (Windows NT-2000-XP). Además también es posible implementarla en otros servidores, aunque hace falta la instalación de las extensiones adecuadas.

Se puede mezclar código ASP con código HTML y código de script en cliente, sin tener que compilarlos por separado. Esto facilita la construcción de páginas ASP, siendo posible escribirlas en un simple editor de textos (Notepad, Word, etc.).

El código ASP se ejecuta en el servidor web, produciendo como salida HTML puro (junto con código de script en el cliente, si así se precisa). Por este motivo, su resultado es interpretado correctamente por cualquier navegador web estándar, aunque se utilice VBScript, compatible tan sólo con Internet Explorer y los navegadores en él basados (aunque es posible ejecutar código VBScript en Netscape Navigator usando el plug-in ScriptActive, de la casa NetCompass).

Permite usar componentes escritos en otros lenguajes de programación, como Visual Basic, Delphi, etc., que pueden ser llamados desde los propios guiones ASP.

Se ha portado a la plataforma Java, lo que hace que pueda ser utilizado en casi cualquier sistema operativo.

Permite acceder a las bases de datos de una forma rápida y sencilla, siendo éste uno de sus principales usos.

También permite el uso de componentes OLE, acceso a ficheros, a login del sistema, enviar correo, etc.

Tiene persistencia de variables en memoria entre distintas visualizaciones de página, permitiendo con ello manejar sesiones de usuario, con lo que se resuelve uno de los mayores problemas de la programación Web, derivado de la no orientación a conexión del protocolo HTTP.

PHP.

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir, es un lenguaje de programación con variables, sentencias condicionales, ciclos (bucles), funciones.

Características.

No es un lenguaje de marcado como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C.

Pero a diferencia de JavaScript que se ejecuta en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del browser, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. Se requiere para utilizarlo:

- Versión compilada de PHP
- Un servidor web (Apache, PWS, IIS, Etc.).
- Si desea manejar base de datos se recomienda Mysql Server

PERL.

Es un lenguaje interpretado que tiene varias utilidades, pero está principalmente orientado a la búsqueda, extracción y formateado de ficheros de tipo texto. También es muy usado para manejo y gestión de procesos (estado de procesos, conteo y extracción de parámetros característicos, etc.).

Algunas de las ventajas del uso del lenguaje PERL son las siguientes:

Construcción de pequeños programas que pueden ser usados como filtros para obtener información de ficheros, realizar búsquedas.

Características.

- Se puede utilizar en varios entornos, como puede ser Windows 95, OS/2, sin realizar cambios de código, siendo únicamente necesario la introducción del intérprete PERL correspondiente a cada sistema operativo.
- También es uno de los lenguajes más utilizados en la programación de CGI scripts, que son guiones o scripts que utilizan el interface CGI (Common Gateway Interface), para intercambio de información entre aplicaciones externas y servicios de información. Como ejemplo de ello tenemos los programas de búsqueda usados por el browser Netscape.
- El mantenimiento y depuración de un programa en PERL es mucho más sencillo que la de cualquier programa en C.

En general, los programas en Perl se ejecutan en el servidor, como todos los programas CGI, a diferencia de otros programas ejecutados por el cliente (generalmente un navegador como el Internet Explorer de Microsoft o el Navigator), como aquellos escritos en JavaScript o Java. Existen además extensiones al Apache (mod_perl) que permiten ejecutar directamente programas en Perl desde el servidor de HTTP. Mediante una serie de módulos adicionales, tales como el ODBC, Perl puede servir para acceder a bases de datos, desde las gratuitas como MySQL hasta el Microsoft SQL server.

JAVA.

Java es un lenguaje de programación orientado a objetos que ha sido diseñado para crear ambientes distribuidos y heterogéneos en Internet, aunque puede ser usado para elaborar aplicaciones que no necesariamente interactúen con Internet.

Características.

- Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos.
- Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.
- Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como *http* y *ftp*. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

- Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo.
- Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo.

JAVASCRIPT

JavaScript fue creado por la empresa Netscape, JavaScript no es un lenguaje de programación propiamente dicho. Es un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto.

Un programa en JavaScript se integra en una página Web (entre el código HTML) y es el navegador el que lo interpreta (ejecuta). Es decir el JavaScript es un lenguaje interpretado, no compilado (no se genera ningún tipo de fichero objeto o exe).

Características.

- Para programar en Javascript solo se necesita un editor de texto y un navegador para ejecutarlo.
- Es moderno (tiene pocos años).
- Es sencillo (Java, es bastante más complejo).
- Es útil (el desarrollo de Internet, se prevé muy rápido en los próximos años).
- Es potente: permite la moderna POO (programación orientada a objetos).
- Es barato: sólo necesitamos un editor de textos (el "Bloc de Notas" está incluido en el Windows) y un navegador (es gratuito, ya sea el "Internet Explorer" o el "Netscape").
- Es visual: permite la moderna "programación visual" (ventanas, botones, colores, formularios, etc.).

JavaScript proporciona los medios para:

- Controlar las ventanas del navegador y el contenido que muestran.
- Programar páginas dinámicas simples.
- Evitar depender del servidor Web para cálculos sencillos.
- Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet.
- Simular el comportamiento de las macros CGI cuando no es posible usarlas.
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.
- Comunicarse con el usuario mediante diversos métodos.

HTML.

HTML es una implementación del standard SGML (Standard Generalized Markup Language), estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones.

Metalinguaje para definir lenguajes de diseño descriptivos; proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones.

El HTML se creó en un principio con objetivos divulgativos. No se pensó que la Web llegara a ser un área de ocio con carácter multimedia, de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación se han ido incorporando modificaciones con el tiempo, estos son los estándares del HTML.

Características.

- Permite crear lenguajes de codificación descriptivos.
- Define una estructura de documentos jerárquica, con elementos y componentes interconectados.
- Proporciona una especificación formal completa del documento.
- No tiene un conjunto implícito de convenciones de señalización.
- Soporta, por tanto, un conjunto flexible de juegos de etiquetas.
- Los documentos generados por él son legibles.

XML.

XML es un metalinguaje que deriva de SGML- sólo se transforma en un lenguaje cuando está especificado (como HTML o como innumerables e inabarcables especificaciones que ya existen) – que utiliza la marca para definir una estructura contenedora de información.

Características.

- XML permite crear una estructura de marca con absoluta libertad para estructurar información.
- La flexibilidad y la potencia de esa estructura permite definir información con una riqueza enorme (Objetos completos, imágenes, procesos complejos, etc.).
- Esa información se expresa en texto puro, entendible para cualquier procesador en cualquier lugar.

- La estructura se puede definir y hacer estricta, con lo que se tiene un lenguaje compresible en cualquier entorno tal como HTML.
- XML es un estándar internacional, Inter. Empresas, multiplataforma y multientorno.
- La particularidad más importante del XML es que no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea a su antojo, dependiendo del contenido del documento.

	Licencias	Plataformas	Servidor de web	Facilidad de uso	Ejecución	Conexión a la base de datos
PHP	Open Source	Unix, Linux, Windows	Apache	Compleja	Del lado del servidor	Facilidad de conexión, con una sintaxis simple, cada base de datos tiene su propio conjunto de instrucciones.
ASP	Propietario	Windows, Unix	IIS	Fácil	Del lado del servidor	Facilidad de conexión, las funciones son las mismas independientemente de la base de datos.
PERL	Open Source	Unix, Linux, Windows	Apache	Compleja	Del lado del servidor	Facilidad de conexión, las funciones son las mismas independientemente de la base de datos.

Tabla 2.21 Tabla comparativa de Lenguajes de Programación

CAPÍTULO 3. ANÁLISIS DEL SISTEMA

Buscando una adecuada gestión de proyectos, así como una óptima administración de los recursos y las tareas que se tienen en los proyectos, hemos realizado un levantamiento de información para obtener los requerimientos de las áreas tanto administrativas como de desarrollo para conocer las necesidades de cada nivel participante. Los requerimientos se han desglosado en la siguiente sección en donde se destacan los puntos más importantes que deben estar contenidos en nuestro desarrollo.

3.1 Requerimientos

Para obtener una adecuada administración de los proyectos, las actividades que se van realizando y los recursos humanos que le han sido asignados, debemos contar con una herramienta de software que nos permita obtener los siguientes requerimientos.

Poder registrar todos los proyectos que se van a desarrollar o bien que ya se encuentran en desarrollo considerando su fecha de inicio y término, lo cual nos permitirá estimar para no sobrepasar los límites en recursos humanos con los que cuenta la empresa de desarrollo de software, también nos permitirá controlar el no excederse del momento en que el desarrollo debe ser entregado.

Cada proyecto estará dividido en varias tareas las cuales deben estar definidas de forma clara y perfectamente delimitadas en tiempo, para lo cual utilizaremos fecha de inicio y fecha de finalización para cada una de las actividades. Cada actividad deberá estar considerada como una parte del total del proyecto de desarrollo. No necesariamente las tareas deberán de estar de forma secuencial, es decir, una tarea no inicia al término de otra pues se pueden tener más de una tarea en ejecución al mismo tiempo.

Una vez definidas las tareas que se van a desarrollar en el proyecto, el siguiente paso será asignar los recursos a cada una de ellas considerando las capacidades de cada persona así como los roles que desempeñen dentro de los proyectos. Existirán tareas propias del análisis del sistema en donde serán asignadas personas expertas en esta fase. Esto dará como resultado y dependiendo de la dimensión del desarrollo así como la fecha que se ha comprometido de entrega a asignar un número variable de recursos humanos.

Teniendo el proyecto completo en tareas y recursos humanos asignados nos interesa obtener la información tanto exclusiva como general de la estructura final que ha tomado el proyecto para lo cual se deben generar los reportes que nos especifiquen qué personas

están asignadas a cada tarea y proyecto, al igual que conocer si una persona está involucrada en más de una actividad.

Para conocer el tiempo que cada persona está dedicando al desarrollo de proyectos resulta de suma importancia registrar la cantidad de horas o porcentaje de jornada laboral que se dedica a cada tarea, para lo cual se debe tener el registro de hora de inicio de labores y hora de término de la misma. Al realizar este registro podremos obtener el total de horas laborales de cada persona. En este registro de entrada y salida o inicio y fin de una jornada laboral será necesario que se especifique el porcentaje del día que la persona ha dedicado a cada tarea asignada, lo cual nos permitirá dar un seguimiento a los avances en cada tarea.

En la actualidad los desarrollos de software se realizan en las propias instalaciones de los clientes o bien en lugares llamados fábricas de software, en donde se hace un tipo de maquila de desarrollo de acuerdo a las especificaciones previamente definidas en el análisis y diseño del sistema; esto nos lleva a tener a nuestro personal distribuido, impidiendo registrar su inicio y fin de jornada laboral en un único lugar físico pues esto provocaría que el personal se traslade a un mismo lugar por lo que se requiere de un sistema que nos permita registrar ambas horas, la de inicio y fin de jornada laboral desde puntos remotos y a un bajo costo.

Con el registro diario de cada persona tendremos la oportunidad y facilidad de conocer en línea y al instante el personal que se encuentre actualmente trabajando y quienes aún no inician su jornada laboral. También nos debe permitir obtener resultados de tiempo y recursos humanos asignados hasta este momento a un proyecto, de esta forma podremos tener la información para la reasignación de tareas a cada persona dependiendo del rol en el cual se encuentre trabajando.

Uno de los aspectos más importantes es la seguridad del sistema la cual deberá estar basada en la utilización de usuarios y contraseñas, lo que nos permitirá que solo personal autorizado ingrese al sistema. El nivel de seguridad se encontrará en cada pantalla que se desee abrir.

3.2 Propuesta de Solución

Atendiendo a los requerimientos anteriores, se propone la creación de un sistema de software que permita tener el control en tiempo, actividades y recursos humanos asignados a todos los proyectos que se van a desarrollar en la empresa. Este software se propone bajo un esquema tipo Web el cual nos permitirá su acceso desde cualquier punto en donde se tenga una conexión a la red mundial de Internet.

La interfaz gráfica, Front - End será considerando sólo la información básica y necesaria, sin una navegación complicada en el manejo de la misma.

Toda la información estará almacenada en una base de datos robusta que soporte un gran volumen de información y que pueda controlar ella misma la integridad referencial.

3.2.1 Justificación

La finalidad de desarrollar el sistema en una plataforma Web es acercarlo a todo el personal que labora fuera de las instalaciones administrativas, permitiendo con esto que ninguna persona escape del control de sus actividades diarias.

Así se podrá ver de forma inmediata y completa todas las tareas que forman el proyecto de desarrollo del sistema en el que se este trabajando. También permitirá ver los recursos humanos asignados a las diferentes tareas y con esto conocer si se están aplicando bien los recursos humanos en la atención de la totalidad del sistema en desarrollo.

Debido al acceso que cualquier persona puede tener al sistema a través de páginas de Internet se hace necesario agregar accesos controlados por medio de un Usuario y una Contraseña, éstas serán controladas por dos mecanismos; el primero será la base de datos en donde se tendrá esta información y la segunda será controlada por cada página a la que se tenga acceso al sistema.

3.3 Modelado de Datos

El modelado de datos hace uso de diagramas de entidad-relación en donde se describen e identifican los datos y sus relaciones. Adicionalmente nos permite conocer los detalles del almacenamiento de datos y sus relaciones con los procesos dentro del modelo del flujo de datos.

A través de una representación gráfica vamos a realizar un análisis del comportamiento de la información que manejará nuestro sistema.

3.3.1 Diagrama Entidad - Relación

El diagrama de Entidad – Relación nos mostrará las relaciones entre las diferentes entidades dentro del Sistema de Administración de Proyectos, Actividades y Recursos Humanos Asignados.

Diagrama Entidad - Relación Operación

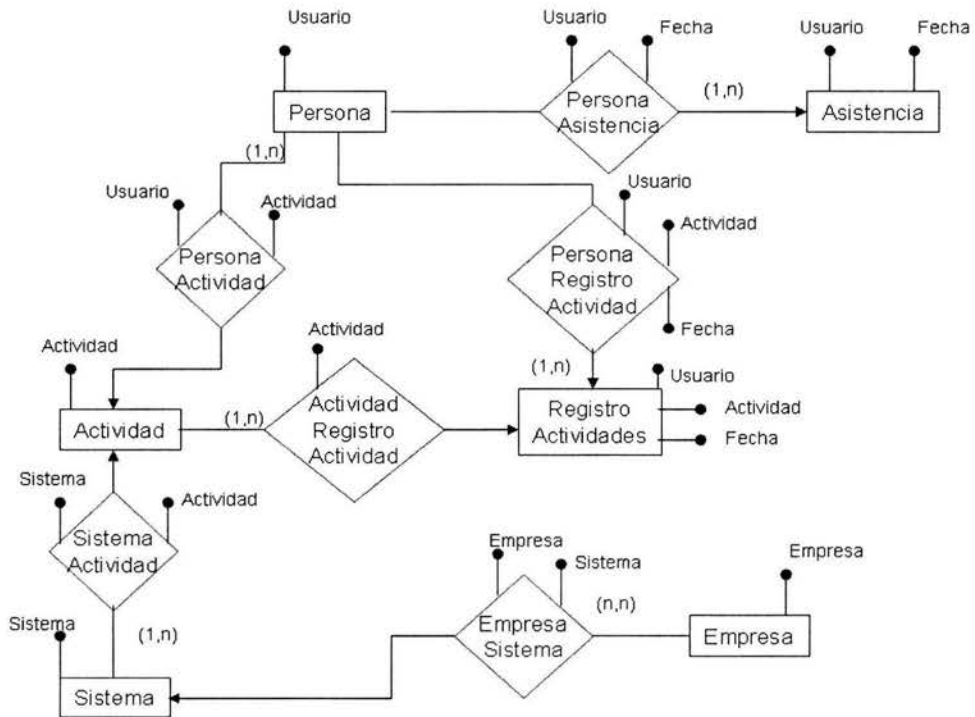


Figura 3.1

- La entidad Proyecto

Entidad para mantener la información de los proyectos que se estén desarrollando. Los proyectos tendrán solo un campo de llave principal la cual lo identifica como un proyecto único.

- La entidad Pantalla

Almacenará los nombres de cada pantalla que tendrá el sistema. Cada pantalla estará asociada a un sistema con lo que aseguramos que pueden existir dos pantallas con el mismo nombre pero asociadas a sistemas diferentes.

- La entidad Persona

La entidad Persona nos va a guardar todo lo relacionado a los accesos de las personas al sistema. Utilizamos un campo de llave principal el cual es la clave de la persona, lo que nos permite que no existan más de un usuario con la misma clave.

- La entidad Asistencia

Aquí llevamos el registro de horario de inicio y terminación de las actividades de cada persona que está participando en cada proyecto.

- La relación Persona – Asistencia

Relación que nos sirve para conocer a quien pertenece cada asistencia, por lo que queda implícita en la entidad de asistencia.

- La entidad Actividad

Es una entidad que opera como catálogo en donde se tendrá todas las actividades correspondientes a cada proyecto.

- La entidad Registro - Actividades

Todas las actividades asignadas a cada persona serán registradas con porcentajes de día dedicados en esta entidad. Aquí es donde estará la mayor parte de información que lleve el control de tiempos dedicados a cada actividad.

- La relación Actividad – Registro - Actividades

Solo es una relación que quedará dentro de la entidad de registro de actividades.

- La entidad Proyecto (sistema)

Contiene la información de los proyectos que se tienen en desarrollo.

- La relación Proyecto - Actividad

Aquí estarán almacenadas todas las actividades asociadas a todos los proyectos en donde se incluirán fecha de inicio y término para tener contemplados avances reales a cada una de las actividades.

- La relación Persona – Registro - Actividad

Estará dentro de la entidad registro de actividades en donde cada registro de actividad llevará la persona a quien se le esta registrando dicha actividad.

3.3.2 Diagrama de Flujo de Datos

Los diagramas de flujo de datos se usan para representar el flujo de la información. En su forma más simple constan de los siguientes elementos: entradas, salidas y proceso. A las entradas y salidas se les denomina como unidades externas. Con más detalle se pueden figurar los elementos que a continuación se listan: unidad externa, proceso, elementos de datos (este nos muestra la dirección del flujo de información) y el almacén de datos.

La unidad externa, puede representar tanto entradas como salidas y se representa por un rectángulo.

El proceso representa una transformación de los datos y se representa con un círculo.

Los elementos de datos se representan con una flecha que indica la dirección del flujo de información, estas flechas deben etiquetarse.

El almacén de datos, es un depósito de datos que se almacenan para ser usados por uno o más procesos; se representan mediante dos líneas paralelas etiquetadas.

3.3.2.1 Diagrama de procesos.

En el nivel cero, el diagrama de flujo de datos sólo brinda una visión muy genérica del sistema. Será en diagramas posteriores cuando se pueda comprender el funcionamiento del sistema.

En este nivel, el diagrama suele conocerse como diagrama de contexto figura 3.2 y sólo da idea de los flujos de datos de entrada y de salida del software.



Figura 3.2 Nivel 0

En el nivel 1 del diagrama de datos se obtiene una visión más específica del sistema como se muestra en la figura 3.3.

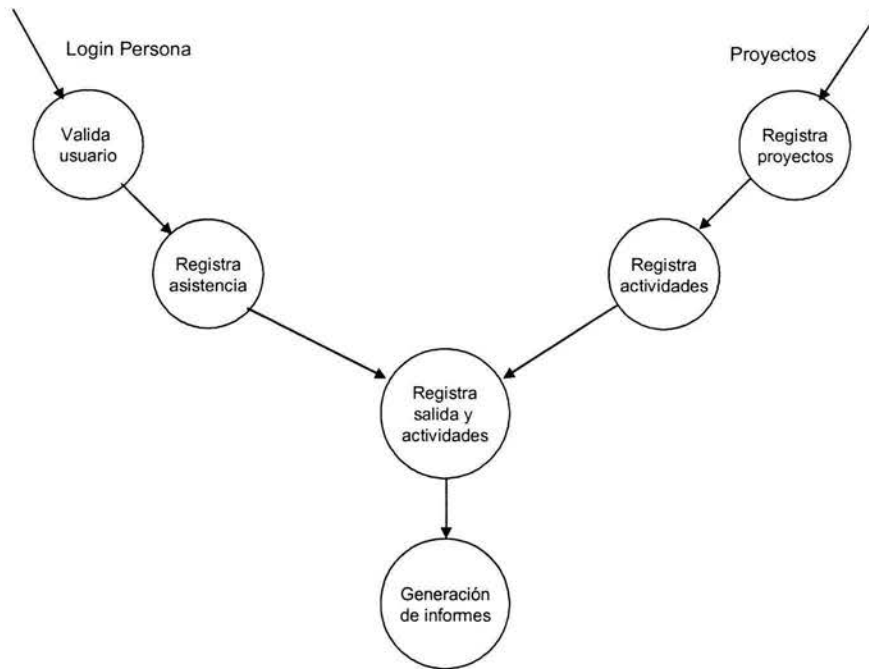


Figura 3.3

Como se observa, con el flujo de datos de usuario se procede a la validación. Dicha validación se detallará en el siguiente nivel, en donde se revisa el rol que desempeñará el usuario al ingresar.

Una vez que el usuario ha sido validado, se registrará en primer lugar la hora de inicio de la jornada laboral.

Por otro lado se deben registrar los proyectos sobre los cuales se va a trabajar. A cada proyecto se le van asignar sus actividades con fechas de inicio y fin, lo que nos permitirá controlar sus avances y no exceder los tiempos límites.

Una vez teniendo las actividades asignadas a las personas, se podrán registrar los porcentajes de tiempo asignados a cada actividad, el total de cada actividad en porcentaje debe ser igual a un 100% ocupado en cada día.

Con la información registrada por cada persona participante en los proyectos será posible obtener los reportes en donde se mostrarán las actividades, resúmenes y avances en los proyectos.

Diagrama de Flujo de Datos para el nivel 2.

Catálogos



Figura 3.4

Relaciones

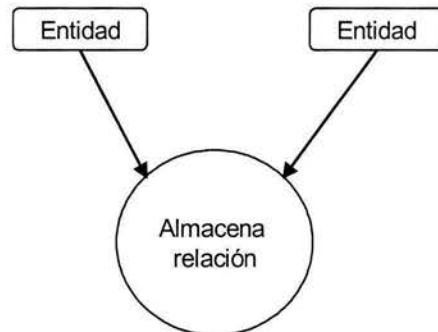


Figura 3.5

El flujo de datos para este módulo sólo requiere de los datos que se ingresarán en los catálogos figura 3.4, para el caso de relaciones figura 3.5 se utilizarán los datos de dos entidades para generar un valor que mantendrá la relación entre ellos.

Modelo Seguridad.

Alta usuario

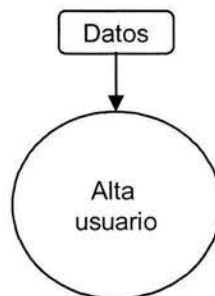


Figura 3.6

Para el alta de usuarios figura 3.6 serán necesarios sólo los datos personales.

El flujo de datos figura 3.7 es el más importante dentro del módulo de seguridad. Es aquí donde se tendrá la información en donde se validará si un usuario puede acceder o no a una pantalla.

Operación con seguridad

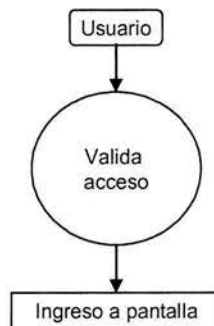


Figura 3.7

Cada proyecto tendrá varias actividades las cuales serán almacenadas en una entidad de relación figura 3.8.

Asignación de actividades a proyectos



Figura 3.8

Aquí se asigna a cada persona una o varias actividades de uno o varios proyectos sobre los que va a trabajar durante un periodo de tiempo figura 3.9.

Asignación de actividades a personas



Figura 3.9

El ingreso a cada pantalla figura 3.10 vendrá validado por los permisos que tenga una persona.

Operación Asistencia Entrada



Figura 3.10

La asistencia de entrada figura 3.10 será el primer proceso al cual los usuarios van a ingresar en donde solo se tomará su hora de inicio de labores. Para acceder a esta pantalla se deberá proporcionar el usuario y la contraseña.

Operación Asistencia Salida

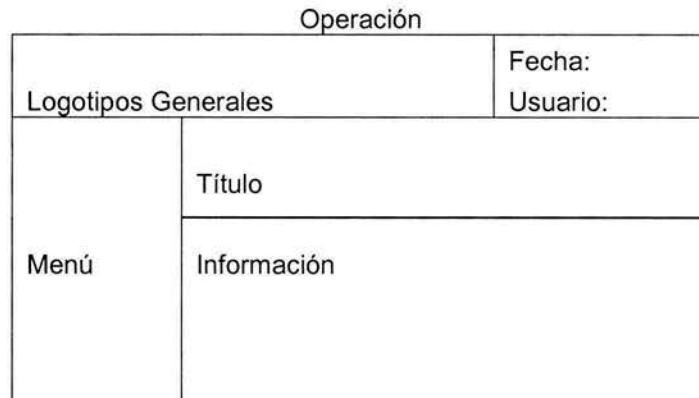


Figura 3.11

La operación de salida figura 3.11 será el registro de sus actividades. Para conocer que persona es la que se está registrando, se utilizará la validación de inicio proporcionando el usuario y la contraseña. La información que debe ingresar es el porcentaje de tiempo que dedicó desde la hora en que inicio sus labores hasta el momento que da por terminada su jornada laboral. Cada actividad estará ponderada de tal forma que el total en porcentaje por día dedicado a cada actividad debe sumar 100%.

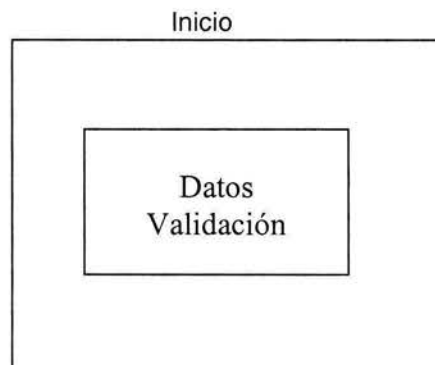
3.4 Análisis de la Interfaz gráfica

La estructura de la interfaz gráfica presenta los siguientes detalles mostrando de forma clara y navegable para cualquier usuario y con cuadros de diálogos explicativos, claros y sencillos en donde al usuario no le importe las tareas internas del sistema y solamente le interese la información básica para ingresar a la base de datos.



(Pantalla 3.1)

Dentro de la sección de Logotipos Generales vamos a colocar los distintivos de la empresa. La Fecha contendrá la información de la fecha de operación del servidor. El Usuario será el nombre completo de la persona que se firmó al inicio del sistema. La sección de Menú será una ventana dinámica en donde se colocarán las opciones de acceso dependiendo del contexto en el que se encuentre el usuario. La sección mayor de Información estará las pantallas de operación de uso común como el registro de entrada y salida, en donde se tendrá el título de cada pantalla en la parte superior de la pantalla (Pantalla 3.1).



(Pantalla 3.2)

La Pantalla de Inicio (Pantalla 3.2) sólo contendrá los campos para registrar el usuario que ingrese al sistema.

Consultas

Logotipos Generales		Fecha:
		Usuario:
Menú	Filtro	
	Encabezado de Columnas	
	Resultado del Filtro	

(Pantalla 3.3)

Las consultas tendrán la estructura de las pantallas generales de operación (Pantalla 3.3), sin embargo la sección de información se dividirá en dos partes, en donde la parte superior será el lugar en donde se coloquen los campos sobre los cuales se pueda realizar la selección de la información a consultar. La parte inferior contendrá los resultados de la búsqueda realizada a partir del filtro superior.

Catálogos

Logotipos Generales		Fecha:
		Usuario:
Menú	Área mantenimiento	
	Encabezado columnas	
	Lista registros actuales	

(Pantalla 3.4)

El mantenimiento de catálogos tendrá dos secciones (Pantalla 3.4), la superior será utilizada para los campos de edición de la información que se va a ingresar, o bien a la que se le harán cambios. La sección inferior será utilizada para mostrar los registros actuales del catálogo al que se le esté dando mantenimiento.

CAPÍTULO 4. DISEÑO DEL SISTEMA.

4.1 Diseño de la base de datos.

Una base de datos es una colección de datos o archivos relacionados de una manera estructurada, almacenados electrónicamente y pueden ser editados, unidos, organizados, y consultados. Es un método de organizar información en un formato uniforme.

La clave para que una base de datos sea efectiva está en el diseño de la estructura. Esta debe permitir que los datos puedan ser manipulados con flexibilidad y que suministre la información requerida que ayude a tomar decisiones correctas.

Pasos a seguir en el diseño de una base de datos:

1. Identificar los datos que se van a recoger y almacenar.
2. Decidir cómo organizar la información dentro de los campos.
3. Seleccionar los tipos de datos para cada campo.
4. Identificar el formato para cada campo.
5. Determinar el alcance para cada campo.
6. Agrupar los campos.

También es importante hacer las siguientes consideraciones:

- La finalidad de la base de datos y como afecta al diseño. Crear un plan para la base de datos que se ajuste a esa finalidad.
- Las reglas de normalización de la base de datos para evitar errores en el diseño de la base de datos.
- La protección de la integridad de los datos.
- Los requisitos de seguridad de la base de datos y los permisos de los usuarios.
- Los requisitos del rendimiento de la aplicación. El equilibrio entre el tamaño de la base de datos y la configuración del hardware es también importante de cara al rendimiento.
- El mantenimiento.
- La estimación del tamaño de la base de datos.

4.1.1 Modelo relacional.

El modelo relacional es una forma de ver los datos; es decir, para representar los datos mediante tablas y manipular esa representación mediante operadores. El modelo relacional se ocupa de tres aspectos de los datos: su estructura, su integridad y su manipulación.

La figura 4.1 muestra el modelo relacional de operación.

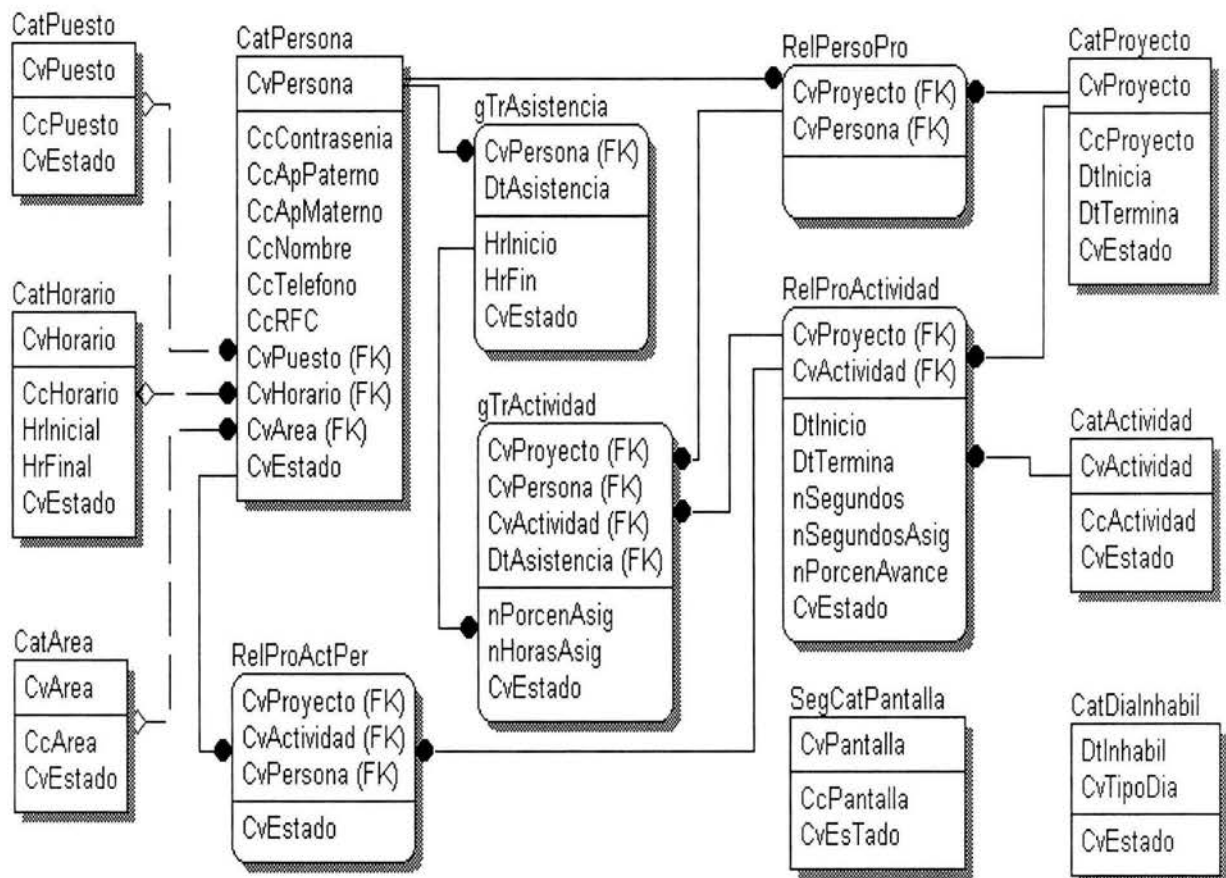


Figura 4.1

4.1.2 Normalización.

Las reglas de normalización están diseñadas para la prevención de anomalías e inconsistencia en los datos, así como en las operaciones realizadas y sus relaciones.

Primera forma normal.

Las tablas principales y que contienen la información necesaria para conocer las actividades de los proyectos y sus actividades se muestra en tres tablas.

Persona
CvPersona
CcContrasenia
CcApPaterno
CcApMaterno
CcNombre
CcTelefono
CcRFC
Puesto
Horario
Area

Proyecto
CvProyecto
CvActividad
DtInicaActividad
DtTerminaActividad
CcProyecto
Fecha Inicia
Fecha Termina

Asistencia
CvPersona
DtAsistencia
HrEntrada
HrSalida

Segunda forma normal.

Para evitar la dependencia directa de campos que no son parte de la llave primaria separamos las tablas principales y también se crearon catálogos que hacen que la información sea solo dependiente de la llave primaria.

Persona
CvPersona
CcContrasenia
CcApPaterno
CcApMaterno
CcNombre
CcTelefono
CcRFC
CvPuesto
CvHorario
CvArea

Puesto
CvPuesto
CcPuesto

Area
CvArea
CcArea

Horario
CvHorario
CcHorario
HrInicial
HrFinal

Proyecto
CvProyecto
CcProyecto
DtInicial
DtTermina

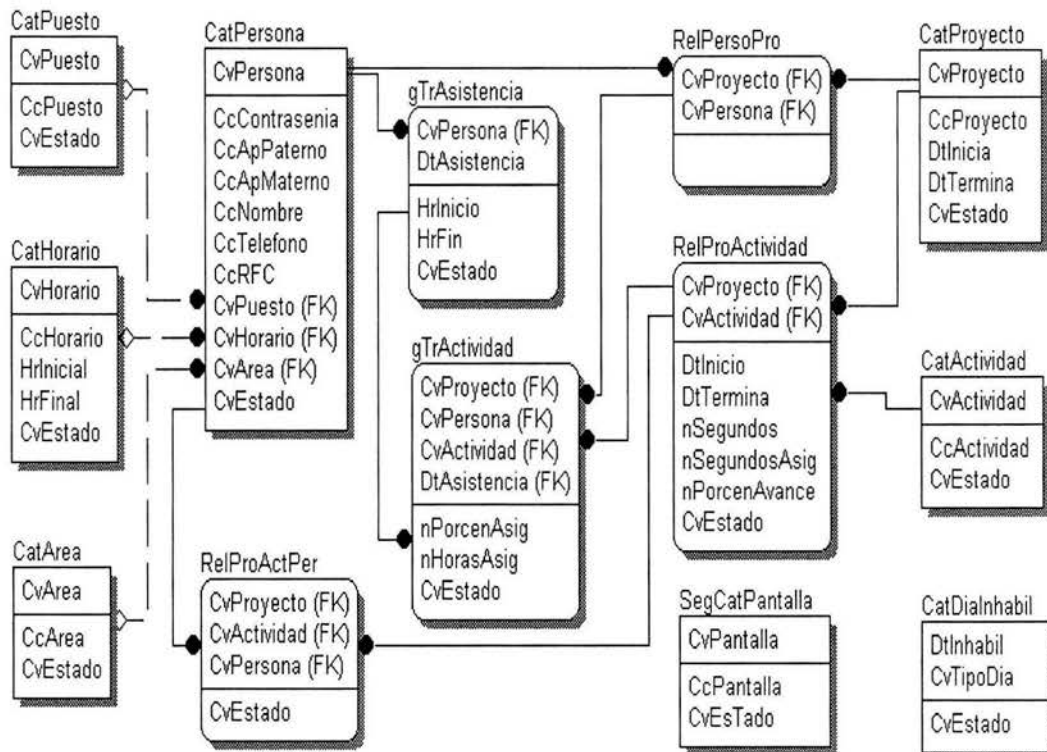
Actividad
CvProyecto
CvPersona
CvActividad
DtAsistencia
nPorceAsig
nHorasAsig

Proyecto Actividad
CvProyecto
CvActividad
DtInicio
DtTermina
nSegundos
nsegundosAsig
nPorcenAvance

Asistencia
CvPersona
DtAsistencia
HrInicio
HrFin

Tercera forma normal.

En la tercera forma normal se llega al modelo relacional mostrado en la figura 4.1



4.1.3 Diccionario de Datos.

El diccionario de datos es un listado organizado de todos los elementos de datos que son pertinentes para el sistema, con definiciones precisas que permiten que el usuario y el analista del sistema tengan una misma comprensión de las entradas y de las salidas.

Entidad: CatPersona

En esta entidad tendremos almacenada toda la información referente a los usuarios del sistema, así como sus datos generales.

Nombre: Catpersona				
Atributos:				
	Nombre	Tipo	Tamaño	Observaciones
K	CvPersona	String	16	Clave de Persona
	CcContrasenia	String	16	Contraseña de Persona para acceso al sistema
	CcApPaterno	String	20	Apellido paterno
	CcApMaterno	String	20	Apellido Materno
	CcNombre	String	20	Nombre de Persona
	CcTeléfono	String	20	Telefono
	CcRFC	String	13	Registro Federal de Contribuyentes
	CvPuesto	String	4	Puesto que desempeña en la empresa
	CvHorario	String	4	Tipo de horario el que debe laborar la persona
	CvArea	String	4	Área a la que pertenece
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: CatPuesto.

Catálogo en donde se definen los puestos que ocupa cada persona que labora en la empresa.

Nombre: CatPuesto				
Atributos:				
K	Nombre	Tipo	Tamaño	Observaciones
	CvPuesto	String	4	Clave de Puestos en la empresa
	CcPuesto	String	30	Descripción del Puesto
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: CatHorario

Catálogo para definir los diferentes horarios aplicados a las personas que laboran en la empresa. En esta entidad consideramos los diferentes horarios debido a que no todas las personas trabajarán bajo el mismo horario, de esta forma podrá haber personas que trabajen en horarios matutinos o vespertinos o quienes tengan horas de trabajo diferentes.

Nombre: CatHorario				
Atributos:				
K	Nombre	Tipo	Tamaño	Observaciones
	CvHorario	String	4	Clave de tipos de Horarios que se llevan en la empresa
	CcHorario	String	30	Descripción del horario que se lleva en la empresa
	HrInicial	DateTime	8	Hora en que inicia la jornada para el tipo de horario definido con la clave
	HrFinal	DateTime	8	Hora en que finaliza la jornada para el tipo de horario definido con la clave
CvEstado	String	4	Estado actual del registro en la tabla	

Entidad: CatArea

Catálogo para las áreas que existen en la empresa y a las que pertenecen las personas que estén incluidas en los proyectos.

Nombre: CatArea				
Atributos:				
K	Nombre	Tipo	Tamaño	Observaciones
	CvArea	String	4	Clave de Área
	CcArea	String	30	Descripción de el área en la empresa
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: RelProActPer

Relación de Actividades de un proyecto asignadas a una persona.

Nombre: RelProActPer				
Atributos:				
K	Nombre	Tipo	Tamaño	Observaciones
	CvProyecto	String	4	Clave de Área
	CvActividad	String	4	Clave de Actividad
	CvPersona	String	16	Clave de Persona
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: CatDialnhabil

Catálogo para días inhábiles; los días dentro de esta entidad pueden ser diferentes tipos (Festivos, Descanso).

Nombre: CatDialnhabil				
Atributos:				
	Nombre	Tipo	Tamaño	Observaciones
K	DtInhabil	DateTime	8	Fecha De Día Inhabil
	CvTipoDia	String	4	Clave para el tipo de día
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: gTrAsistencia

Entidad de trabajo en donde se guardará un registro diario de la asistencia de cada persona dada de alta en el sistema. En esta entidad se guardará la clave de la persona así como la fecha de registro. A la hora de entrada se generará un registro en donde se incluirá la hora de Inicio de labores, al término de la jornada solo se actualizará la hora de Fin o termino de la jornada.

Nombre: gTrAsistencia				
Atributos:				
	Nombre	Tipo	Tamaño	Observaciones
K	CvPersonal	String	16	Clave de Persona que registra su asistencia
K	DtAsistencia	DateTime	10	Fecha en que se registra su asistencia
	HrInicio	DateTime	8	Hora en la que registra la persona el inicio de jornada laboral
	HrFin	DateTime	8	Hora en la que termina la persona su jornada laboral
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: RelPersoPro

Entidad que relaciona una persona con uno o más proyectos en los que debe participar.

Nombre: RelPersoPro				
Atributos:				
	Nombre	Tipo	Tamaño	Observaciones
K	CvProyecto	String	4	Clave de Proyecto
K	CvPersona	String	16	Clave de Persona

Entidad: RelProActividad

Entidad que relaciona las actividades asignadas a un proyecto. Para cada actividad se define también una fecha de inicio y término con la finalidad de conocer si se esta en tiempo para su desarrollo.

Nombre: RelProActividad				
Atributos:				
	Nombre	Tipo	Tamaño	Observaciones
K	CvProyecto	String	4	Clave de Proyecto
K	CvActividad	String	4	Clave de Actividad
	DtInicio	Date	8	Fecha en que inicia o debe iniciar la actividad
	DtTermina	Date	8	Fecha en que termina o debe terminar la actividad
	nSegundos	Real	16	Segundos para toda la actividad
	nSegundosAsig	Real	16	Segundos Asignados a la actividad
	nPorcenAvance	Real	16	Porcentaje de Avance de acuerdo al tiempo asignado
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: CatProyecto

Catálogo de Proyectos que tiene la empresa. Es importante notar que cada proyecto es tomado de una empresa, de esta forma si se tienen dos proyectos de ventas, sabremos a que empresa corresponde cada uno.

Nombre: CatProyecto				
Atributos:				
	Nombre	Tipo	Tamaño	Observaciones
K	CvProyecto	String	4	Clave de Proyecto
	CcProyecto	String	30	Descripción del proyecto
	DtInicia	Date	8	Fecha en que inicia o debe iniciar el proyecto
	DtTermina	Date	8	Fecha en que termina o debe terminar el proyecto
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: CatActividad

Catálogo de las diferentes actividades que se pueden realizar en el desarrollo de proyectos, estas actividades son genéricas y aplican para todos los proyectos.

Nombre: CatActividad				
Atributos:				
	Nombre	Tipo	Tamaño	Observaciones
K	CvActividad	String	4	Clave de actividad
	CcActividad	String	30	Descripción de la actividad
	CvEstado	String	4	Estado actual del registro en la tabla

Entidad: SegCatPantalla

Catálogo de pantallas. Aquí se almacenan todas las pantallas con que cuentan los sistemas que se tengan en la empresa.

Nombre: SegCatPantalla				
Atributos:				
K	Nombre	Tipo	Tamaño	Observaciones
	CvPantalla	String	4	Clave de Pantalla
	CcPantalla	String	30	Descripcion de la Pantalla
	CvEstado	String	4	Estado actual del registro en la tabla

4.2 Selección de elementos y herramientas de desarrollo.

En general, una herramienta es cualquier dispositivo que, cuando se emplea en forma adecuada, mejora el desempeño de una tarea, tal como el desarrollo de sistemas de información basados en computadora. En general las herramientas se agrupan en las siguientes categorías: Análisis, diseño y desarrollo.

Herramientas para análisis

Estas herramientas ayudan a documentar un sistema existente, ya sea éste manual o automatizado, y a determinar los requerimientos de una nueva aplicación.

- Herramientas de especificación.
- Herramientas para presentación.

Herramientas para diseño

Las herramientas para diseño apoyan el proceso de formular las características que el sistema debe tener para satisfacer los requerimientos detectados durante las actividades de análisis.

- Herramientas de especificación.
- Herramientas para presentación.

Herramientas para el desarrollo

Estas herramientas ayudan al análisis a trasladar los diseños en aplicaciones funcionales

- Herramientas para ingeniería de software.
- Herramientas para pruebas.

4.2.1 Manejador de bases de datos

En el capítulo 2 se realiza el estudio de diferentes manejadores de bases de datos, la mayoría de ellos con soporte para el manejo, intercambio y administración de la información por medio del protocolo HTTP (Internet), Como resultado de este estudio se tomó la determinación del uso de SQL Server para el alojamiento y administración de la información de que será objeto el software que se desarrolla.

Las características que nos llevan a la utilización e implementación de SQL Server son las siguientes:

Requerimientos mínimos:

Procesador	Computadora con procesador Intel Pentium II o similar
Sistema operativo	Las versiones de 64-bit de la familia Microsoft Windows Server
Memoria	128 Megabytes (Mb) de RAM o más es recomendable
Disco duro	300 megabytes (MB) de espacio disponible en disco duro
Unidad de disco	Unidad de CD-ROM
Monitor	VGA o de mayor resolución

Características de SQL Server.

- El servidor de bases de datos Microsoft SQL Server cuenta con una perfecta integración con el sistema operativo Windows NT Server, que es el sistema que será utilizado para la implementación del software en desarrollo, y con otras herramientas populares de Microsoft como son Access y Excel.

- SQL Server es una aplicación completa que realiza toda la gestión relacionada con los datos. El servidor Web sólo tiene que enviarle una cadena de caracteres (la sentencia SQL) y esperar a que le devuelvan los datos. El servidor SQL se encarga de todo. Esta división de papeles es mucho más conveniente para la publicación Web.
- SQL Server puede recibir conexiones (clientes que desean manipular datos) desde cualquier ordenador conectado a Internet. En este aspecto es igual que un servidor Web o un servidor de correo. La diferencia es el número de puerto y, naturalmente, el protocolo con el que se comunicará con el cliente. Un navegador es un cliente para un servidor Web, pero no conoce el protocolo para hablar con un servidor de bases de datos. Access 2000 es un cliente para SQL Server.
- Cliente-servidor es el nuevo paradigma de la programación. Dentro de poco todo será "cliente-servidor" como antes lo fue "orientado a objetos". La programación cliente/servidor se utiliza cuando queremos realizar aplicaciones que utilicen redes y que comuniquen entre sí a varios ordenadores. Básicamente consiste en que el programa se divide en dos partes:
 - La parte Cliente, que reside en el equipo donde está el usuario y se encarga de la interacción con éste.
 - La parte Servidor, que reside en un ordenador conectado a la red permanentemente y se encarga de manipular los datos.

Ambas partes de la aplicación se comunican entre sí utilizando algún protocolo de red TCP/IP. La justificación de este paradigma es la minimización del tráfico de red. El ancho de banda de las redes de datos, sobre todo de Internet, es actualmente el recurso más precioso por lo escaso y caro que es. Si hay que optimizar un programa para algo, ese algo es utilizar el menor tráfico de red posible para economizar el ancho de banda.

- SQL Server admite procedimientos guardados (stored procedures) realizados en lenguaje SQL. Se trata de procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Los procedimientos guardados podrían ser la "parte servidor" de una aplicación cliente-servidor. La parte cliente podría ser una aplicación ASP, que puede acceder directamente a SQL Server.

- SQL Server puede manejar perfectamente bases de datos de TeraBytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos. Es un servidor y está pensado para gestionar tantos clientes simultáneos como admita la potencia del hardware del equipo en el que esté instalado.

Ventajas que ofrece SQL Server para publicar datos en Internet

- Estabilidad para el servidor Web.
Si utilizamos bases de datos mdb y recibimos muchas visitas o la base de datos crece mucho, existen riesgos de inestabilidad para el servidor Web. Por el contrario, SQL Server reside en un ordenador distinto al servidor Web, y está preparado para admitir cualquier número de visitas y tamaño de bases de datos. Utilizando SQL Server liberamos al servidor Web de las tareas relacionadas con los datos con lo que mejorará notablemente su rendimiento y estabilidad.
- Acceso directo a los datos
Con SQL Server es posible obtener un DSN en nuestro equipo local que apunte a una base de datos del servidor. Después podemos utilizar aplicaciones para acceder directamente a las tablas del servidor.
- SQL Server y ASP

ADO (ActiveX Data Objects) permite acceder de igual forma a bases de datos independientemente de su origen. Con esto es posible consultar bases de datos en la Web. Si utiliza actualmente ADO para acceder a bases de datos mdb, tendrá que cambiar muy poco código para utilizar SQL Server. Realmente la única diferencia está en la cadena de conexión. Si utiliza la base de datos, su cadena de conexión será del estilo a:

"DSN=midsn"

Si utiliza SQL Server debe añadir el login y contraseña:

"DSN=midsn;UID=login;PWD=contraseña"

Por lo demás no cambia nada. El DSN para acceder a SQL Server se obtiene en el panel de control del Servidor Virtual. Es necesario escribir un login y contraseña en un archivo de texto (ASP).

4.2.2 Servidor de Web

Es evidente que hasta hace relativamente poco tiempo, la mayoría de los servidores Web estaban basados en plataformas UNIX. Esta situación ha cambiado y está cambiando, de tal manera que muchos administradores de servidores Web han decidido utilizar Windows, algo que para muchos era una opción poco confiable.

En la actualidad, y más aún en el futuro, Internet está convirtiéndose en el ámbito de negocio. Además, sus mecanismos de comunicación se están trasladando a las redes locales de las corporaciones, tanto en lo referente a los mecanismos y protocolos de comunicación, como en las técnicas de desarrollo de aplicaciones y medios de acceso a la información por parte de los empleados, colaboradores y asociados. Es por todo estos motivos que un sistema operativo con vocación de liderar el mercado no puede tener éxito si no ofrece de manera nativa la posibilidad de que la información pueda compartirse, no sólo mediante los mecanismos convencionales en una red local, sino utilizando tecnología Web. Es decir, es necesario que la plataforma facilite la creación tanto de intranets como la posibilidad de definir la presencia en Internet.

En los primeros meses de 1998, Microsoft hizo público Windows NT 4.0 Option Pack (paquete opcional de Windows NT 4.0), una extensión al sistema operativo de Microsoft que dió un auténtico vuelco a sus funcionalidades como servidor en Internet. , Option Pack convierte a un servidor Windows NT en algo que se parecía poco, en el ámbito del web, a lo que incluía el paquete básico del sistema operativo, ofreciendo, a modo de resumen, los siguientes servicios básicos:

- **Internet Information Server 4.0:** el antecesor de la actual versión de IIS incluía servidores web y FTP.
- **Active Server Pages 2.0:** Microsoft proporcionó una nueva versión de la exitosa plataforma de servidor para el desarrollo de aplicaciones web utilizando técnicas de Scripting.
- **Servidor de noticias NNTP:** servidor de news compatible con el estándar Network News Transport Protocol (NNTP).
- **Servidor de correo SMTP:** el paquete incluye un servidor de correo saliente (SMTP) que principalmente nos va a permitir establecer conexiones para la transmisión de correo que, entre otras cosas, van a facilitar la inclusión en nuestro servidor de páginas web con formularios cuyos resultados serán enviados a una dirección de correo en lugar de ser procesados directamente con un CGI o página de servidor activo.
- **Microsoft Management Console:** MMC, tan familiar para los administradores en plataforma Microsoft desde Option Pack, SQL Server 7

y definitivamente asentada con Windows 2000 como herramienta de administración, fue incorporada por primera vez.

- **Transaction Server (MTS):** primera versión realmente funcional del elemento de servidor para el desarrollo y gestión de aplicaciones en nuestro servidor que hagan uso transaccional de componentes.
- **Index Server 2.0:** motor de búsqueda en web para Windows NT e IIS 4.0.
- **Certificate Server:** servidor para la creación de certificados que permitan la implementación de los mecanismos de conexiones seguras de Secure Sockets Layer.
- **Site Server Express:** versión recortada de las herramientas de análisis de Site Server 3.0.
- **Extensiones para FrontPage 98:** extensiones necesarias para poder crear sedes web con FrontPage 98.

A través del tiempo los servicios de Microsoft han sido mejorados con las nuevas versiones de IIS y actualmente los Servicios de Microsoft Internet Information Server (IIS) 6.0 con Windows Server 2003 proporcionan capacidades de servidor Web integrado, confiable, escalable, seguro y administrable en una intranet, una extranet o en Internet. IIS 6.0 incorpora mejoras significativas en la arquitectura para cubrir las necesidades de los clientes alrededor del mundo.

IIS 6.0 y Windows Server 2003 introducen muchas características nuevas para la administración, disponibilidad, confiabilidad, seguridad, rendimiento y escalabilidad de los servidores de aplicaciones Web. IIS 6.0 también mejora el desarrollo y la compatibilidad internacional. IIS 6.0 y Windows Server 2003 proporcionan la solución para servidores Web más confiable, productiva, conectada e integrada.

¹PERSONAL WEB SERVER

Personal web server, es ideal para intranets, hogares, escuelas y pequeños grupos de negocios. En corporativos, en algunos departamentos se pueden enviar documentos HTML y compartir información con cualquier persona dentro de la Intranet de la compañía. Los usuarios de Windows 95/98SE/ME/NT pueden compartir contenidos del web, de una manera fácil tal como compartir folders en una red.

También representa una gran solución en cuanto a costo se refiere, hablando de pequeños negocios y escuelas, porque no se necesita dedicar una PC como servidor de WEB, en lugar de eso, se pueden tener computadoras normales para funcionar con el personal web server. Los usuarios domésticos, pueden experimentar el publicar en web, utilizando su computadora como servidor de web y publicar de manera fácil su información en internet.

¹ <http://www.studiodeluxe.net/pws/index.htm>, <http://www.tergestesoft.com/~eddysworld/pws.htm>

Personal Web Server transmite información en páginas HTML usando http(Hypertext Transport Protocol). Este proporciona mucha de la funcionalidad de Microsoft Personal Web Server, incluyendo la habilidad para:

- Publicar páginas web en internet, sobre una red de área local o en una Intranet.
- Soporta programas de Microsoft ActiveX.
- Transmitir o recibir archivos utilizando servicio FTP.

Para Windows 98 proporciona una interfaz de administración gráfica diseñada para usuarios que no tienen experiencia previa en administrar sitios web. Esta interfaz también proporciona una herramienta de administración por default para Windows NT Workstation; sin embargo, con Windows NT Workstation, se puede administrar el sitio personal para publicar con Internet Service Manager, de la misma manera que la herramienta de administración usada para controlar Internet Information Server.

El servidor de Web que vamos a utilizar es Personal Web debido a que éste es la plataforma estandar que tenemos en las oficinas en donde estamos realizando el proyecto y además cuenta con las características básicas de Internet Information Server.

Comparación de productos

Características	PWS para Windows NT Workstation	PWS para Windows 98
Usos típicos	Características completas de desarrollo en sitio o para publicación personal en una Intranet corporativa.	Publicación personal en una Intranet corporativa de bajo volumen.
Servicio WWW	Si	No
Servicios FTP	Si	No
Límite de conexión	10	10
Active Server Pages	Si	Si
Fuentes de Publicación	Locales y de red	Sólo locales
Autenticación	Básica o Windows NT pregunta/respuesta	No

Requerimientos para instalar Personal Web Server:

Procesador	33MHz 486 mínimo	Recomendado 90MHz Pentium
RAM	16Mb	Recomendado 20-32 Mb
Espacio en Disco	30Mb	Recomendado 40Mb
Monitor	VGA	Recomendado Super VGA

4.2.3 Herramienta de Programación

Como se menciona en el capítulo 2 se cataloga a los lenguajes de programación para el Web, como lenguajes del lado del servidor y lenguajes del lado del cliente. En la construcción de páginas para Internet también podemos hablar de 2 tipos de páginas: las páginas estáticas y las páginas dinámicas.

Las páginas estáticas se construyen con lenguaje HTML, son sencillas de crear, sin embargo ofrecen pocas ventajas tanto para los desarrolladores como para los visitantes ya que sólo presentan textos planos con imágenes y algunos contenidos multimedia.

Las páginas dinámicas incluyen efectos especiales y funcionalidad, para lo cual se utilizan otros lenguajes de programación a parte del HTML. No hay que perder de vista que el HTML es la base del desarrollo Web y que generalmente al escribir una página dinámica el código de los otros lenguajes de programación se incluye embebido dentro del mismo código HTML.

Páginas dinámicas del lado del cliente

En estas páginas toda la carga de procesamiento de los efectos y funcionalidades la soporta el navegador. El código necesario para crear los efectos y funcionalidades se incluye dentro del mismo archivo HTML y es llamado SCRIPT. Cuando una página HTML contiene scripts de cliente, el navegador se encarga de interpretarlos y ejecutarlos para realizar los efectos y funcionalidades.

Páginas dinámicas del lado del servidor

Son reconocidas, interpretadas y ejecutadas por el propio servidor. Este tipo de páginas son especialmente útiles en trabajos en los cuales se tiene que acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad no se pueden realizar cálculos en el equipo del usuario. Cabe destacar que las páginas dinámicas del lado del servidor son necesarias porque para hacer la mayoría de las aplicaciones Web se debe tener acceso a muchos recursos externos al ordenador del cliente, principalmente bases de datos alojadas en el servidor de Internet.

Las ventajas de este tipo de programación son que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable.

Para escribir páginas dinámicas de servidor existen varios lenguajes, tales como Common Gateway Interface (CGI) comúnmente escritos en Perl, Active Server Pages (ASP), Hipertext Preprocesor (PHP), y Java Server Pages (JSP).

Actualmente para el desarrollo del sistema se necesita contar con una tecnología que permita crear páginas dinámicas, que puedan ser ejecutadas del lado del servidor, debe ser flexible y fácil en su programación y que no represente demasiadas dificultades para efectuar conexiones a una base de datos. Tanto el sistema operativo, como el servidor Web y la base de datos están basados en tecnologías de Microsoft (Windows 2000/NT, IIS, SQL Server) por lo que también se requiere que la herramienta pueda interactuar y explotar al máximo las capacidades de estos elementos.

Si bien la tecnología JSP tiene grandes ventajas, se requieren conocimientos sobre java y programación orientada a objetos, además de que al ser tecnología java la programación es muy estricta en su desarrollo.

PHP y PERL a pesar de ser gratuitos lo cual los hace muy populares requieren cierto nivel de conocimientos de programación para implementarlos (lenguaje C particularmente), además de que su mayor capacidad la desarrollan cuando se ejecutan en entornos Unix o Linux.

ASP tiene muchas ventajas. La mayor para el caso particular de este desarrollo es ser una tecnología nativa de IIS; es decir forma parte de las tecnologías de Microsoft que van a implementar en el desarrollo del sistema, por lo que no será necesario obtener otra herramienta de manera separada e integrarla a la plataforma Microsoft seleccionada. Otra ventaja importante es que reemplaza la forma tradicional de intercambiar información entre usuarios. Los métodos tradicionales de intercambio cliente-servidor involucraban la tecnología Common Gateway Interface (CGI). CGI requiere una aplicación separada desde un documento Web para procesar todos los datos regresados al servidor. Esto tiene desventajas tanto en la seguridad como en el procesamiento de recursos. Active Server Pages permiten que el código sea incrustado en un documento HTML y que corra en el servidor, es un concepto simple. El programa que necesita correr en el servidor se coloca entre etiquetas especiales (script tags). Estas etiquetas le dicen al servidor que no envíe el código que está entre ellas al cliente, por lo que no será necesario hasta que se reciba una solicitud del cliente, después se procesará en el servidor basado en la información enviada de regreso desde el cliente y nunca dejará el servidor haciendo por lo tanto a ASP muy seguro y fácil de programar. ASP se ejecuta sobre el servidor y envía datos de regreso al cliente (a través del browser). ASP es compatible con múltiples plataformas.

Existen varios lenguajes que se pueden usar para hacer ASP. El más comúnmente utilizado es VBScript y es por mucho el más fácil, ya que es nativo de Microsoft. Otra gran ventaja con respecto a PHP es que ASP permite de una manera sencilla la conexión a bases de datos diferentes tales como Acces. Sql Server a Oracle o

alguna otra base de datos; en tanto que para lograr esto en PHP se requiere implementar una API diferente en la reconexión.

Por lo cual la tecnología ASP (figura 4.1) nos proporciona la facilidad de implementación y programación que necesitamos para el sistema, la seguridad, velocidad y bajo consumo de recursos y la posibilidad de poder proporcionar un mantenimiento más rápido. Aunado a la plataforma con tecnologías Microsoft de la que es parte y que permitirá un desarrollo que explote en mayor capacidad las características de cada uno de los componentes necesarios para desarrollar e implementar el sistema.

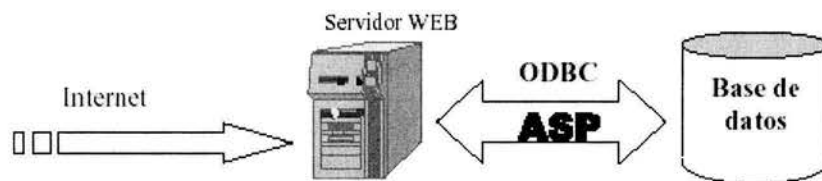


Figura 4.1 Elementos de ASP

Forma de trabajar de ASP

Al solicitar información esto es lo que sucede Figura 4.2:

1. Desde el navegador del usuario se solicita cierto archivo ASP al servidor Web.
2. El script del lado del servidor comienza a ejecutarse con el código ASP.
3. ASP procesa el archivo solicitado de manera secuencial (arriba-abajo), ejecuta algunos comandos contenidos en el archivo, y produce una página en formato HTML.
4. La página Web es devuelta al navegador, lista para ser visualizada.

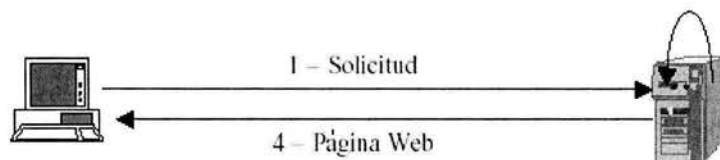


Figura 4.2 Método de trabajo de ASP

Como su código es ejecutado en el servidor, es el servidor Web el que realiza todo el trabajo y las páginas HTML estándar son generadas y enviadas al navegador. Esto significa que sus páginas están limitadas únicamente al soporte que le brinda el servidor de Internet.

De acuerdo a las características descritas anteriormente, se elige ASP por ser una tecnología con condiciones de seguridad muy amplias, flexible en su programación y que permite una interacción total con las tecnologías de Microsoft utilizadas en el desarrollo del sistema. Todo esto permitirá ahorro de tiempo en los desarrollos, un mantenimiento más sencillo, consumo menor de recursos de hardware y software lo que finalmente impactará de manera positiva en el producto final que se entrega al usuario.

4.2.4 Hardware y Arquitectura de Desarrollo

El desempeño que tienen los sistemas basados en procesadores de 1-GHz es superior en tres áreas que causan mucha expectativa, aplicaciones de negocios, Internet y sistemas de 3-D. Con las necesidades de funcionamiento de aplicaciones en Internet, la necesidad de calidad de imágenes, sonido y video, los fanáticos de los juegos que siempre requieren mayor poder en las gráficas de 3-D y la velocidad de respuesta que cada día se explota más en las áreas de negocio.

Es importante señalar que no todas las computadoras que tengan procesador de gran velocidad tienen la misma configuración, y el desempeño que puedan tener no va a depender únicamente de la arquitectura del procesador o a la velocidad del reloj. Otros factores que influyen en el desempeño son los otros componentes del sistema, como son conectados y la configuración del equipo y esto puede ser una gran diferencia al tomar una decisión.

La elección de la arquitectura de hardware, está dada por las características del software que se va a utilizar en el desarrollo del sistema, estamos hablando de herramientas de Microsoft. Recordando que la introducción de microprocesadores como el 386 SX influyeron en el desarrollo de los entornos gráficos de Windows. De este modo la plataforma elegida es Intel, de acuerdo a los comparativos presentados en el capítulo 2 es evidente las ventajas que ofrecen los productos Intel; de manera específica se va a implementar sobre un equipo con procesador Pentium III o superior.

La arquitectura de desarrollo implica tener las siguientes características:

- El usuario debe tener acceso a un equipo con conexión a Internet.
- La información debe publicarse por medio de un servidor Web.
- La información debe estar almacenada y ser susceptible de ser modificada.

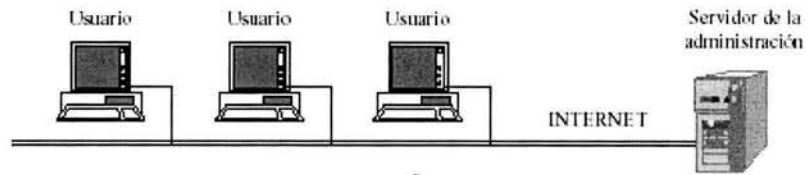


Figura 4.3 Infraestructura

Los datos publicados a través de Internet deben cumplir tres propiedades fundamentales:

1. La información debe transformarse en datos que puedan ser transmitidos a través de la red. Es lo que se conoce como *publicar*.
2. Esta información sólo debe poder ser manipulada por personal autorizado.
3. Debe estar puesta al día, lo que implica añadir aquello que sea nuevo y eliminar lo que se quede obsoleto.

CAPÍTULO 5. IMPLEMENTACIÓN Y PRUEBAS.

5.1 Implementación del Sistema

La etapa de implementación implica el inicio de la codificación de algoritmos y estructuras de datos, definidos en la etapa de diseño, ésto se lleva a cabo para el lenguaje de programación seleccionado y/o para el sistema de bases de datos elegido.

Uno de los motivos de esta etapa es que la eficacia de las etapas anteriores a la implementación, solo se demuestra una vez que el sistema se implementa y se utiliza. También es de gran importancia efectuar revisiones formales al producto, ésto con el fin de:

- Descubrir errores en la función, la lógica o en la implementación del software.
- Verificar que el software cumple con los requisitos establecidos en el análisis
- Verificar que el software se ha desarrollado bajo ciertos estándares (definidos en el análisis) y que ha sido un desarrollo uniforme.

Al realizar un desarrollo acorde al análisis y realizar revisiones formales al software, se garantiza la calidad del mismo, y se minimizan los posibles errores en el producto que se entrega al usuario final.

5.1.1 Desarrollo de la Base de Datos

El desarrollo de la base de datos la hemos realizado a través de las herramientas propias de SQL Server.

La instalación de SQL Server se realizó con las opciones que presenta por default a excepción del lugar donde se instala la base de datos y el programa, para nuestro caso ha quedado en una partición d:\basedatos\win98. El nombre del servidor de base es muy importante debido a que será utilizado para configurar las conexiones a las bases.

Una vez instalado SQL Server fue creada la base de datos apoyados por el wizard; la base de datos la hemos llamado ADMPROTE. Este nombre es importante porque será utilizada en archivos donde se incluye como parámetro de conexión. La creación de la base también se realiza con las opciones por default.

Para la creación de las tablas nos hemos apoyado del wizard de SQL Server, el cual nos permite de forma gráfica agregar campos incluyendo el tipo de datos que almacenará, así como su longitud y la opción de almacenamiento de valores nulos.

Con el apoyo del wizard en la creación de tablas también es sencillo establecer las llaves de cada tabla, sólo se necesita seleccionar los campos que formarán parte de la llave y seleccionar el icono que establece como llaves a dichos campos.

El agregar campos es totalmente sencillo y dinámico pues basta con ponerle un nombre y un tipo.

5.1.2 Implementación de la interfaz gráfica

El desarrollo de la interfaz gráfica es la presentación final para los usuarios que nos permite interactuar con la base de datos.

Principiamos con la creación de Frames para separar la pantalla en áreas de trabajo, información general y menús.

Los frames se han realizado de forma fija con las etiquetas de HTML.

```
<frameset rows="*%" onload="reDisplay('0',true);" cols="18%,*%"
frameborder="yes" border="1" framespacing="1">
  <frame src="Menu_Seguridad.asp" name="MENU" scrolling="NO" noresize>
  <frameset rows="8%,*%" framespacing="1" frameborder="NO" border="1">
    <frame src="Arriba_Seguridad.asp" name="Superior" scrolling="NO" noresize>
    <frame src="Trabajo_Seguridad.asp" name="CONTENIDO" scrolling="NO"
noresize>
  </frameset>
</frameset>
```

Para obtener un mejor desempeño y evitar navegar a otras páginas mientras este en uso el sistema, se ha utilizado una ventana exclusiva, abierta al momento de registrar un usuario y contraseña válida.

El registro de usuario y contraseña se ha realizado en una página con código HTML puro, en donde hacemos uso de una hoja de estilos la cual utilizamos en todas las pantallas del sistema a excepción de los reportes.

En la hoja de estilos definimos las características generales del color de fondo, los tipos de letra, los tamaños de cajas de texto así como de los objetos select, también están definidos los colores y tipos de letra usadas en las consultas.

```
body {background-color: #A7BEDC;}
input.General {
  COLOR: Black;
  FONT-FAMILY: arial;
  FONT-SIZE: 9px;
  height: 17px;
  background-color: #B0C0EC;
}.....
```


La validación de acceso al sistema se realiza a través de una pantalla que solo contiene dos campos tipo texto, uno utilizado para la clave de usuario y otro para la contraseña, además contiene un botón submit, el cual activa una pantalla de tipos ASP en donde se realiza una conexión a la base de datos a través de un componente de conexión de windows.

En esta primera conexión se va directamente a la base de datos para verificar si existe el usuario y corresponde a la contraseña proporcionada. En caso de que uno de los datos sea erróneo, el ASP devolverá una pantalla de error y volverá a pedir el usuario y contraseña.

Una vez validado el usuario se abre una nueva pantalla del browser para operar el sistema.

La pantalla principal del sistema está dividida en Frames los cuales se manejan con un propósito específico.

Del lado izquierdo está el menú con todos los accesos al sistema, éste lo hemos dividido en módulos de acuerdo a su operación. Cada opción tiene una liga directa a una pantalla la cual contiene una validación de acceso.

```
tocTab[jr++] = new Array ("0", "Operaci&oacute;n", "contenido.html");
tocTab[jr++] = new Array ("0.1", "Registro Entrada", "Entrada_Personal.asp");
tocTab[jr++] = new Array ("0.2", "Registro Salida", "Salida_Personal.asp");

<% Dim CvPantalla
    CvPantalla = "Actividades.asp"
%>

<!--#include file="Include/ValidaAcceso_asp.inc"-->
<FrameSet rows="40%,*">
    <frame name="FORMA" src="Mnto_Actividad.asp">
    <frame name="CONSULTA" src="RS_Actividad.asp">
</frameset>
<!--#include file="Include/ValidaAccesoFin_asp.inc"-->
```

La validación existe en todas las pantallas de acceso directo incluidas en el menú.

El include de validación de acceso, contiene una conexión directa a la base de datos, para verificar si el usuario tiene permisos de accesos a dicha pantalla, en caso de tener los permisos la pantalla será abierta, de lo contrario mostrará una ventana con un mensaje indicando que no tiene permisos para acceder a dicha pantalla.

```

Set ObjConn=CreateObject("AdmProTe.ClsAdmTe")
'verificar conexion
okConn=objConn.ConectaDB()
'Recuperar datos por parametros

if (okConn) then

    StrSelect="select * from relpersopantalla "
    StrWhere = " CvPersona = " & Session("CvPersona") & " and CvPantalla = "
    & CvPantalla & ""
    StrOrder= ""
    Set rsPermiso = ObjConn.selecciona (StrSelect, StrOrder, StrWhere)

```

En las pantallas en donde se utilizan dos frames, como es el caso de los mantenimientos, en la parte superior dejamos un ASP con referencias a un archivo encargado de realizar las actualizaciones.

Para realizar las actualizaciones los archivos que se encuentran detrás de la operación de guardar o actualizar, requieren al menos de dos parámetros, uno es utilizado para saber que tipo de operación se va a realizar y el otro u otros para realizar los cambios. Los parámetros son recibidos con instrucciones de Request.Form para cada parámetro de entrada en actualizaciones.

```

CvActividad = Request.Form("CvActividad")
CcActividad = Request.Form("CcActividad")
CvEstado = Request.Form("Estado")

```

Para la actualización utilizamos un componente desarrollado en Visual Basic, el cual contiene la conexión a la base de datos y funciones con parámetros de entrada para las actualizaciones o para insertar nuevos registros.

El componente es del tipo dll, el cual debe ser registrado en Windows para su utilización. La forma de registrarlo es a través del comando de Windows:

```

Regsvr32 Ruta y Nombre del Archivo dll

```

Este componente requiere de un archivo de texto de donde toma los parámetros de conexión a la base de datos.

```

[ADMPROTE]
DATABASE=ADMPROTE
SERVER=JAVENDANO
USER=USR_ADMPROTE
PASS=USR_ADMPROTE

```

El archivo se llama AdmProte.ini y debe estar al mismo nivel en el que se ponga el componente de Visual Basic, porque por default lo tomará de esta ruta.

Para la parte de consultas utilizamos una conexión directa a la base de datos a través de un ADO de Windows para obtener un RecordSet del cual podemos utilizar parámetros de paginación y número de registros desplegados en cada página.

El despliegue de la información obtenida en los RecordSet es pintada con un código de HTML con ASP para recorrer el RecordSet.

Todos los cálculos realizados por el sistema son llevados a cabo en los mismos programas de ASP que los incluyen.

Debido a que se están utilizando variables de sesión para validar el acceso a pantallas, después de un tiempo si las pantallas no son actualizadas dichas variables adquieren un valor nulo por default, por lo que el sistema podrá cerrar cualquier acceso a las pantallas después de este tiempo.

5.2 Pruebas realizadas al Sistema

Las pruebas son una parte muy significativa del proyecto de adecuación de las aplicaciones, no sólo por su importancia en el logro de resultados correctos sino por el tiempo y recursos requeridos. Se estima que demandan del orden del 60% del total del proyecto.

- **PRUEBA DE ACEPTACIÓN:** Sirve para que el usuario pueda validar si el producto final se ajusta a los requisitos fijados, es decir, si el producto está listo para ser implantado para el uso operativo en el entorno del usuario.
- **PRUEBAS DE INTEGRACIÓN:** Su objetivo es realizar pruebas para detectar errores asociados con la interacción de los módulos, así como las interfaces entre componentes de la arquitectura del software.
- **PRUEBAS DEL SISTEMA:** Proceso de prueba de un sistema integrado de hardware y software para comprobar si cumple los requisitos especificados.

Etapas de Pruebas

Prueba de especificaciones: (código)

El analista examina las especificaciones que el programa debe hacer y cómo debe desempeñarlas bajo diferentes condiciones, no es una prueba completa, pero se piensa que si el programa cumple con las especificaciones no fallará.

Prueba Unitaria:

Se prueban los programas que conforman un sistema, a veces se denomina prueba de programas.

Primero se enfoca a los módulos independientes, para localizar errores en la lógica y en la codificación, se trata de evitar aquellos que interactúen.

La prueba unitaria debe llevarse a cabo de abajo hacia arriba, es decir comenzando con los módulos más pequeños y de menor nivel uno a la vez. Se llama *prueba ascendente*.

La *prueba descendente*, es exactamente al revés de la anterior, se comienza por los módulos de arriba hacia los de abajo, pero como no se llega a probar los módulos inferiores se utilizan los que se llaman módulos *esclavos*, simplemente para mandar mensajes de OK ó de ERROR para los módulos superiores, los módulos de menor nivel no se prueban, por eso a veces se combina la prueba ascendente con la descendente.

Prueba de Sistemas

No prueba el Software, sino la integración de cada módulo en el Sistema, también hace pruebas para encontrar discrepancias entre el sistema y su objetivo original, las especificaciones actuales y la documentación de sistemas. La preocupación principal es la compatibilidad de los módulos individuales.

Ej. error en el tipo de un campo que da una salida irreal.

Pruebas especiales de Sistemas

1. Prueba de Carga Pico:

Se dan por lo general en los sistemas en línea, tipo Bancario, probando qué pasa si todos los cajeros entran al mismo tiempo, ó salen al mismo tiempo. Es una prueba donde se buscan situaciones reales.

2. Prueba de almacenamiento:

Los analistas cuando diseñan el sistema, deben analizar también el espacio que puede llegar a ocupar, funcionando a los límites, buscando cierta compatibilidad con el espacio de Disco que se va a utilizar en la computadora que se va a correr. Justamente esto se tiene que probar antes de la puesta en marcha.

3. Prueba de desempeño en tiempo

Se pone en práctica antes de la puesta en marcha para determinar cuánto tomará recibir la respuesta a una consulta u obtener una copia de respaldo de un archivo, el problema a detectar radica en que un sistema puede funcionar bien con una cantidad determinada de transacciones y ponerse inaceptablemente lento cuando esté cargado en su totalidad, la idea es no tener que padecer la pérdida de datos reales.

4. Prueba de Recuperación

Sólo con crear una situación de pérdida ó de falla de datos donde los usuarios se vean obligados a volver a cargar y recuperar una copia de respaldo, los analistas pueden determinar si los procedimientos de recuperación son adecuados.

5. Prueba de procedimiento

El analista pone siempre mucho énfasis en la creación de los manuales de procedimiento, ya que un buen diseño de ellos es insustituible, sobre todo siendo lo más críticos posible en los detalles del manejo del sistema incluyendo todo esto en la documentación.

Lo que en realidad el analista necesita es ejercer la prueba de RRHH. donde se observa en el usuario las acciones y reacciones que va teniendo en el manejo del sistema. Cuidarse de pantallas en blanco donde quizás el usuario apague, desconecte líneas TE, etc., tratar de poner siempre carteles si hay algún proceso para que el usuario se informe. En resumen, es responsabilidad del analista anticipar las preguntas que surgirán en la mente de los usuarios cuando interactúan con el sistema. Otra prueba importante es ver como el usuario carga los datos en el sistema.

6. Diseño de Datos de prueba

- ✓ Utilización de datos reales de prueba.

Estos datos son los que se extraen de los archivos de la empresa. Los analistas les piden a los usuarios que carguen datos como si estuvieran en realidad realizando sus actividades normales en la empresa. Conseguir datos reales suficientes no siempre se logra y a veces no se ven todas las combinaciones o formatos que pueden llegar a introducirse. Lamentablemente, se llega a ignorar en algunos casos, los datos que realmente generan fallas en el sistema.

- ✓ Utilización de datos artificiales de prueba.

Estos datos se crean únicamente para propósitos de prueba, dado que se arman para probar todo aquello que con los datos reales no se pudo. Hacen posible la prueba de todas las rutas lógicas y de control a través del programa.

7. Bibliotecas de prueba

Es un conjunto de datos desarrollados para probar completamente un sistema de programas, se almacenan en forma legible en Disco y se utilizan por todas las personas que trabajan con un sistema específico. Estas bibliotecas sirven también para probar el sistema cuando ya está avanzado, como un mantenimiento mientras el sistema va evolucionando y los programas se van modificando y se deben volver a probar. Por la tanto las Bibliotecas de datos se deben mantener a través de la vida del sistema, ya que conforme se realiza cada cambio se tengan disponibles otra vez datos confiables para probar el sistema.¹

¹ http://www.cema.edu.ar/~asilvestri/Etapa_de_Pruebas.doc

Las primeras pruebas consistieron en la aceptación por los usuarios. La prueba se llevo a cabo desde el inicio del diseño en donde a los usuarios se les mostró la pantalla en un documento para su autorización y validación. Una vez validadas las pantallas por los usuarios se dió inicio a su desarrollo.

Terminado el sistema se comenzó con las pruebas unitarias con el propósito de detectar errores en los programas. Las pruebas se realizaron de forma ascendente tomando primero los catálogos para continuar con las pantallas de operación.

Después de haber probado cada módulo y pantalla iniciamos las pruebas sobre el sistema completo, observando la secuencia de la información desde la carga de catálogos con información real, al igual que con proyectos y actividades que se están realizando al momento de las pruebas. Realizamos entradas y salidas de personal cuidando que la salida tuviera variaciones en horas y minutos.

Ya realizadas las pruebas de operación completas continuamos haciendo pruebas conectando a varios usuarios al mismo tiempo en la misma red para observar el comportamiento con carga.

Para cada uno de los usuarios se fueron grabando sin ningún problema y con una respuesta inmediata.

Otra de las pruebas realizadas fue al dar de baja la base datos mientras se operaba el sistema, con el propósito de verificar el error generado. De dicha prueba se obtuvo un error de SQL Server el cual se refleja en las pantallas como error en la conexión a la base de datos.

Para continuar con las pruebas del sistema armamos una serie de datos para la obtención de información al momento de registrar entradas y salidas de personal en diferentes días y en diferentes horas con la finalidad de observar los avances en tareas al cabo de varios días, para lo cual el sistema tuvo un comportamiento sin existencia de problemas.

CONCLUSIONES

Mantener un control estricto sobre los tiempos dedicados a cada actividad por parte de los participantes en el desarrollo de un proyecto se ha vuelto en la actualidad un factor importante para obtener los mejores resultados en beneficio de la empresa y los clientes. El sistema contempla una administración completa de las actividades de los recursos humanos en cada proyecto, esto nos permite controlar al mismo tiempo los gastos y la proyección de futuros proyectos. Operando en un ambiente Web permite que desde cualquier lugar en donde se tenga una conexión a Internet y un navegador, es posible registrar entradas y salidas de los recursos humanos que se encuentran trabajando en los diferentes proyectos. Para las personas dedicadas a controlar los tiempos y actividades también les facilita el observar los avances y registros de asistencia desde el lugar en el que se encuentren sin necesidad de trasladarse hasta el lugar en donde labora cada persona, además ahorra tiempo en envíos de documentación que respalde la asistencia y los tiempos dedicados a las actividades asignadas a cada persona. Utilizando los recursos que nos brinda Internet y la buena planeación de nuestros proyectos y actividades podemos obtener beneficios inmediatos en costos y tiempos los cuales se reflejan en ganancias para la empresa y todo su personal.

Actualmente en el mercado existen muchas herramientas dedicadas a la planeación de Proyectos y sus recursos, sin embargo pocas son las dedicadas en darle seguimiento tomando información directamente de las personas dedicadas al desarrollo de las actividades, el llevar esta herramienta directamente a estas personas nos proporciona información inmediata de los avances de cada actividad que forman parte de un proyecto, además de proporcionar información de horas de entradas y salidas de cada persona. Con esta información es posible hacer planes más precisos y reales de los avances de proyectos al igual de los retrasos que se puedan presentar, para tal caso es posible tomar medidas antes de que se generen problemas, anticipando soluciones que eviten que nuestros proyectos se salgan de control. El incluir la herramienta en ambientes Web nos permite llevarla a cualquier lugar en donde se tenga una conexión a Internet.

El sistema refleja la información proporcionada por los usuarios de forma inmediata a la base de datos centralizada, con lo cual los responsables de los proyectos pueden conocer en tiempo real la situación de cada actividad asignada a cada proyecto. Estos medios permiten llevar al día avances de proyectos permitiendo con esto tomar decisiones anticipadas para cualquier contingencia que se pueda presentar, apoyando con esto la entrega oportuna de las diferentes actividades que incluyen los proyectos. Toda esta información brinda a la empresa la oportunidad de tener mejores resultados en la

planeación y entrega lo cual se ve reflejado en clientes satisfechos y por consecuencia el crecimiento de la empresa.

La participación en la elaboración de este trabajo permitió conocer herramientas de desarrollo de sistemas en plataformas Web mismas que hoy en día se están aplicando en todas las empresas y que se ha vuelto uno de los recursos con mayor uso a nivel mundial, de igual forma se logró incursionar en el proceso completo de un desarrollo de sistemas.

Las experiencias obtenidas al desarrollar este sistema han sido de valiosa utilidad pues han permitido abrir nuevas formas de operar de los sistemas que actualmente se manejan en la empresa, los cuales siguen trabajando bajo esquemas propietarios y herramientas totalmente cerradas a uso interno.

La experiencia obtenida al realizar el desarrollo de este sistema nos ha introducido en ambientes cliente/servidor al igual que en las herramientas y el uso de Internet no solo para realizar páginas de propaganda, también darle un uso aplicativo y propio para el desarrollo de empresas a nivel operativo.

BIBLIOGRAFÍA

FUNDAMENTOS DE BASES DE DATOS, Henry F. Korth, Abraham Silberschatz, McGraw-Hill, México 1993

FUNDAMENTOS Y MODELOS DE BASES DE DATOS, Adoración de Miguel, Mario Piattini, Alfaomega Grupo Editor, Colombia 1998

DISEÑO Y GESTIÓN DE SISTEMAS DE BASES DE DATOS, Angel Lucas Gomez, Paloma Romera García, Editorial Paraninfo sa, España 1993

INTERNET LIBRO DEL VAVEGANTE, Carballar, José A., Editorial: RA-MA, España 2000

ACTIVE SERVER PAGES 3.0 con ejemplos Reselman Bob, 1a ed. Buenos Aires: Prentice Hall, 2000 600p

REFERENCIA SITIOS INTERNET

<http://www.timepanic.com>

<http://www.alexcorp.com>

<http://www.timesheetelite.com>

<http://office.microsoft.com/home/default.aspx>

<http://programacion.com>

<http://es.tldp.org>

<http://www.ucm.es/info/Psyap/Prieto/alum9798/intranet01/server.htm>

http://www.osmosislatina.com/aplicaciones/servidor_web.htm

<http://www.microsoft.com/catalog>

<http://www.studiodeluxe.net/pws/index.htm>

<http://www.tergestesoft.com/~eddysworld/pws.htm>

http://www.cema.edu.ar/~asilvestri/Etapa_de_Pruebas.doc

<http://www.studiodeluxe.net/pws/index.htm>,

<http://www.pcmag.com/article2/0,4149,84182,00.asp>

http://training.gbdirect.co.uk/courses/perl/comparison_php_versus_perl_vs_asp_jsp_vs_vb_script_web_scripting.html

<http://mitglied.lycos.de/harueckner/hardware/>

Agradecimiento especial.

Agradecemos al PAT todo el apoyo proporcionado para el desarrollo de este proyecto.

Para obtener nuestra titulación el PAT siempre nos ha respaldado con asesores comprometidos a tener una tesis de calidad y con aplicación a la situación actual que se vive a nivel mundial. Con la experiencia obtenida a través del PAT queremos puntualizar su importancia en la formación de Ingenieros titulados, al igual agradecemos el apoyo incondicional que nuestra directora de tesis nos a brindado.