



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA



FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

TESIS

SISTEMA DE ADMINISTRACION Y
RENDIMIENTO DE ACTIVIDADES
EN UN AMBIENTE DE TROQUELES

PARA OBTENER EL TITULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

PEDRO LÓPEZ CASIQUE

ASESOR:

DR. IVO H. PINEDA TORRES

Puebla, Puebla, Noviembre 2008

DEDICATORIA

Dedico este trabajo a toda mi familia.

Por la enorme paciencia en mis años de escuela, su entendimiento en mis años de trabajo, por cada uno de sus sacrificios, por su herencia: mi educación; a mis padres Pedro y Rosalvia. Y aunque veíamos muy lejos este día, al fin después de muchos pero muchos meses hoy si les puedo decir que terminé mi tesis.

A mis dos hermanos, mi cuñada y sobrina, Héctor, Azucena, Kenny y Andrea, que en muchos momentos me daban esos momentos de empuje, más bien de presión para que terminara.

A mi novia Zuri su apoyo para retomar, continuar y terminar la tesis, y claro por sacrificar su tiempo para mi.

AGRADECIMIENTOS

Para aquellas personas tan importantes que estuvieron y aun siguen estando junto a mi, por sus buenos deseos y consejos.

Al Doc. Ivo por aceptar ser mi tutor de tesis, por tener esos minutos para revisar mis avances y mejor aún, por corregirme y guiarme al final de este trabajo.

A cada uno de ustedes, GRACIAS.

INDICE

INTRODUCCION	1
CAPITULO I. MARCO TEORICO	2
1.1. Un poco de Historia	3
1.2. Internet	3
1.3. World Wide Web	4
1.3.1. Antecedentes Históricos	4
1.3.2. Características	4
1.3.3. HTTP	5
1.3.4. HTML	5
1.3.5. CSS	6
1.4. Modelo Cliente / Servidor	6
1.5. Ingeniería Web	8
1.5.1. Características de las WebApps	8
1.5.2. Proceso de la IWeb	8
1.6. Servidores Web	9
1.6.1. Apache HTTP Server	10
1.7. Lenguajes de Programación Web	10
1.7.1. JavaScript	11
1.7.2. PHP	11
1.8. Bases de Datos	13
1.8.1. Sistema Administrador de Bases de Datos (DBMS)	14
1.8.2. Modelos de Datos	15
1.8.2.1. Modelo Jerárquico	15
1.8.2.2. Modelo de Red	16
1.8.2.3. Modelo Relacional	16
1.8.3. Normalización	16
1.8.4. MySQL	17
CAPITULO II. ANALISIS DEL SISTEMA	19
2.1. Planteamiento del Problema	20
2.2. Análisis del Sistema	20
2.2.1. Especificación de Requerimientos	20
2.2.2. Diagrama de Flujo de Datos	22

CAPITULO III. DISEÑO DEL SISTEMA	26
3.1. Modelo Funcional	27
3.2. Modelo Entidad-Relación	28
3.3. Diseño Lógico de la Base de Datos	28
3.4. Normalización	31
3.5. Diccionario de Datos	34
3.6. Criterio de Diseño	36
3.7. Herramientas de Implementación	36
CAPITULO IV. RESULTADOS Y ANALISIS	38
4.1. Pruebas y Su Origen	39
4.2. Tipos de Pruebas con Información Real	39
4.3. Interfaces del Sistema	40
4.4. Análisis de Resultados	61
Conclusiones	62
Bibliografía	63

INTRODUCCION

La industria de hoy en día requiere de sistemas computacionales que no solo ayuden a facilitar la labor diaria, si no también al almacenamiento de la información de forma segura. Las grandes cantidades de información en papel hacen una tarea tediosa a la hora de consultar cualquier dato, esto aunado a la falta de integridad de la información permite que se busquen nuevas herramientas que atiendan a todas las necesidades del usuario como el obtener la información de forma rápida, confiable y sobretodo de forma integra.

Muchas son las tecnologías aplicadas en el desarrollo de software, en la mayoría de los casos se trata de un software monousuario, limitando muchas funciones y tareas. La arquitectura Cliente-Servidor es una tecnología que se implementa de forma exitosa a las aplicaciones Web que no son muy comunes en las industrias, instituciones de gobierno o en instituciones publicas, sin embargo a diferencia de las aplicaciones monousuario, las aplicaciones Web pueden ser bien aprovechadas en estos organismos, y sobretodo tomando en cuenta que la mayoría de estas aplicaciones cuentan con una base de datos la cual da un gran soporte al ser un medio de almacenamiento de datos en grandes cantidades.

El objetivo de este proyecto de investigación es brindar una herramienta que de soporte a las tareas de la industria en general, mediante una base de datos relacional permite al usuario final mantener una administración de las actividades a desarrollarse de forma diaria por sus empleados, así mismo se establecen métricas de rendimiento de efectividad del personal para la toma de decisiones en los diferentes proyectos.

Este documento se encuentra compuesto por cuatro capítulos descritos a continuación:

Capitulo I. Marco Teórico. Se dan a conocer los fundamentos para todo aquel lector interesado en la lectura y comprensión de este documento, así como conceptos muy específicos para el desarrollo de este proyecto de tesis.

Capitulo II. Análisis del Sistema. Se describe el análisis de requerimientos de información, y se muestran los diagramas que ayudan a una mejor comprensión de los componentes del sistema en forma general y la comunicación que hay entre dichos componentes.

Capitulo III. Diseño del Sistema. Se considera el modelo Entidad-Relación existente en el sistema, así como la normalización resultante para evitar la redundancia de datos, y por último se muestra la estructura final de la base de datos.

Capitulo IV. Resultados y Análisis. Se describen las pruebas realizadas al sistema con información real, y una serie de figuras que ilustran el resultado de la implementación de la aplicación Web, con lo cual se muestran los resultados obtenidos al codificar los modelos provenientes del análisis y diseño.

Conclusiones.

Bibliografía.

CAPITULO I

MARCO TEORICO

1.1. UN POCO DE HISTORIA

Durante el desarrollo de Internet, se ha producido una evolución en los servicios que la red ha ido ofreciendo, así como en las herramientas de software. El uso más común que se le ha dado a Internet es el de publicación de información, y es ahí donde las facilidades de software se están dando; El servicio de FTP y sobre todo la WWW, son los servicios más utilizados, así como las páginas Web que se envían entre los servidores WWW y los navegadores de los clientes mediante el protocolo HTTP. El desarrollo y mantenimiento de páginas Web es uno de los servicios más utilizados en el ámbito de Internet, debido a sus fines comerciales. En un principio, las páginas Web eran estáticas, es decir, que el único proceso realizado era el de visualización de sus contenidos. En el momento en que se tiene la necesidad de que haya una mayor interacción entre los usuarios y los sistemas Web, surge el hecho de unificar y procesar las peticiones del cliente con el fin de ofrecerle mejor información.

La capacidad de procesamiento que sustentan las páginas Web dinámicas se puede llevar a cabo siguiendo alguno de los siguientes modelos:

- Páginas activas en el cliente: el procesamiento se lleva a cabo en la máquina del usuario.
- Páginas activas en el servidor: el procesamiento en el equipo donde reside el servidor de Web.
- Mezcla de páginas activas en el servidor y en el cliente: procesamiento mixto.

La ventaja de las páginas activas en el lado del cliente es la descarga de trabajo del equipo servidor, puesto que el procesamiento se lleva a cabo en la máquina del cliente. Además se hace un mejor aprovechamiento de la comunicación, ya que se minimiza el tráfico de datos entre cliente y servidor, al evitar continuos trasposos de información. No obstante, el uso de páginas activas en el servidor se hace imprescindible en aquellas ocasiones en las que la información debe encontrarse centralizada. Lenguajes de programación como PHP y ASP permiten la realización de páginas Web activas en el lado del servidor, a la vez que permiten ejecutarse con páginas activas en el cliente, sin mencionar que el acceso a las bases de datos es sencillo.

1.2. INTERNET

Desde su comienzo, a mediados de los años 70's, se ha convertido en un servidor de más de 20 millones de usuarios.

Uno de los protocolos mas conocidos para la conexión a Internet es llamado TCP/IP (Transfer Control Protocol / Internet Protocol), este es el que se encarga de entre otras funciones de contabilizar y transmitir los datos entre computadoras que se encuentren en red. Una red existe cuando 2 o más computadoras se encuentran conectadas de forma que se puedan compartir datos o recursos. Cada una de estas computadoras se denomina nodo de red, y además si una de ellas proporciona un servicio específico como verificación de contraseñas o administración de servicios es denominado servidor.

Cada nodo de red es identificado por un número IP único, este numero esta formado por cuatro octetos tomando la siguiente forma: *xxx.xxx.xxx.xxx* donde cada octeto *xxx* representa un número decimal entre 0 y 255, por ejemplo 168.101.122.1, es así como se identifican todas las computadoras en la red.

1.3. WORLD WIDE WEB

Debido a la gran generación de la información que se genera, con la utilización de las redes de computadoras se ha creado un modelo de manejo de datos llamado Cliente-Servidor, el cual es un paso previo a los sistemas Web.

Para solucionar las dificultades de acceso a la información, que se tenían en ese entonces, un grupo de investigadores del CERN (Laboratorio de Física de Partículas de Ginebra) ideó un nuevo servicio de información junto con un sistema para estructurar y enlazar tipos muy variados de datos. Su nombre, el *World Wide Web* “la telaraña mundial”.

1.3.1. Antecedentes Históricos

Fue en 1989, cuando Tim Berners-Lee empezó a diseñar un sistema para facilitar el acceso a la información del CERN. Dicho sistema empleaba el hipertexto para estructurar una red de enlaces entre los documentos. Una vez obtenida la aprobación para continuar el proyecto, nació el primer navegador Web, llamado World-WideWeb (sin espacios). En 1992 el sistema ya se había extendido fuera del CERN. El número de servidores “estables” había aumentado. En 1993 la Web ya era merecedora de un espacio en el New York Times. Éste es el año del lanzamiento de Mosaic, un navegador para X-Window/ Uníx que con el tiempo se convertiría en Netscape y que fue un factor clave de popularización de la Web. En 1994 se fundó el WWW Consortium, que se convertiría en el motor de desarrollo de los estándares predominantes en la Web (<http://www.w3c.org>). A partir de ese momento, el crecimiento ya fue constante, convirtiéndose hacia finales de los noventa en el servicio representante de Internet y dando lugar al crecimiento imparable de los servicios actuales.

1.3.2. Características

La Web se basa en dos puntales fundamentales: el protocolo HTTP (*HyperText Transfer Protocol*) y el lenguaje HTML (*Hypertext Markup Language*). El primero de ellos permite una implementación simple y sencilla de un sistema de comunicaciones que nos permite enviar cualquier tipo de información y ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones y reduzcan los costes de despliegue. El otro proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple.

1.3.3. HTTP

Es el protocolo base de la WWW. Es un protocolo simple, orientado a la conexión. La razón de que esté orientado a la conexión es porque emplea para su funcionamiento un protocolo de comunicaciones (TCP, Transfer Control Protocol) de modo conectado, un protocolo que establece un canal de comunicaciones entre el cliente y el servidor, y por el que pasa el flujo de bytes que constituyen los datos que haya que transferir, de forma contraria existen los protocolos no orientados a la conexión que dividen los datos en pequeños paquetes (datagramas) y los envían, pudiendo llegar por vías diferentes del servidor al cliente. Durante cada transferencia de datos, la conexión se independiza de la anterior, sin relación alguna entre ellas, hasta el punto de que para

transferir una página Web se tiene que enviar el código HTML del texto, así como las imágenes que la componen. Existe una variante de HTTP llamada HTTPS (S por Secure) que utiliza el protocolo de seguridad SSL (Secure Socket Layer) para cifrar y autenticar el tráfico entre cliente y servidor, siendo ésta muy usada por los servidores Web de comercio electrónico, así como por aquellos que contienen información personal o confidencial.

Cabe destacar que todos los recursos proporcionados mediante HTTP deberán ser referenciados mediante una URL (Universal Resource Locators).

1.3.4. HTML

Una página Web es un documento multimedia, porque está constituida por texto, imágenes, sonido, animaciones, etc., pero, también es un documento en el que se pueden añadir enlaces a otras páginas y secciones, convirtiéndose así en documentos hiper-media. Las páginas Web están escritas en un lenguaje de denominado HTML (HyperText Markup Language). Para visualizar la página Web, es necesario la existencia de un “navegador”, este es el encargado de tomar el documento en formato HTML e interpretarlo, para mostrar al usuario el resultado final. HTML fue desarrollado por Tim Barners-Lee en conjunto con un grupo de científicos del Centro Europeo de Investigación Nuclear (CERN) en 1992.

Un documento HTML comienza y termina con la marca HTML. Esta marca indica que se trata de un documento HTML, para que no se confunda con otro tipo de lenguajes de marcas, que son similares. Un documento HTML consta de una cabecera (se usa la marca HEAD) y del cuerpo (se usa la marca BODY). Mientras que la cabecera incluye información asociada a algún documento, el cuerpo incluye la información que constituye al propio documento.

<HTML>

<HEAD>

Cabecera del documento (titulo, metainformación, estilos, etc.)

</HEAD>

<BODY>

Cuerpo del documento (texto, imágenes, etc.)

</BODY>

</HTML>

La cabecera es el primer elemento de un documento. En ella se encuentran otros elementos, como el título del documento: <TITLE> titulo documento </TITLE>. Otro elemento que podemos encontrar en la cabecera es el elemento META, este elemento permite añadir meta-información del documento, como puede ser el autor, una breve descripción del documento, etc. El elemento STYLE, si se usa dentro de la cabecera, permite asociar una hoja de estilo al documento.

Aunque HTML es una buena opción para presentar la información en Web, también tiene algunas restricciones, como por ejemplo que no es suficiente para crear documentos de tipo interactivo y multimedia que se requieren hoy en día. Una vez que la pagina Web es “cargada” por el navegador, esta permanece estática, no es posible modificar su contenido. Para resolver este tipo de problemas, surge la necesidad de

crear nuevos lenguajes que puedan trabajar de forma dinámica, como PHP, ASP, entre otros. Este tipo de lenguajes trabajan con archivos denominados *scripts*, los cuales contienen el código que interpreta el navegador para realizar una tarea específica.

1.3.5. CSS

CSS (Cascading Style Sheets) es un lenguaje que se usa para la definición de la apariencia visual que pueden tener los elementos de un documento HTML. Está basado, al igual que en el caso de las marcas HTML, en un conjunto de atributos y valores que definen las propiedades de presentación. Una definición en CSS consta de una lista de pares atributo-valor separados por punto y coma (;). El atributo se separa del valor por medio de dos puntos (:):

atributo1: valor1; atributo2: valor2;...; atributoN: valorN

CSS ofrece atributos para casi cualquier elemento:

- Fuentes, tanto en tipo ó tamaño.
- Colores, de los caracteres, del fondo de la página, de los enlaces, etc.
- Alineación entre los diferentes elementos del documento: texto, imágenes, tablas, etc.
- Bordes y marcos para la representación de objetos.
- Sangrados e interlineados de párrafos.

Cualquier marca HTML puede ir acompañada de una especificación de estilo usando CSS por medio del atributo STYLE. Por ejemplo, si se quiere que el tamaño de la fuente de un párrafo sea de 12 puntos y que el color del texto sea rojo se puede usar la siguiente marca:

<P STYLE="font-size: 12pt; color: red">

- STYLE admite como parámetro una lista de especificaciones CSS. La verdadera potencia de las especificaciones de estilo aparece cuando se usa la marca de estilo en la cabecera del documento.

1.4. MODELO CLIENTE / SERVIDOR

Se trata de un modelo para el desarrollo de sistemas de información, en la cual los procesos se dividen en procesos independientes que cooperan entre si para intercambiar información, servicios o recursos.

Se llama *Cliente* al que inicia la solicitud, y *Servidor* al proceso que responde dichas solicitudes; la comunicación en este modelo se da cuando el cliente envía una solicitud a través de la red hacia el servidor, este realiza el trabajo que le es pedido o busca la información solicitada para después devolver una respuesta y desplegarla.

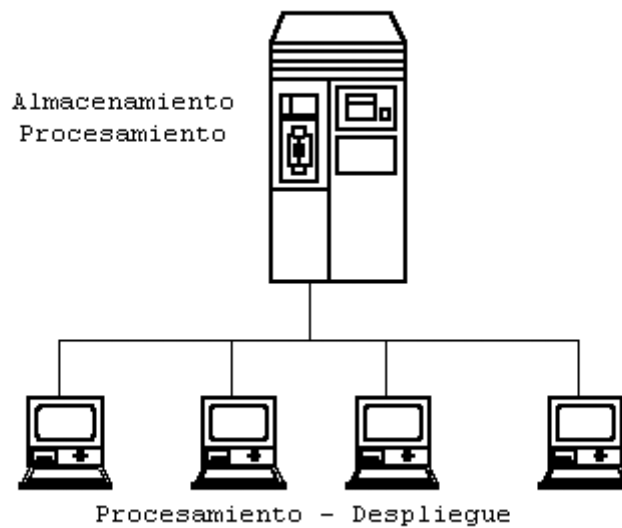


Figura 1. Estructura de un modelo Cliente / Servidor.

El esquema Cliente-Servidor facilita la integración entre sistemas diferentes y comparte información, permitiendo, que las máquinas ya existentes puedan ser utilizadas pero usando interfaces mas amigables para el usuario

Al favorecer el uso de interfaces gráficas más amigables, los sistemas bajo este esquema tienen una interacción más intuitiva con el usuario. Una ventaja que presenta este modelo con respecto a uno centralizado, es que no siempre es necesario transmitir información gráfica por la red ya que esta puede estar en el cliente, y con lo cual se aprovecha mejor el ancho de banda de la red.

VENTAJAS

Las ventajas sobre sistemas centralizados, en un modelo Cliente-Servidor, son varias, por ejemplo:

- El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. El cliente se conecta al servidor solo cuando es necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre para otra conexión.
- El tiempo de desarrollo de aplicaciones son mucho más cortos.
- El Cliente no necesita conocer la forma de trabajo del Servidor, solo necesita su interfaz externa.

1.5. INGENIERIA WEB

En 1998, Roger Pressman moderó una mesa redonda virtual con representantes de la Ingeniería del Software tradicional y del desarrollo software basado exclusivamente en Internet, donde el debate se centró en discutir si valía la pena aplicar un proceso de ingeniería a las aplicaciones con base en Internet, o qué características tenían éstas que justificaran el no utilizarlo. La conclusión general fue que aplicar un proceso de ingeniería nunca es una mala idea pero que éste debería adaptarse a los requerimientos de cambio continuo y rapidez siempre presentes en el proceso de desarrollo Web.

Los sistemas y aplicaciones basadas en Web (WebApps), hacen posible que los usuarios finales dispongan de un gran contenido y una mejor funcionalidad. La Ingeniería Web (WebE) no es una copia de la Ingeniería del Software (IS), solo toma muchos conceptos y principios básicos de la IS, dando una gran importancia a las mismas actividades técnicas de gestión.

1.5.1. Características de las WebApps

Las características con las que se puede mantener la continua evolución y el proceso de desarrollo de las WebApps son:

- **Inmediatez:** Comercializar un sitio Web puede lograrse en días o semanas, es por eso que se recomiendan usar métodos de planificación, análisis, diseño, implementación y comprobación, las cuales deben adaptarse a los tiempos estimados de forma real en el desarrollo de estas WebApps.
- **Seguridad:** El contenido es confidencial y debe proporcionar formas seguras de transmisión de la información, es por ello que se deben implementar fuertes medidas de seguridad en toda la infraestructura que da soporte a la WebApp y por supuesto dentro de la misma aplicación.
- **Estética:** Tanto la apariencia como la interacción debe ser atractivo, e intuitiva para el usuario final, este es un punto importante el todo desarrollo.

1.5.2. Proceso de la WebE

El proceso de la Ingeniería Web (WebE) comienza con la formulación del problema que identifica claramente las metas y los objetivos para resolver las WebApps, posteriormente entra el proceso de planificación en la cual se estima el coste global del proyecto, evalúa los riesgos asociados con el desarrollo y define una programación temporal del desarrollo; el tercer proceso es el análisis, aquí se establecen los requisitos técnicos y se identifican los objetos del contenido que se incorporarán, posteriormente se diseña la interfaz arquitectónica y del navegador. Por último el sistema se implementa utilizando lenguajes y herramientas especializados en la Web, y finaliza con las pruebas.

La WebE hace uso de un modelo de proceso iterativo e incremental porque la línea de tiempo de desarrollo es muy corta para la WebApp.

1.6. SERVIDORES WEB

Un servidor es un tipo de software que proporciona ciertos servicios, ya sea hacia los usuarios o hacia terminales que lo solicitan. Por ejemplo, en el modelo Cliente-Servidor, el cliente podría ser una computadora que realiza solicitudes de información a través de un programa de correo (Outlook Express por ejemplo) y, el servidor le entrega los datos en forma de correos electrónicos en respuesta a su solicitud. Ahora bien, un Servidor Web es un programa que proporciona datos en forma de páginas Web, hipertextos o páginas HTML; textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio del protocolo HTTP. Con esto, un Servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que comúnmente se conoce como navegador ó navegador Web.

Por ejemplo, al teclear `http://www.nickma.net` en un navegador, éste realizará una petición HTTP al servidor que tiene asociada dicha url, aquí el servidor responde al cliente enviando el código HTML de la página; el navegador cuando recibe el código, lo interpreta y lo muestra en pantalla.

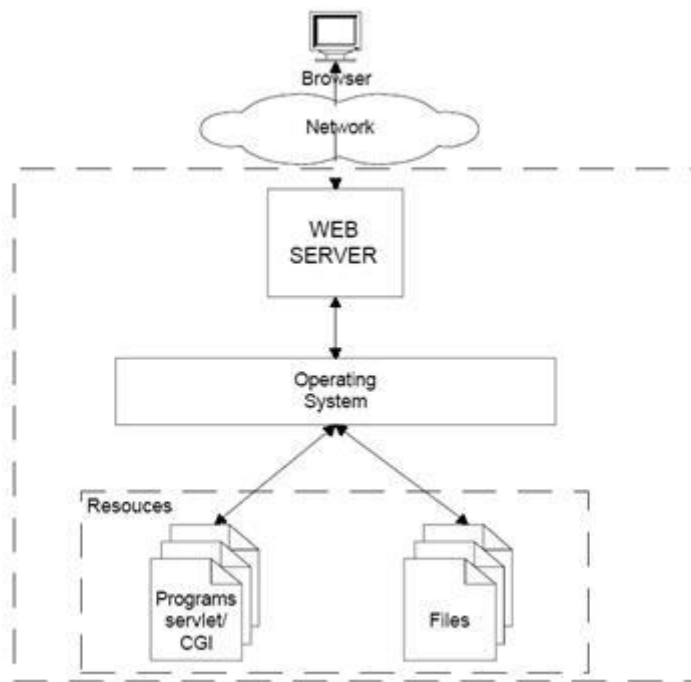


Figura2. El Servidor es el responsable de proporcionar el acceso a los recursos solicitados que están bajo el control del sistema operativo.

Aquí se muestran algunos Servidores Web como ejemplo:

- AOLserver
- **Apache**
- Java Server
- Microsoft Internet Information Services
- Samba Server
- Xitami

1.6.1. Apache HTTP Server

El servidor HTTP Apache es un servidor Web de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows y otras, que implementa el protocolo HTTP. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches aplicados al servidor de NCSA. Era, en inglés, “A Patchy Server” (Un Servidor Parcheado).

Apache se desarrolla dentro del proyecto HTTP Server (HTTPD) de la Apache Software Foundation. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red, en el 2005 Apache era el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo.

Apache es reconocido en muchos ámbitos empresariales y tecnológicos, esto se debe a varias razones, por ejemplo:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante. Esto le da una transparencia a este software de manera que si se requiere ver que es lo que se esta instalando como servidor, se puede saber, sin ningún secreto.
- Es un servidor altamente configurable y muy sencillo ampliar las sus capacidades. Actualmente existen muchos módulos para Apache que son adaptables a este, y se encuentran a disposición para su instalación.
- Trabaja con Perl, PHP y otros lenguajes de script.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

1.7. LENGUAJES DE PROGRAMACION WEB

Un lenguaje de programación permite interpretar las instrucciones que han de ser ejecutadas en una computadora, es un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático. Se puede definir sobre que datos deben de operar las instrucciones, como deben ser almacenados y transmitidos.

Cualquier programa debe ser compilado, o ser interpretado para que pueda ser ejecutado por el ordenador. También existen lenguajes de scripting que son ejecutados a través de un intérprete y no necesitan compilación.

1.7.1. JavaScript

Inicialmente se llamó Mocha, luego LiveScript y finalmente fue renombrado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995. JavaScript es un lenguaje orientado a las páginas Web, su sintaxis es semejante a la del lenguaje Java pero de una forma más sencilla.

Aporta características dinámicas como datos, variables, modos de acceso, interactividad con el usuario, personalización, etc., todo esto es sobre las páginas Web escritas en HTML. Se pueden conseguir interesantes efectos en las aplicaciones Web, como por ejemplo comprobar la validez de la entrada de formularios, abrir y cerrar ventanas (popup), cambiar dinámicamente el aspecto y los contenidos de una página, cálculos matemáticos sencillos y mucho más.

JavaScript permite crear aplicaciones específicamente orientadas a su funcionamiento en la red Internet. Usando JavaScript, se pueden crear páginas HTML dinámicas que procesen la entrada del usuario y que sean capaces de gestionar datos persistentes usando objetos especiales, archivos y bases de datos relacionales.

Aquí se muestra como se integra código JavaScript en un documento HTML:

```
<HTML>
<HEAD>
<TITLE>Ejemplo de JavaScript</TITLE>
</HEAD>
<BODY>
<b>Esto es texto normal de un documento HTML</b>
<p>

<SCRIPT LANGUAGE="JavaScript">
  document.write("Texto generado desde JavaScript")
</SCRIPT>

<p>
<b>Esto es, de nuevo, HTML</B>
</ BODY >
</HTML>
```

1.7.2. PHP

PHP se ha concretado como uno de los más importantes lenguajes de programación en entornos Web. Mucha de su sintaxis es tomada de C, Java y Perl con un par de características adicionales únicas y específicas de PHP. El propósito del lenguaje es permitir que los desarrolladores Web escriban páginas generadas dinámicamente con rapidez. PHP significa PHP: Hypertext Preprocessor (PHP: Pre-

procesador de Hipertexto). La primera palabra del acrónimo es el acrónimo mismo, a este tipo de acrónimo es llamado recursivo.

PHP suele ser utilizado conjuntamente con Perl, Apache, MySQL o PostgreSQL en sistemas Linux, formando una combinación barata, potente y versátil. Se reconoce como LAMP (formado por las iniciales de los diversos productos). Es un lenguaje que opera del lado del servidor, que se utiliza básicamente para generar páginas dinámicas y que facilita de forma sencilla el acceso a diferentes bases de datos.

Las diferencias en el uso de PHP frente a otros lenguajes de scripting que trabajan en el lado del servidor, y en concreto sobre el ASP (Active Server Pages) de Microsoft, son:

- Libre y Abierto
- Multiplataforma
- Soporte para varios Servidores Web
- Fácil acceso a Bases de Datos
- Presenta una integración perfecta entre Apache-MySQL
- Seguro
- Además PHP está orientado a objetos

La manera en como funciona PHP es la siguiente: si solicitamos a nuestro servidor una página PHP, éste envía dicha página al intérprete de PHP que la ejecuta y devuelve el resultado (generalmente es código HTML) al servidor Web, el cual, a su vez, se lo enviará al cliente. Por ejemplo, se tiene una página PHP con el siguiente contenido:

```
<?php echo "<h1>¡Hola Pedro!</h1>"; ?>
```

Si este se encuentra en un fichero con extensión .php el servidor enviará la página al intérprete de PHP, el cual ejecuta la página y obtiene como resultado:

```
<h1>¡Hola Pedro!</h1>
```

Posteriormente el servidor se lo enviará al navegador del cliente que ha solicitado la página. El mensaje aparecerá en la pantalla de este último. PHP permite mezclar en la misma página HTML y PHP, lo que facilita notablemente el trabajo con éste

Un punto clave de PHP son las funciones. Las funciones en PHP pueden recibir o no, parámetros y siempre pueden devolver un valor. Las funciones sirven para dar mayor modularidad al código, evitando así la repetición de código, permitiendo el aprovechamiento de código entre proyectos. Se cuenta con dos mecanismos para que PHP muestre información al usuario: por un lado se pueden escribir páginas HTML corrientes, insertando sólo el código PHP que requerimos en medio del código HTML. Por otro lado, se puede usar PHP para generar contenido dinámico. Para ello se debe usar las instrucciones de PHP de salida de datos, la más importante, *echo*.

Para recoger información del usuario, se utilizan los formularios de HTML. Como PHP fue diseñado para crear aplicaciones Web, el acceso a los valores introducidos por el usuario en los campos del formulario es realmente fácil en PHP, ya que éste define un vector llamado REQUEST al que accedemos con el nombre del campo como índice y que contiene el valor contenido en éste al ejecutar el programa PHP. Además proporciona métodos para acceder a un gran número de sistemas de bases de datos (MySQL, PostgreSQL, Oracle, ODBC, etc.). Esta funcionalidad es imprescindible para el desarrollo de aplicaciones Web complejas.

PHP permite operar con la complejidad creciente de las aplicaciones Web, dando lugar entre otras cosas a:

- Control de sesiones.
- Control de identidad de usuarios.
- Plantillas HTML.
- Carritos de compra.
- Creación de imágenes dinámicamente.
- Manejo de *cookies*.
- Transferencia de ficheros.
- Múltiples protocolos de comunicaciones: HTTP, FTP, etc.
- Creación de ficheros PDF.
- Interfaces a multitud de bases de datos: Oracle, Sybase, etc.
- Expresiones regulares.

1.8. BASES DE DATOS

Un banco o base de datos es un conjunto de datos, que pertenecen al mismo contexto, almacenados de tal forma que se puedan usar posteriormente.

Una parte importante en el desarrollo de aplicaciones Web es la elección del Sistema Gestor de Base de Datos (DBMS por sus siglas en inglés) que se usará. En la actualidad hay bastantes DBMS de código libre, muchos de ellos del mismo nivel cualitativo que algunos de los DBMS comerciales más conocidos.

Hay cuatro componentes fundamentales contenidos en un Sistema de Base de Datos:

1. **DATOS.** Es toda aquella información contenida en una Base de Datos.
2. **HARDWARE.** Son los volúmenes de almacenamiento secundario, tales como cintas, discos, etc., y donde reside la Base de Datos.
3. **SOFTWARE.** Se refiere al DBMS la cual maneja todas las solicitudes de acceso a la Base de Datos formuladas por los usuarios.
4. **USUARIOS.** Aquella persona que requiere del uso del Sistema de Base de Datos para realizar cualquier acción permitida sobre ella.

1.8.1. Sistema Administrador de Bases de Datos (DBMS)

DBMS es un conjunto de programas que se encarga de manejar todos los accesos a la Base de Datos. Se compone de un Lenguaje de Definición de Datos (DDL: Data Definition Language), de un Lenguaje de Manipulación de Datos (DML: Data Definition Language), y de un Lenguaje de Consulta (SQL: Structured Query Language).

El primero de ellos, DDL, describe todas las estructuras de información y los programas que se usan para crear, actualizar e introducir la información que contiene una Base de Datos; DML manipula los datos de la Base de Datos, realiza consultas, inserciones, eliminaciones y modificaciones de datos. Y por último SQL es empleado por el usuario para extraer información de la Base de datos, también permite al usuario hacer solicitudes de datos sin tener que escribir algún programa, con tan solo usar instrucciones propias.

Existen diferentes objetivos que debe cumplir un DBMS, entre las cuales se mencionan las siguientes:

- Abstracción de la información.
- Independencia de datos.
- Redundancia mínima de información.
- Consistencia.
- Seguridad.
- Integridad.
- Respaldo y recuperación.
- Controlar la concurrencia.
- Minimizar el tiempo de respuesta a una solicitud

El objetivo de un DBMS es proporcionar un entorno conveniente y eficiente para ser utilizado al extraer, almacenar, y manipular cualquier tipo de información de la base de datos. Todas las peticiones de acceso a la base de datos, se manejan centralizadamente por medio del DBMS, por lo que este funciona como un tipo de interfaz entre los usuarios y la base de datos.

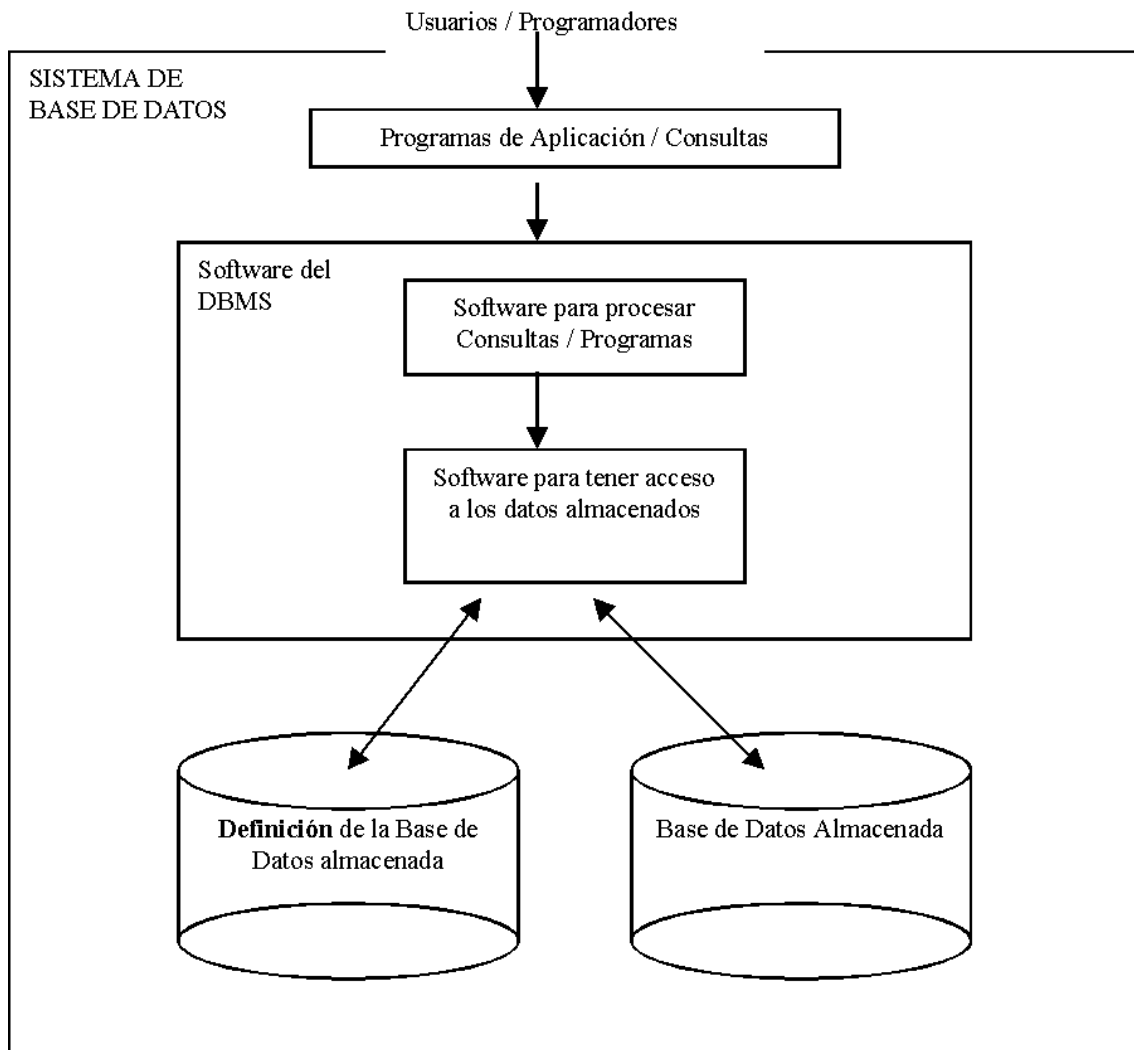


Figura 3. Imagen que muestra el esquema de un DBMS.

1.8.2. Modelos de Datos

Las bases de datos se pueden clasificar de acuerdo a su modelo de administración de datos. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

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

1.8.2.1 Modelo Jerárquico.

Los datos se organizan de forma similar a un árbol, en donde un nodo padre de información puede tener varios hijos. Este modelo es utilizado en aplicaciones que manejan gran cantidad de información.

1.8.2.2. Modelo de Red.

Aquí se permite que un mismo nodo tenga varios padres, ofrece una solución eficiente a la redundancia de datos, que en el caso anterior, el modelo jerárquico no cuenta.

1.8.2.3 Modelo Relacional.

Este es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. La idea fundamental es el uso de "relaciones", las cuales se consideran como conjuntos de datos llamados "tuplas". Las relaciones se dan entre tablas las cuales están compuestas por registros (filas de una tabla), y los campos (columnas de una tabla).

1.8.3. Normalización

Básicamente, es el proceso de organizar eficientemente los datos en una base de datos, mediante reglas aplicadas a las relaciones obtenidas. Las ventajas de la normalización son las siguientes:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Se han desarrollado una serie de guías para asegurarse que la base de datos esta normalizada, estas guías se conocen como Formas Normales: 1FN, 2FN y 3FN son las 3 formas normales para la normalización de una base de datos.

Un concepto fundamental dentro de la normalización es la dependencia funcional, esto es, matemáticamente hablando:

“un valor Y esta determinado por un valor X, y se expresa así: $Y = f(X)$ ”

Hablando con respecto a las relaciones de las bases de datos, una dependencia funcional se da entre los atributos A y B si el valor de A esta determinado por el valor de B, entonces se dice que A es funcionalmente dependiente de B, y se denota:

B \longrightarrow A

A continuación se proporciona la descripción de cada una de las formas normales:

Primera Forma Normal (1FN)	Las columnas repetidas deben eliminarse y colocarse en tablas separadas.
Segunda Forma Normal (2FN)	Todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas, donde una dependencia parcial describe aquellos datos que no dependen de la llave primaria de la tabla para identificarlos.
Tercera Forma Normal (3FN)	Las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas, donde una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen de otras columnas que tampoco son llave.

La normalización es una técnica que se utiliza para crear relaciones lógicas apropiadas entre tablas de una base de datos. Ayuda a prevenir errores lógicos en la manipulación de datos. La normalización facilita también agregar nuevas columnas sin romper el esquema actual ni las relaciones.

1.8.4. MySQL

MySQL es un DBMS, multihilo y multiusuario, es decir, soporta una gran carga de procesos de forma eficiente y también puede ser usada por varios usuarios, cabe destacar que esta desarrollado en su gran parte en ANSI C.

Existen varias API's (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, entre ellos se encuentra C, C++, C#, Pascal, Java , Lisp, Perl, PHP, Python, etc., cada uno de estos utiliza una API específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

MySQL es muy utilizado en aplicaciones Web, es muy común que se le ligue con PHP para el desarrollo de aplicaciones Web complejas. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor MyISAM.

Por otra parte, funciona sobre múltiples plataformas, incluyendo BSD, FreeBSD, GNU/Linux, Mac OS X, Novell Netware, OS/2 Warp, QNX, Solaris, Windows (incluyendo 95, 98 y 2000). Se trata de un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

Hay características distintivas y que son implementadas solo por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

CAPITULO II

ANALISIS DEL SISTEMA

2.1. PLANTEAMIENTO DEL PROBLEMA

El poseer información electrónica, y que se encuentre disponible para aquellas personas que les sea necesario y prescindible para terminar ciertas tareas o actividades, es de vital importancia en la mayoría de las organizaciones. Contar con un centro de información (por ejemplo una base de datos) donde un usuario pueda solicitar la información requerida para la elaboración de sus actividades facilita en gran medida el desarrollo de sus actividades, sin contar que se adquieren grandes beneficios, como por ejemplo, tener la información de forma íntegra, y en el momento que se solicite.

El origen del desarrollo de este software se da por la necesidad de obtener una herramienta que administre de forma eficiente la asignación de actividades para la elaboración de proyectos. Se automatizaron una serie de tareas que hoy en día se realizan de forma manual. La importancia de contar con un software de este tipo mejora a las empresas en el sentido de llevar un mejor control, no solo del rendimiento de cada una de las personas involucradas en un cierto proyecto sino también a tener un control de tiempos para la elaboración de los mismos.

2.2. ANALISIS DEL SISTEMA

De forma general, este software realiza la asignación de actividades a usuarios para la creación y seguimiento de proyectos, también lleva un control de rendimiento para cada uno de los usuarios relacionados con un proyecto en específico. En la creación de proyectos se involucran diferentes áreas o departamentos de la organización, estas áreas están a cargo por diferentes personas (jefes de departamento), las cuales son quienes se responsabilizan por los seguimientos de las actividades asignadas para un buen producto final.

Una parte importante en la creación de proyectos es la designación de los tiempos para su culminación, es por eso que se asignan 2 fechas (fechas primarias) en la creación de una nueva actividad, estas fechas permitirán en un futuro, determinar el nivel de rendimiento para cada uno de los usuarios, este rendimiento es calculado de forma interna aplicando diferencias de tiempos, y de acuerdo a una serie de condiciones el sistema mostrará en pantalla de una forma visual (por colores, explicado más adelante) el estatus en que se encuentra el desarrollo de “x” actividad.

2.2.1. Especificación de Requerimientos

Para brindar una herramienta eficaz, y confiable para la administración de actividades y rendimiento se han definido varios requerimientos que a continuación se describen.

La primera de ellas es la autenticación de usuarios, la cual permitirá brindar seguridad en el manejo de la información. Para poder acceder al sistema el usuario tendrá que *logearse* por medio de un nombre de usuario y una contraseña. Una vez dentro del sistema, y dependiendo el tipo de acceso (ó privilegio), el usuario podrá realizar cualquier acción permitida, como se muestra en la siguiente tabla:

	Administrador	Trabajador
Acceso al módulo de usuarios	✓	
Acceso al módulo de departamentos	✓	
Actualización de información de los usuarios	✓	
Eliminación de usuarios	✓	
Consulta de reportes de rendimiento de todos los usuarios	✓	
Consulta de reportes de rendimiento personales	✓	✓
Actualización de actividades	✓	✓
Eliminación de actividades	✓	

Una nueva asignación de actividades, por parte de un usuario administrador, lleva la siguiente secuencia:

- 1) El sistema de forma automática asigna la primera fecha (fecha de inicio), esta es la fecha en la cual da comienzo el desarrollo de la nueva actividad.
- 2) Posteriormente el usuario administrador asigna la segunda fecha (fecha de término), esta es una fecha de propuesta para el término de la actividad.
- 3) Luego se detalla una pequeña descripción de la tarea a realizar.
- 4) Por último se asigna hacia quien va dirigida esa actividad, es decir, elegir que departamento se hará cargo de dicha actividad.

Estas entradas de información producen como salida las actividades asignadas y sus tiempos en las cuales deben ser terminadas, todas estas actividades son mostradas en pantalla, las cuales están ordenadas de acuerdo a la ultima captura efectuada.

El administrador puede visualizar en pantalla una lista de todas las actividades que se encuentran en diferentes estatus de proceso, estos estatus son: tareas finalizadas, tareas en desarrollo y tareas recientemente asignadas. También se muestra el porcentaje de rendimiento de los usuarios con respecto a sus diferentes tareas asignadas.

La tercera fecha (fecha real de entrega), es el resultado de una serie de cálculos aplicados a las dos fechas anteriores, fecha de inicio y fecha de término, este cálculo se efectúa de forma interna en el sistema, la fecha de entrega es la que muestra el porcentaje de rendimiento real del usuario desempeñado en cada una de sus tareas. Este rendimiento se visualiza en un valor porcentual y de una forma mas llamativa para una fácil comprensión, se han asignado diferentes colores para mostrar así el rendimiento, un color verde si la tarea ha sido finalizada en la fecha acordada o antes, un color amarillo si la tarea ha sido finalizada a lo mas en dos días posteriores a la fecha acordada (a la fecha final), y un color rojo si la tarea tuvo un diferencia de 3 días ó más con respecto a la fecha de entrega.

Este tipo de “alertas” le permiten al jefe de proyecto poder tomar las medidas adecuadas y decisiones para que el proyecto se termine con éxito.

En la actualización de los registros se hacen ajustes en las asignaciones, pueden ser extensiones de tiempo, asignar la tarea a un diferente usuario ó simplemente cambiar la descripción de la actividad.

La eliminación de registros se encarga de borrar tareas, por ejemplo, que ya se hayan cumplido, esto hace que la base de datos no acumule registros innecesarios para el administrador y con ello se libere espacio para que el sistema haga sus procesos más rápido.

Hablando de aspectos no funcionales se puede hacer referencia a las interfaces, que en todo momento se presentan de una forma agradable y sobretodo de fácil uso para el usuario final, se maneja un menú para la navegación dentro del sistema, este menú siempre se muestra en una cierta posición para no desubicar al usuario. Este software se ha desarrollado con herramientas que permiten manipular la información de forma dinámica, lo cual rompe con los esquemas tradicionales de programación e interfaces tipo ventanas.

2.2.2. Diagrama de Flujo de Datos

La técnica de diagrama de flujo de datos, es una representación grafica que permite al analista definir entradas, procedimientos y salidas de la información, permitiendo así comprender los procedimientos existentes con la finalidad de optimizarlos, reflejándolos en el sistema propuesto.

El Diagrama de Flujo de Datos (DFD) tiene por objetivo representar gráficamente el sistema a nivel lógico y conceptual, ilustrando los componentes esenciales de un proceso y la forma en que interactúan.

Enseguida se muestra el DFD de nivel contextual para "Sistema de Administración y Rendimiento de Actividades (SARA)".

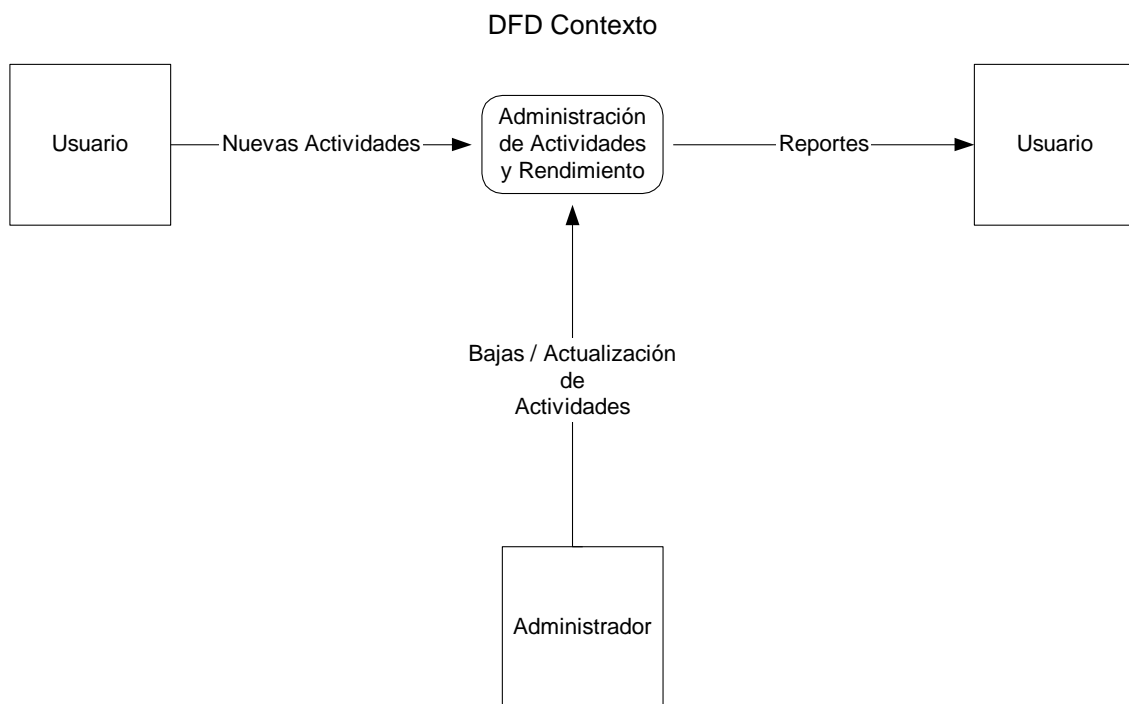


Figura 4. DFD Nivel Contextual.

El Diagrama de Flujo de Datos de Nivel 1 (DFD 1) identifica los distintos subsistemas que interactúan con el usuario.

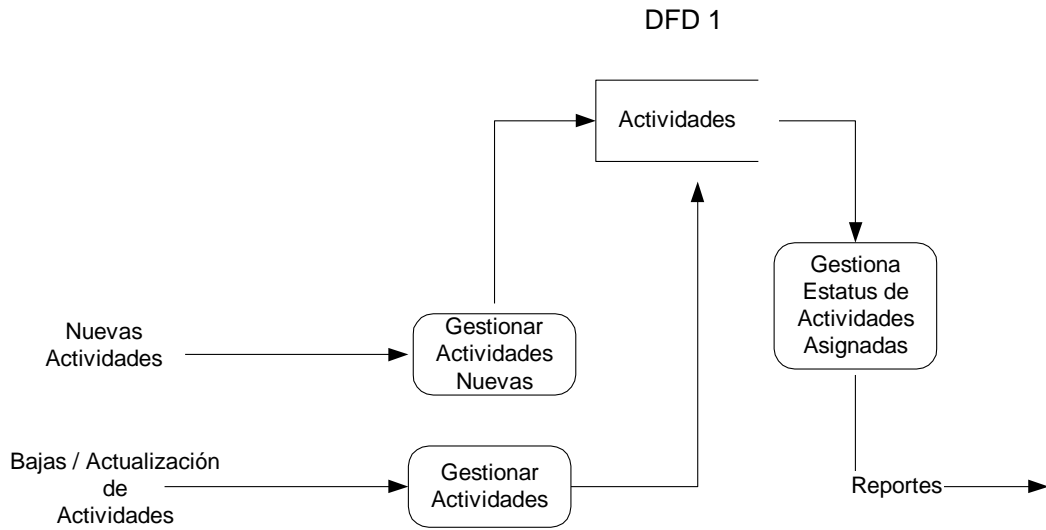


Figura 5. DFD Nivel 1. Subsistemas

A continuación se muestran los Diagramas de Flujo de Datos Nivel 2 (DFD 2), en los cuales se visualizan los diferentes métodos del sistema.

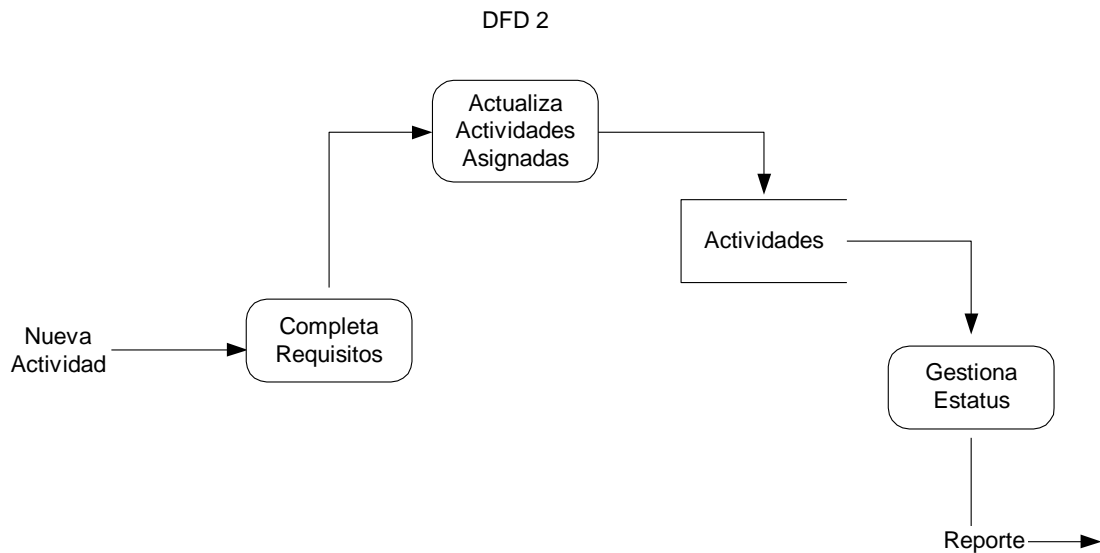


Figura 6. DFD Nivel 2. Nueva Actividad.

DFD 2

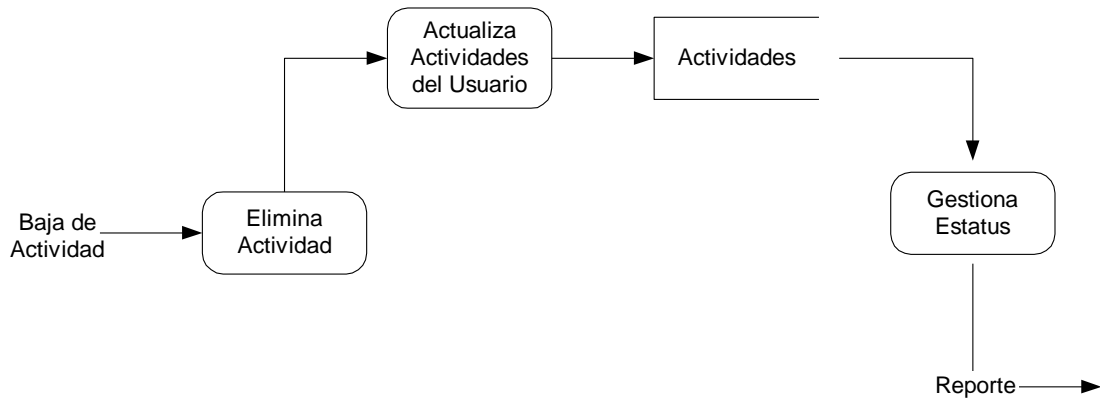


Figura 7. DFD Nivel 2. Elimina Actividad.

DFD 2

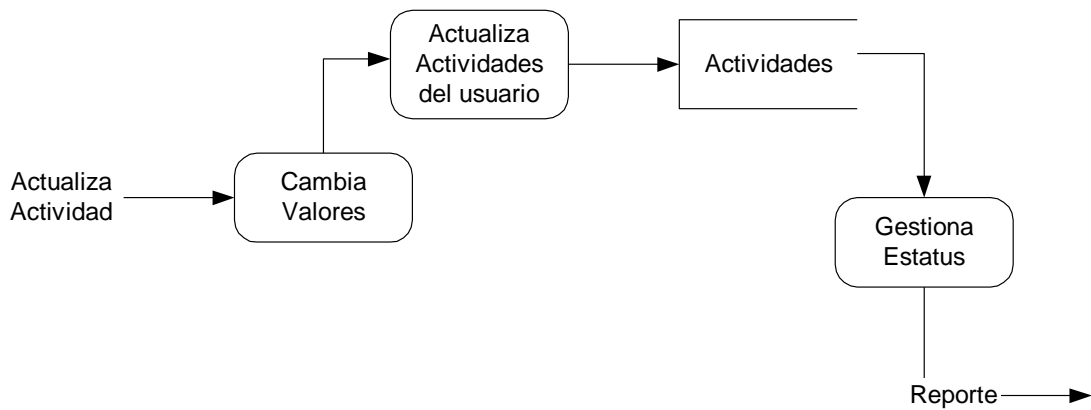


Figura 8. DFD Nivel 2. Actualiza Actividad.

Por último se ilustra el Diagrama de Flujo de Datos Nivel 3 (DFD 3)

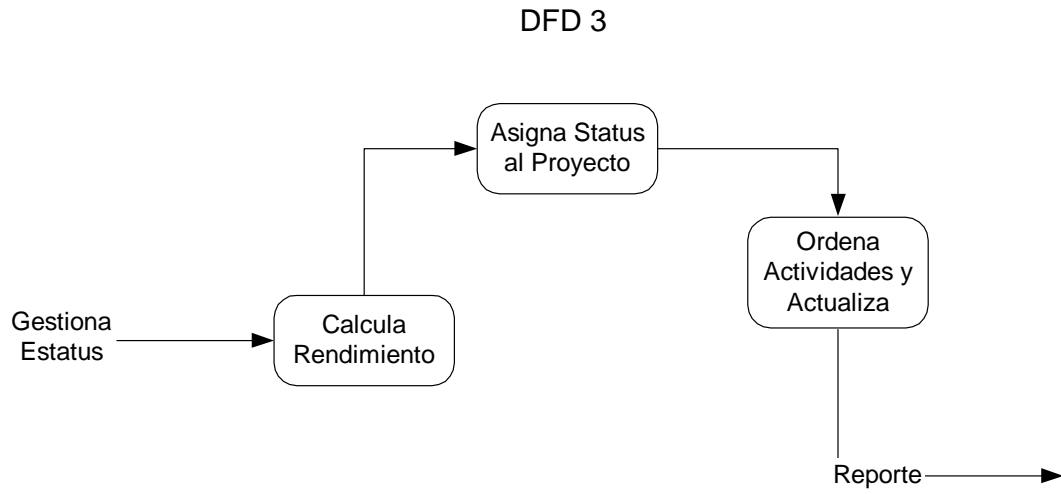


Figura 9. DFD Nivel 3. Gestiona Estatus de Rendimiento (subfunción asociada)

CAPITULO III

DISEÑO DEL SISTEMA

3.1. MODELO FUNCIONAL

El modelo funcional muestra como los valores de salida, que son producidos por el sistema, son derivados a partir de los valores de entrada.

Este sistema cuenta con un módulo de autenticación de usuarios que permite brindar seguridad en el manejo de la información, así como la libre operatividad del mismo sistema. Dicha autenticación se da por medio de un nombre de usuario y una contraseña.

SARA esta compuesto por 3 módulos:

1. MODULO USUARIOS.

- **Altas.** Captura de los usuarios pertenecientes al sistema.
- **Modificaciones.** Actualización de la información almacenada de cada uno de los usuarios.
- **Consulta.** Visualización a detalle de la información de un usuario en específico.
- **Bajas.** Eliminación del registro seleccionado.
- **Estatus.**

2. MODULO ACTIVIDADES.

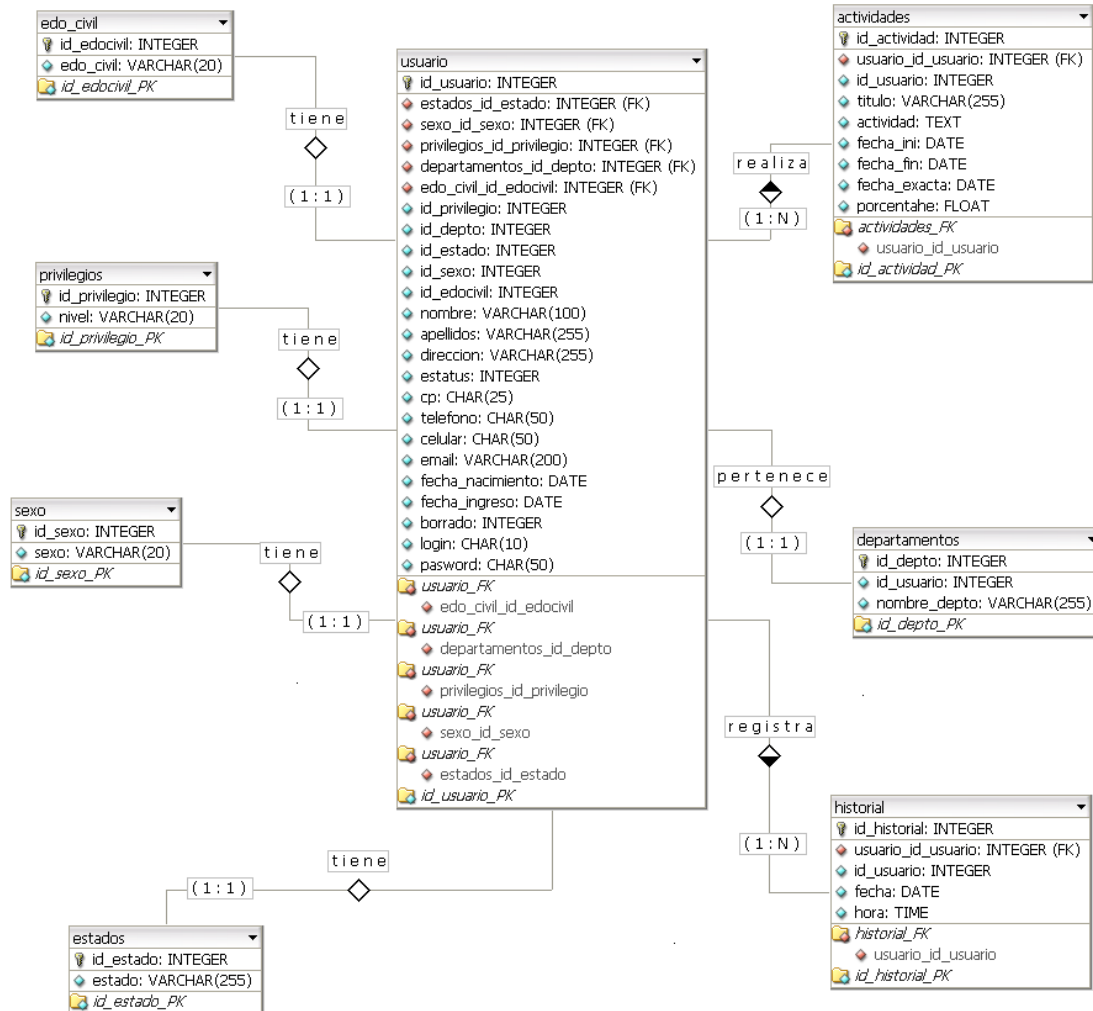
- **Altas.** Asignación de las actividades asociadas a los usuarios.
- **Modificaciones.** Actualización de la información almacenada correspondiente a las actividades.
- **Consulta.** Visualización a detalle de la actividad seleccionada.
- **Bajas.** Eliminación del registro seleccionado.

3. MODULO DEPARTAMENTOS.

- **Altas.** Captura de los departamentos existentes.
- **Modificaciones.** Actualización de la información asociada a cada departamento.
- **Bajas.** Eliminación del registro seleccionado.

3.2. MODELO ENTIDAD-RELACION

Es el modelo más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976.



3.3. DISEÑO LOGICO DE LA BASE DE DATOS

Se trata de un esquema lógico global que se ajuste al modelo de SGBD sobre el que se vaya a implementar el sistema. El objetivo principal del diseño lógico es obtener una representación que use, del modo más eficiente posible, los recursos que el modelo de SGBD posee para estructurar los datos y para modelar las restricciones.

Un primer paso en la fase del diseño lógico consistirá en la conversión de los mecanismos de representación de alto nivel, del modelo relacional, en términos de las estructuras de bajo nivel.

Reducción de diagrama Entidad – Relación a Tablas

actividades			
Campo	Tipo	Nulo	Tipo
id_actividad	int(2)	No	PRIMARY
id_usuario	int(2)	No	INDEX
titulo	varchar(255)	No	
actividad	text	Sí	
fecha_ini	date	No	
fecha_fin	date	No	
fecha_exacta	date	No	
porcentaje	float	No	

departamentos			
Campo	Tipo	Nulo	Tipo
id_depto	int(2)	No	PRIMARY
id_usuario	int(2)	No	INDEX
nombre_depto	varchar(255)	No	

edo_civil			
Campo	Tipo	Nulo	Tipo
id_edocivil	int(2)	No	PRIMARY
edocivil	varchar(20)	No	

estados			
Campo	Tipo	Nulo	Tipo
id_estado	int(2)	No	PRIMARY
estado	varchar(255)	No	

historial			
Campo	Tipo	Nulo	Tipo
id_historial	int(2)	No	PRIMARY
id_usuario	int(2)	No	INDEX
fecha	date	No	
hora	time	No	

privilegios			
Campo	Tipo	Nulo	Tipo
id_privilegio	int(2)	No	PRIMARY
nivel	varchar(20)	Sí	

sexo			
Campo	Tipo	Nulo	Tipo
id_sexo	int(2)	No	PRIMARY
sexo	varchar(20)	No	

usuario			
Campo	Tipo	Nulo	Tipo
id_usuario	int(2)	No	PRIMARY
id_privilegio	int(2)	No	INDEX
id_depto	int(2)	No	INDEX
id_estado	int(2)	No	INDEX
id_sexo	int(2)	No	INDEX
id_edocivil	int(2)	No	INDEX
nombre	varchar(100)	No	
apellidos	varchar(255)	No	
direccion	varchar(255)	No	
estatus	int(2)	No	
cp	char(25)	No	
telefono	char(50)	No	
celular	char(50)	No	
email	varchar(200)	No	
fecha_nacimiento	Date	No	
fecha_ingreso	Date	No	
borrado	int(2)	No	
login	char(10)	No	
password	char(50)	No	

3.4. NORMALIZACION

El objetivo de la normalización es asegurarse que cada campo de cada columna en todas las tablas es directamente dependiente de la llave, teniendo esto en cuenta se obtienen grandes beneficios, tales como, eliminar redundancias, disminuir anomalías y mejorar eficiencias. Este esquema es un buen punto de partida para fomentar un buen diseño de una base de datos.

El modelo relacional sólo requiere de un conjunto de relaciones en primera forma normal, las formas restantes son opcionales, sin embargo, para evitar anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.

- **Primera Forma Normal (1FN)** Incluye la eliminación de todos los grupos repetidos.
- **Segunda Forma Normal (2FN)** Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK).
- **Tercera Forma Normal (3FN)** Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

Tabla “actividades” se encuentra en 3 FN.

Campo
id_actividad (PK)
id_usuario (FK)
titulo
actividad
fecha_ini
fecha_fin
fecha_exacta
porcentaje

Tabla “departamentos” se encuentra en 3 FN.

Campo
id_depto (PK)
id_usuario (FK)
nombre_depto

Tabla “edo_civil” se encuentra en 3 FN.

Campo
id_edocivil (PK)
edocivil

Tabla “estados” se encuentra en 3 FN.

Campo
id_estado (PK)
estado

Tabla “historial” se encuentra en 3 FN.

Campo
id_historial (PK)
id_usuario (FK)
fecha

Tabla “privilegios” se encuentra en 3 FN.

Campo
id_privilegio (PK)
nivel

Tabla “sexo” se encuentra en 3 FN.

Campo
idsexo (PK)
sexo

Tabla “usuario” se encuentra en 3 FN.

Campo
id_usuario (PK)
id_privilegio (FK)
id_depto (FK)
id_estado (FK)
id_sexo(FK)
id_edocivil (FK)
nombre
apellidos
direccion
estatus
cp
telefono
celular
email
fecha_nacimiento
fecha_ingreso
borrado
login
password

3.5. DICCIONARIO DE DATOS

Es un conjunto de metadatos que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Tabla: actividades

Campo	Tipo	Nulo	Comentarios
id_actividad	int(11)	No	Llave primaria de la tabla
id_usuario	int(4)	No	Llave foránea de la tabla, hace referencia la tabla "usuario"
título	varchar(255)	No	Título principal de la actividad
actividad	text	Sí	Contenido explícito de toda la actividad a realizar
fecha_ini	date	No	Fecha propuesta de inicio para desarrollar dicha actividad
fecha_fin	date	No	Fecha propuesta para finalizar la actividad
fecha_exacta	date	No	Fecha real de culminación de la actividad
porcentaje	float	No	Resultado de los cálculos internos

Tabla: departamentos

Campo	Tipo	Nulo	Comentarios
id_depto	int(4)	No	Llave primaria de la tabla
id_usuario	varchar(200)	No	Llave foránea, relacionada con la tabla "usuario"
nombre_depto	int(4)	No	Nombre del departamento

Tabla: edo_civil

Campo	Tipo	Nulo	Comentarios
id_edocivil	int(1)	No	Llave primaria de la tabla
edocivil	char(20)	No	Estado civil

Tabla: estados

Campo	Tipo	Nulo	Comentarios
id_estado	int(4)	No	Llave primaria de la tabla
estado	varchar(200)	No	Nombre del estado de la República Mexicana

Tabla: historial

Campo	Tipo	Nulo	Comentarios
id_historial	int(4)	No	Llave primaria de la tabla
id_usuario	date	No	Llave foránea, relacionada con la tabla "usuario"
fecha	varchar(200)	No	Fecha de acceso al sistema
hora	time	No	Hora de acceso al sistema

Tabla: privilegios

Campo	Tipo	Nulo	Comentarios
id_privilegio	int(11)	No	Llave primaria de la tabla
nivel	char(6)	Sí	Tipo de nivel de acceso

Tabla: sexo

Campo	Tipo	Nulo	Comentarios
idsexo	int(1)	No	Llave primaria de la tabla
sexo	char(15)	No	Tipo de género

Tabla: usuario

Campo	Tipo	Nulo	Comentarios
id_usuario	int(4)	No	Llave primaria de la tabla
id_privilegio	int(4)	No	Llave foránea de la tabla, hace referencia a la tabla "privilegios"
id_depto	varchar(100)	No	Llave foránea de la tabla, hace referencia a la tabla "departamentos"
id_estado	varchar(255)	No	Llave foránea de la tabla, hace referencia a la tabla "estados"
idsexo	varchar(255)	No	Llave foránea de la tabla, hace referencia a la tabla "sexo"
id_edocivil	int(4)	No	Llave foránea de la tabla, hace referencia a la tabla "edo_civil"
nombre	int(2)	No	Nombre del usuario
apellidos	char(25)	No	Apellidos del usuario
direccion	varchar(25)	No	Dirección del usuario
estatus	char(50)	No	Usuario bloqueado o no
cp	char(50)	No	Código postal
telefono	varchar(200)	No	Teléfono particular

celular	char(25)	No	Teléfono celular
email	char(25)	No	Email
fecha_nacimiento	date	No	Fecha de nacimiento del usuario
fecha_ingreso	date	No	Fecha de ingreso a la empresa del usuario
borrado	int(1)	No	Usuario eliminado o no
login	char(10)	No	Login para acceso al sistema
password	char(50)	No	Contraseña para acceso al sistema

3.6. CRITERIO DE DISEÑO

Se analizó el tipo de arquitectura a implementar, y por tratarse de un sistema en un ambiente Web, pero sobretodo el involucrar almacenamiento de información en una base de datos, se optó por trabajar en un ambiente Cliente-Servidor.

Basándose en este análisis se establecieron criterios de diseño para este sistema y se estudiaron entonces las tecnologías a utilizar para el desarrollo, esto es, según el tamaño del sistema, el tipo de plataforma sobre el cual debe funcionar, la complejidad del negocio que el sistema resuelve, etc. Estos criterios contemplan también la funcionalidad capturada en el análisis del problema que el sistema en desarrollo busca resolver.

Básicamente se trata de que estos criterios de diseño mantengan la separación de la lógica del negocio, la presentación y los datos.

3.7. HERRAMIENTAS DE IMPLEMENTACION

Este sistema esta desarrollado en un ambiente Cliente-Servidor, es por ello que se emplearon herramientas de ambiente Web, es decir, el servidor Web sobre el cual se trabajo es Apache, el código fuente esta desarrollado en PHP, mientras que el motor de base de datos es MySQL, además de estas herramientas se utilizaron otras para el diseño como CSS y JavaScript.

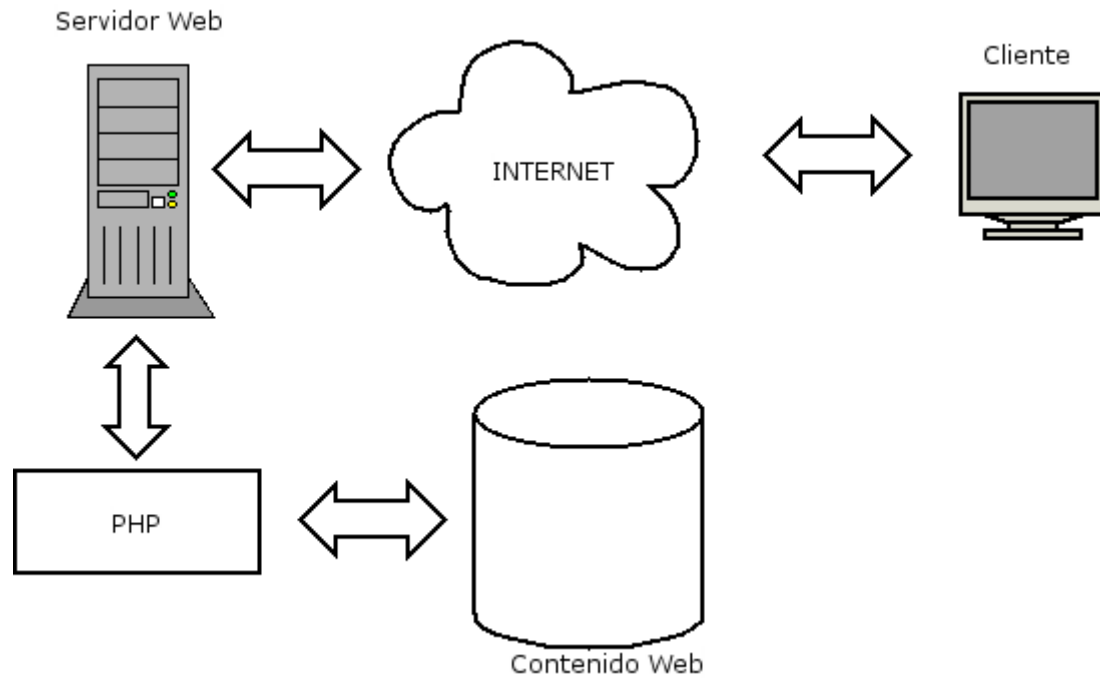


Figura 10. Modelo Cliente – Servidor, consulta Web con PHP.

De manera textual:

1. El Cliente realiza peticiones, desde un navegador, por Internet.
2. El servidor Web Apache atiende esa petición por medio de un script PHP.
3. El código PHP puede interactuar con una base de datos en MySQL u otros programas de contenido Web.
4. El servidor Web apache navegador se ocupa de producir la visualización en le Cliente.

Como se menciono anteriormente, el análisis de diseño permitió establecer el conjunto de herramientas utilizadas, esto es un punto importante, ya que este sistema es multiplataforma y por tanto no se encuentra restringido en funcionalidad, da igual que trabaje en un ambiente Windows o en un ambiente Linux. Además de trabajar en una diversidad de navegadores, como por ejemplo IE, Mozilla, Netscape, Opera.

CAPITULO IV

RESULTADOS Y ANALISIS

4.1. PRUEBAS Y SU ORIGEN

A continuación se describen los 2 tipos de pruebas realizadas, así como el porqué de las mismas.

Las pruebas realizadas fueron de acuerdo al tipo de usuario que puede acceder al sistema, es decir, el usuario Administrador y el usuario Trabajador. Como se verá más adelante, cada uno de ellos cuenta con ciertos privilegios en el manejo de la información, así como en las diferentes herramientas de administración.

Bastó realizar estas pruebas para darse cuenta que el sistema puede responder sin dificultad y con la información segura e íntegra ante diferentes circunstancias, en este caso, a los diferentes tipos de usuarios existentes.

4.2. TIPOS DE PRUEBAS CON INFORMACION REAL

La primera prueba corresponde al usuario Administrador, en las imágenes siguientes se verá el comportamiento del sistema cuando un usuario Administrador está logueado, las diferentes acciones y herramientas que se poseen para la administración del sistema.

También se cuentan con capturas de pantalla que corresponden al usuario Trabajador, es aquí donde se puede comparar la distinción de servicios entre un usuario y otro.

La información capturada y trabajada es real, es decir, es información que se maneja actualmente en una empresa manufacturera de piezas metálicas; cabe señalar que por el momento es poca la información, ya que se quiere conservar la confidencialidad de la misma, por ello solo se pueden observar pruebas de un usuario Administrador, un usuario Trabajador y cinco actividades, pero es información verídica.

Las capturas de pantalla subsiguientes muestran estas pruebas con la información ya almacenada en la base de datos, y que se maneja de forma diaria en la empresa.

4.3. INTERFASES DEL SISTEMA

A continuación se muestran las capturas de pantalla del sistema, tanto para usuarios Administrador o Trabajador.

Página de Inicio

Este es el primer script, es el módulo de autenticación de usuarios, que por medio de un nombre de usuario y contraseña el usuario será validado para concederle los privilegios adecuados según sea su rol dentro de la empresa, Administrador o Trabajador.

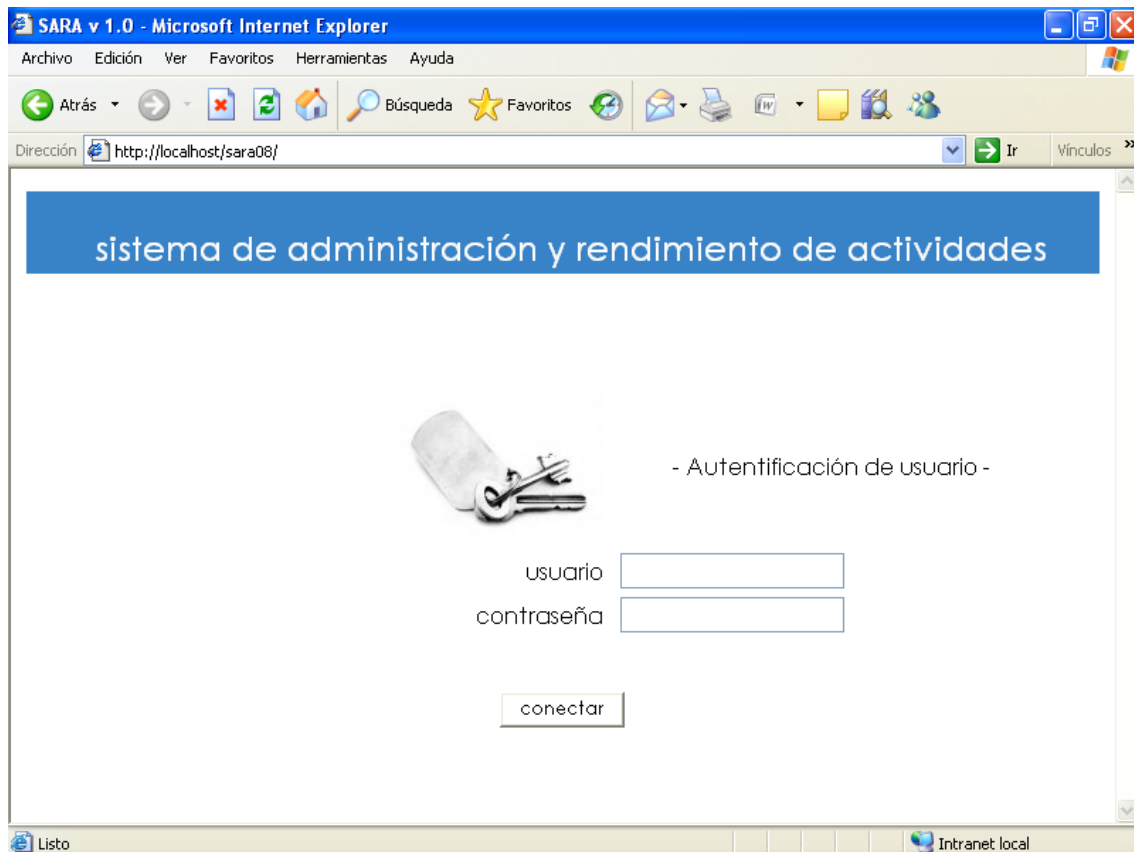


Figura 11. Pantalla inicial, autenticación de usuario.

Nombre de Usuario y Contraseña

El usuario debe proporcionar tanto su nombre de usuario y contraseña para poder ser validado por el sistema, de lo contrario recibirá un mensaje de advertencia.

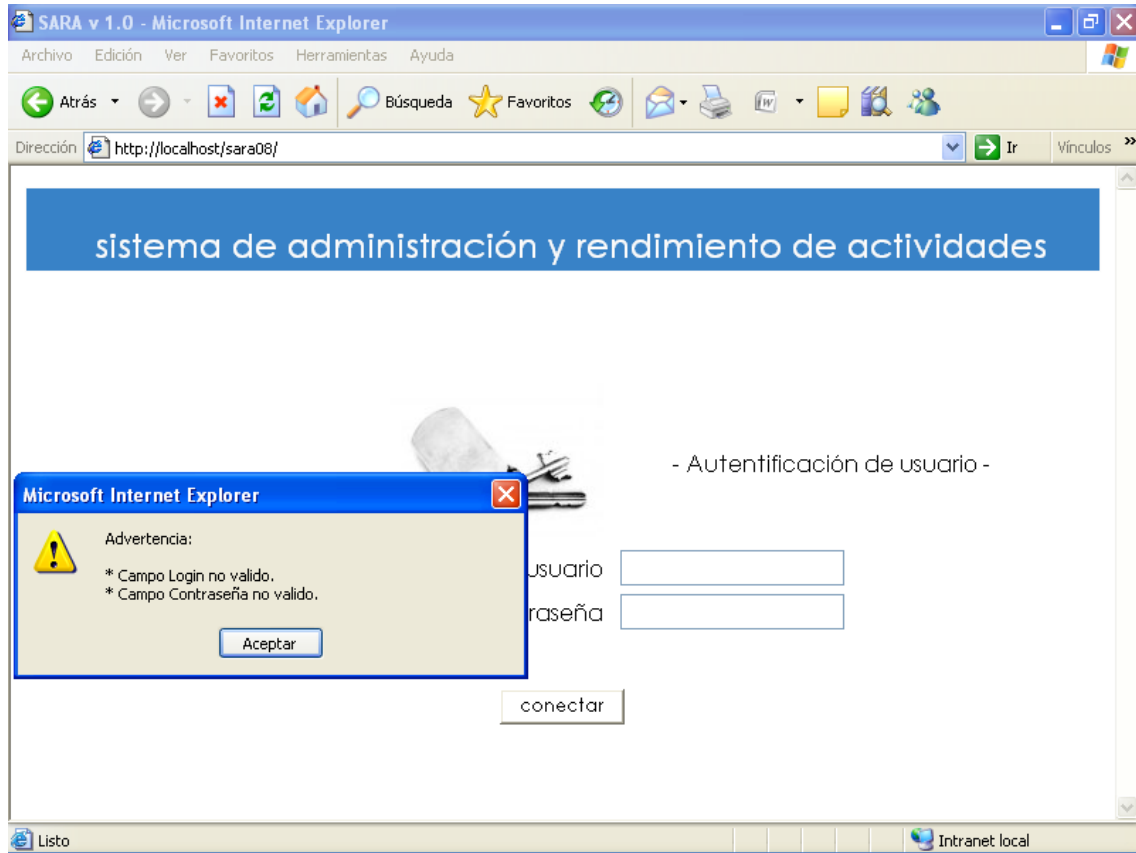


Figura 12. Mensaje de advertencia al intentar ingresar sin datos.

Usuario Inválido

Una vez proporcionados el nombre de usuario y contraseña, el sistema es capaz de validar esa información, denegado el acceso en caso de que alguno de los dos datos anteriores fuese incorrecto.

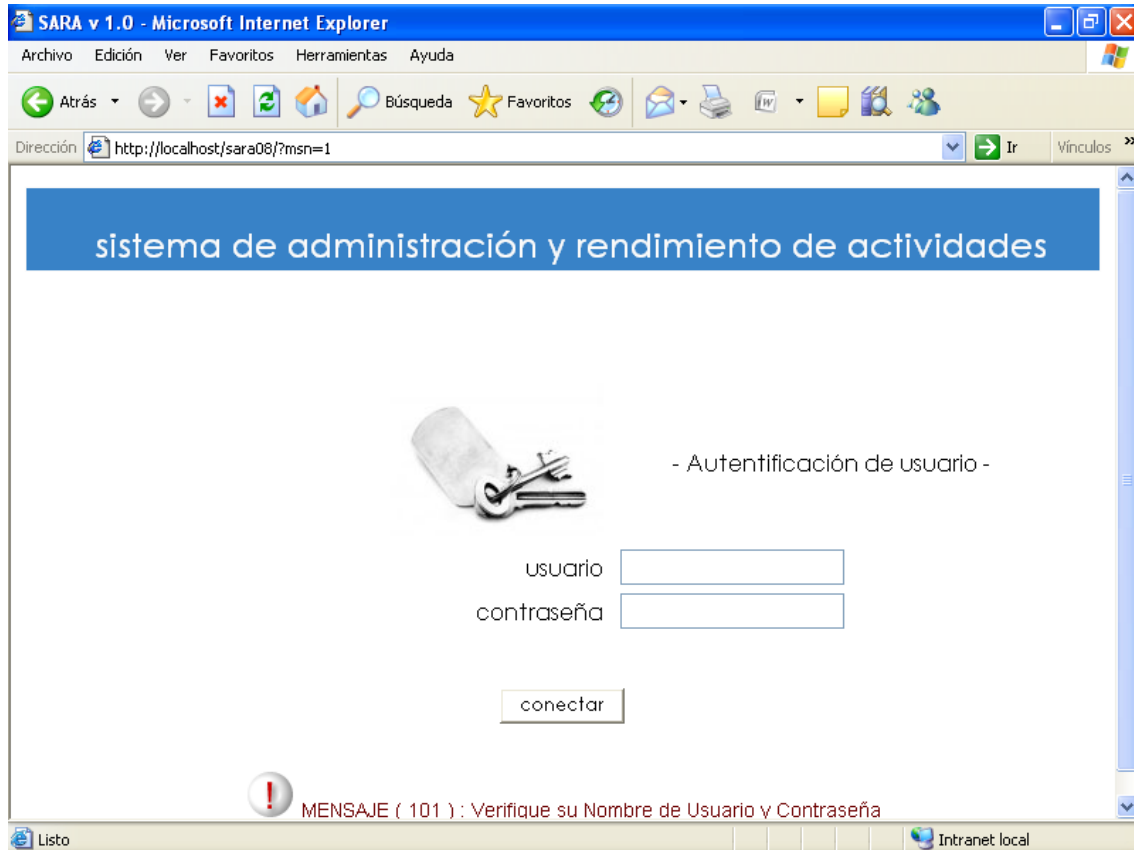


Figura 13. Mensaje de error, el nombre de usuario o contraseña son incorrectos.

Bienvenida

Esta es la pantalla principal, donde se muestra el menú y las opciones posibles de acuerdo al usuario autenticado; así como la fecha y hora del último acceso.

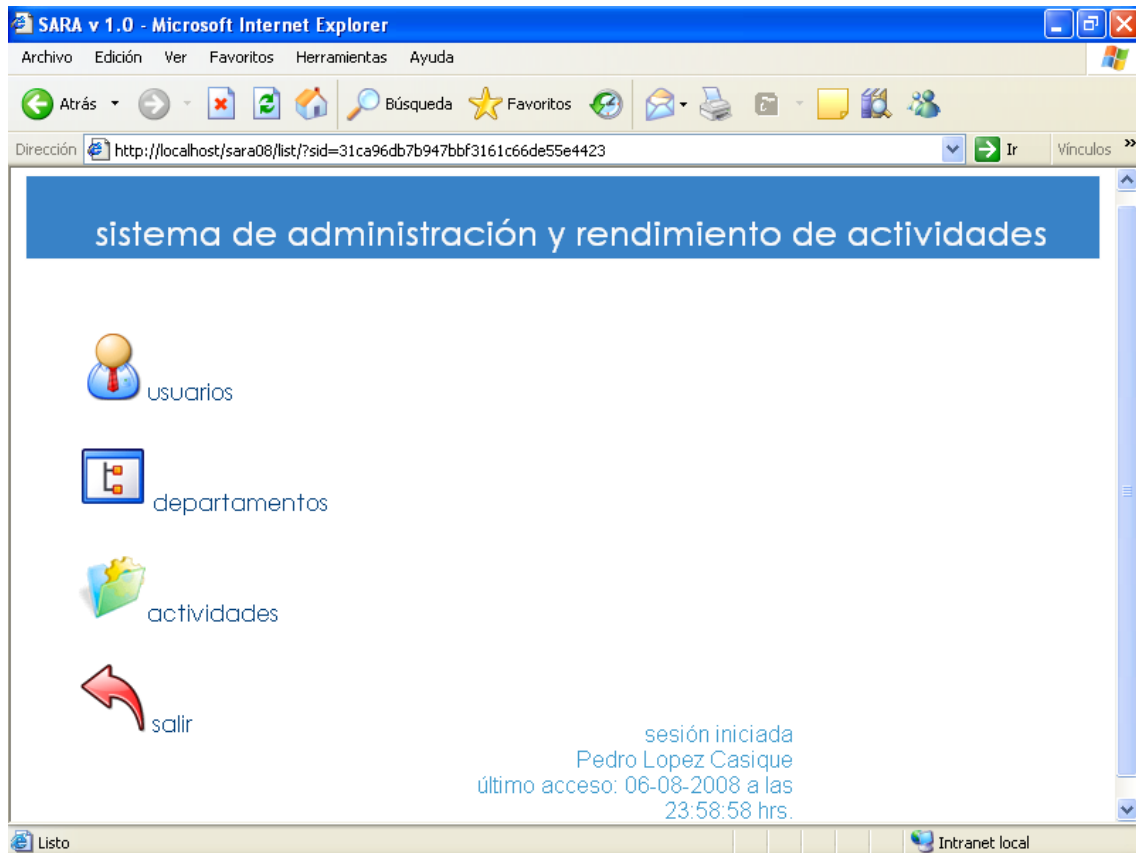






Figura 14. Ingreso al sistema por un usuario Administrador.

Menú

De acuerdo al usuario autenticado es como se le muestran las opciones en el menú.

Botón	Administrador	Trabajador
 usuarios	✓	
 departamentos	✓	
 actividades	✓	✓
 salir	✓	✓

Usuarios

Listado de los usuarios en el sistema, así como la fecha en que fueron registrados.

The screenshot shows a web browser window with the following content:

Browser: SARA - Personal - Microsoft Internet Explorer
 Address: http://localhost/sara08/list/personal.php?sid=31ca96db7b947bbf3161c66de55e4423

Page Title: sistema de administración y rendimiento de actividades

Navigation: inicio | usuarios | departamentos | actividades | salir


Section: Catálogo de Usuarios

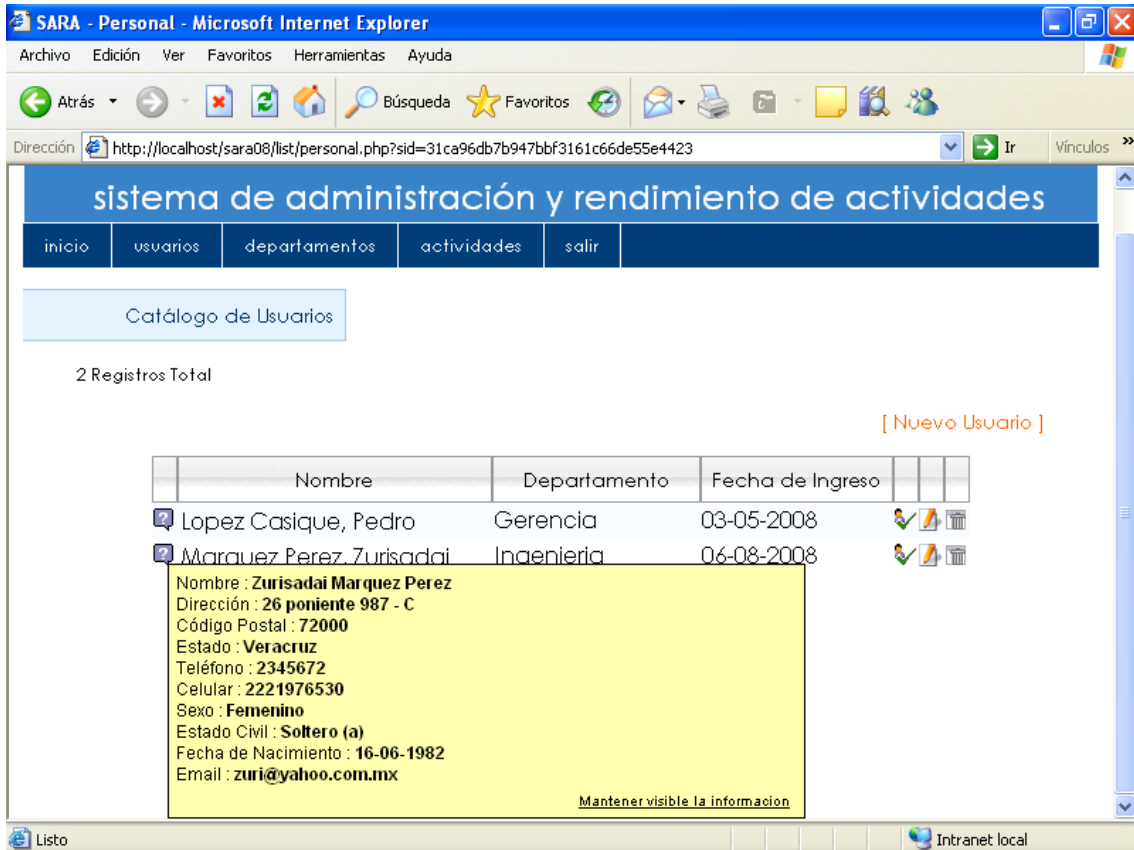
Summary: 2 Registros Total [Nuevo Usuario]

Nombre	Departamento	Fecha de Ingreso			
Lopez Casique, Pedro	Gerencia	03-05-2008	✓	✎	🗑️
Marquez Perez, Zurisadai	Ingenieria	06-08-2008	✓	✎	🗑️









Figura 15. Listado de usuarios registrados para el uso del sistema.

Ficha Personal

Para poder visualizar la información general de algún usuario basta colocarse en el icono  para que se muestre un globo con la información correspondiente al usuario.



The screenshot shows a web browser window titled "SARA - Personal - Microsoft Internet Explorer". The address bar shows the URL: `http://localhost/sara08/list/personal.php?sid=31ca96db7b947bbf3161c66de55e4423`. The page content includes a navigation menu with options: inicio, usuarios, departamentos, actividades, salir. Below the menu is a section titled "Catálogo de Usuarios" showing "2 Registros Total". A table lists two users:

Nombre	Departamento	Fecha de Ingreso			
 Lopez Casique, Pedro	Gerencia	03-05-2008			
 Marquez Perez, Zurisadai	Ingenieria	06-08-2008			

A tooltip is displayed over the second user, showing the following details:

- Nombre : **Zurisadai Marquez Perez**
- Dirección : **26 poniente 987 - C**
- Código Postal : **72000**
- Estado : **Veracruz**
- Teléfono : **2345672**
- Celular : **2221976530**
- Sexo : **Femenino**
- Estado Civil : **Soltero (a)**
- Fecha de Nacimiento : **16-06-1982**
- Email : **zuri@yahoo.com.mx**

At the bottom of the tooltip, there is a link: [Mantener visible la informacion](#). The browser status bar shows "Listo" and "Intranet local".

Figura 16. Información general del usuario seleccionado.

Actualización de Información.

El usuario Administrador es el único que puede actualizar la información de cada uno de los usuarios registrados.

Con la ayuda del calendario integrado, se pueden cambiar las fechas tanto de ingreso como de nacimiento.

The screenshot shows a web browser window titled "SARA - Actualiza Información - Microsoft Internet Explorer". The address bar shows a local URL. The page content includes a navigation menu with "inicio", "usuarios", "departamentos", "actividades", and "salir". A main heading reads "sistema de administración y rendimiento de actividades". Below this is a section titled "Actualiza Información de Usuario".

On the left, a calendar widget for August 2008 is displayed, with the date 8th highlighted. Below the calendar are dropdowns for "mes:" (Agosto) and "año:" (2008), and a button "[ir a fecha]".

The main form contains the following fields:

- Nombre: Zuriadai
- Apellidos: Marquez Perez
- Dirección: 26 poniente 987 - C
- Departamento: Ingeniería (dropdown)
- Estado: Veracruz (dropdown)
- Cp: 72000
- Telefono: 2345672
- Celular: 2221976530
- Email: zuri@yahoo.com.mx
- Sexo: Femenino (dropdown)
- Estado Civil: Soltero (a) (dropdown)
- Fecha Nacimiento: 16-06-1982
- Fecha Ingreso: 06-08-2008
- Login: zuri
- Password: masked with dots

At the bottom of the form are two buttons: "Actualizar" and "Cancelar".

Figura 17. Formulario de actualización de información para el usuario seleccionado.

Eliminación de Usuarios.

Al igual que el caso anterior, el Administrador puede eliminar a cualquiera de los usuarios del sistema.

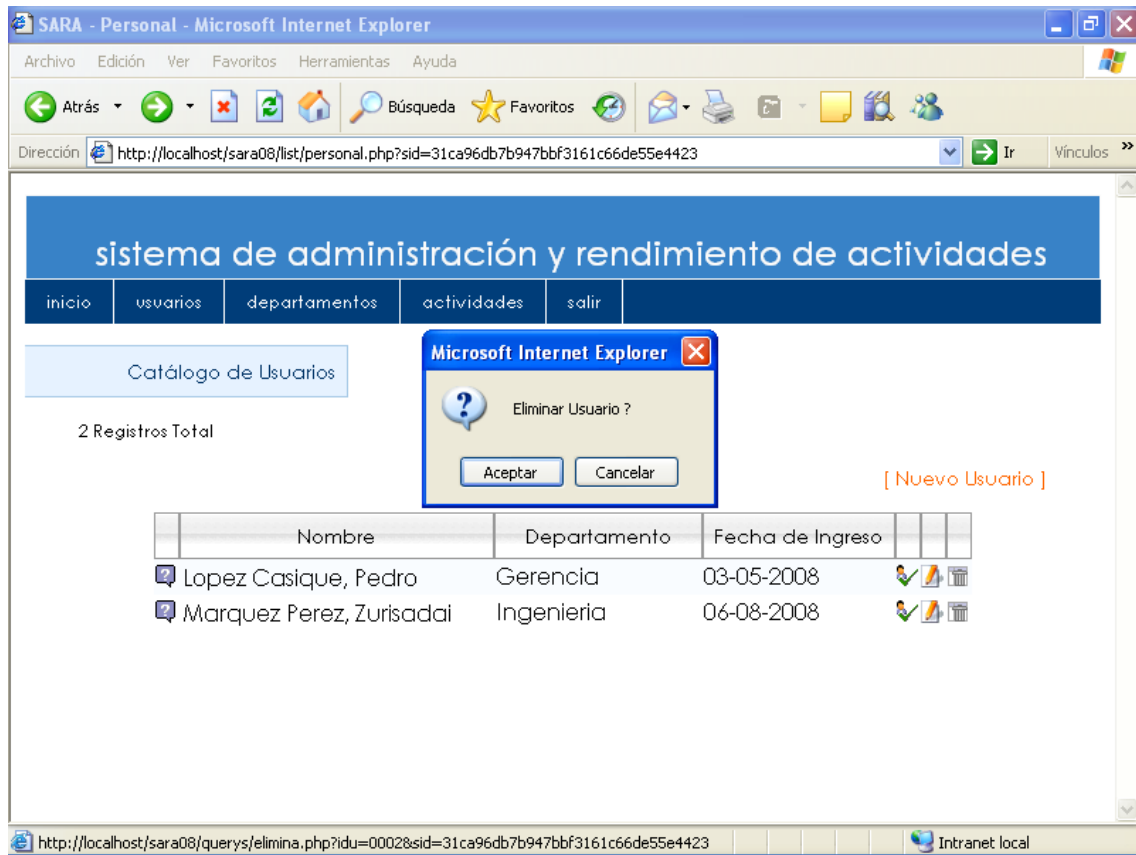


Figura 18. Mensaje de confirmación al querer eliminar el registro de un usuario.

Registro de Usuarios.

Pueden registrarse tantos usuarios como sea necesario, basta completar el formulario de captura para poder así dar de alta a un nuevo usuario Trabajador.

The image shows a screenshot of a web browser window titled "SARA - Nuevo Usuario - Microsoft Internet Explorer". The address bar shows the URL: `http://localhost/sara08/forms/registro.php?sid=31ca96db7b947bbf3161c66de55e44238&existe=yes`. The page content includes a navigation menu with links for "inicio", "usuarios", "departamentos", "actividades", and "salir". Below the menu is a section titled "Nuevo Usuario" containing a registration form with the following fields:

- Nombre:
- Apellidos:
- Dirección:
- Departamento:
- Estado:
- Cp:
- Telefono:
- Celular:
- Email:
- Sexo:
- Estado Civil:
- Fecha Nacimiento:
- Login:
- Password:

At the bottom of the form are two buttons: "Registrar" and "Cancelar". The browser's status bar at the bottom shows "Listo" and "Intranet local".

Figura 19. Formulario de captura para registrar un nuevo usuario al sistema.

Departamentos.

Este es la relación de los departamentos existentes dentro de la empresa; aquí mismo se encuentra el formulario para poder agregar un nuevo departamento.

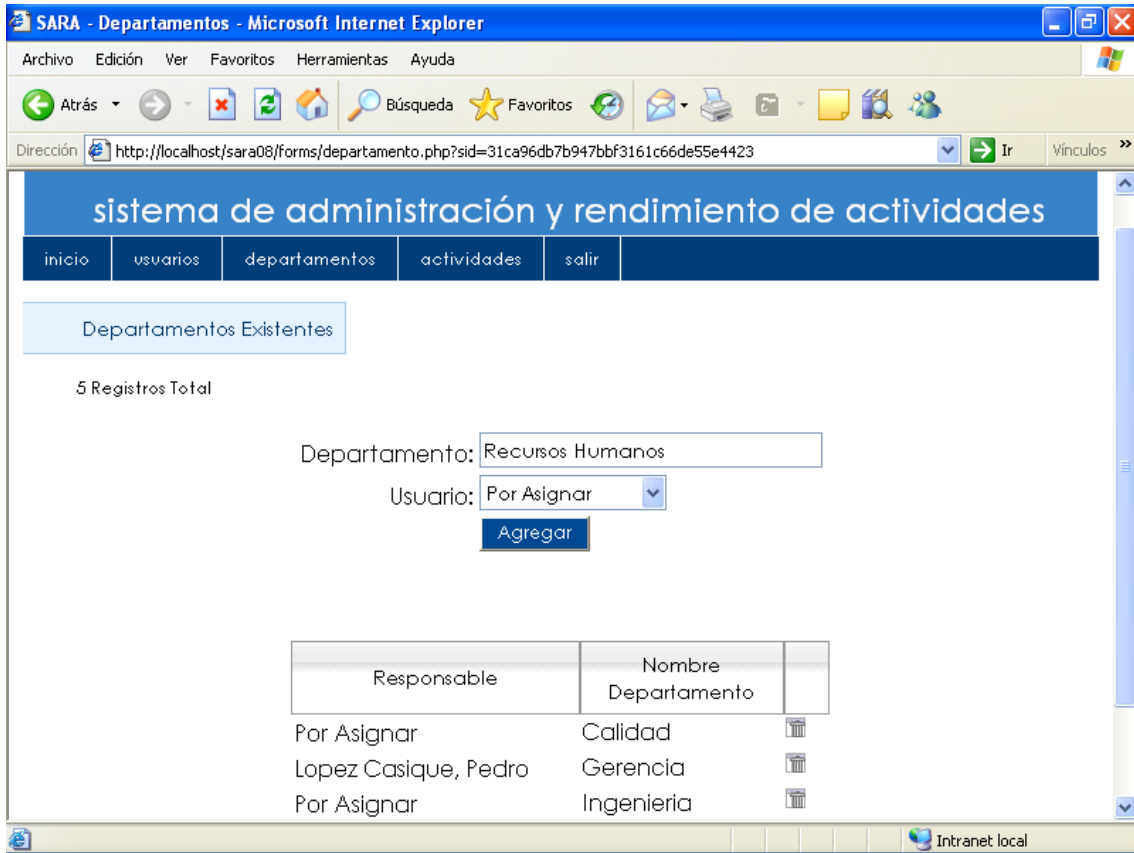


Figura 20. Listado de los departamentos existentes. Formulario de registro de nuevo departamento.

Eliminación de Departamento.

También hay la opción de poder eliminar algún departamento por parte del Administrador.

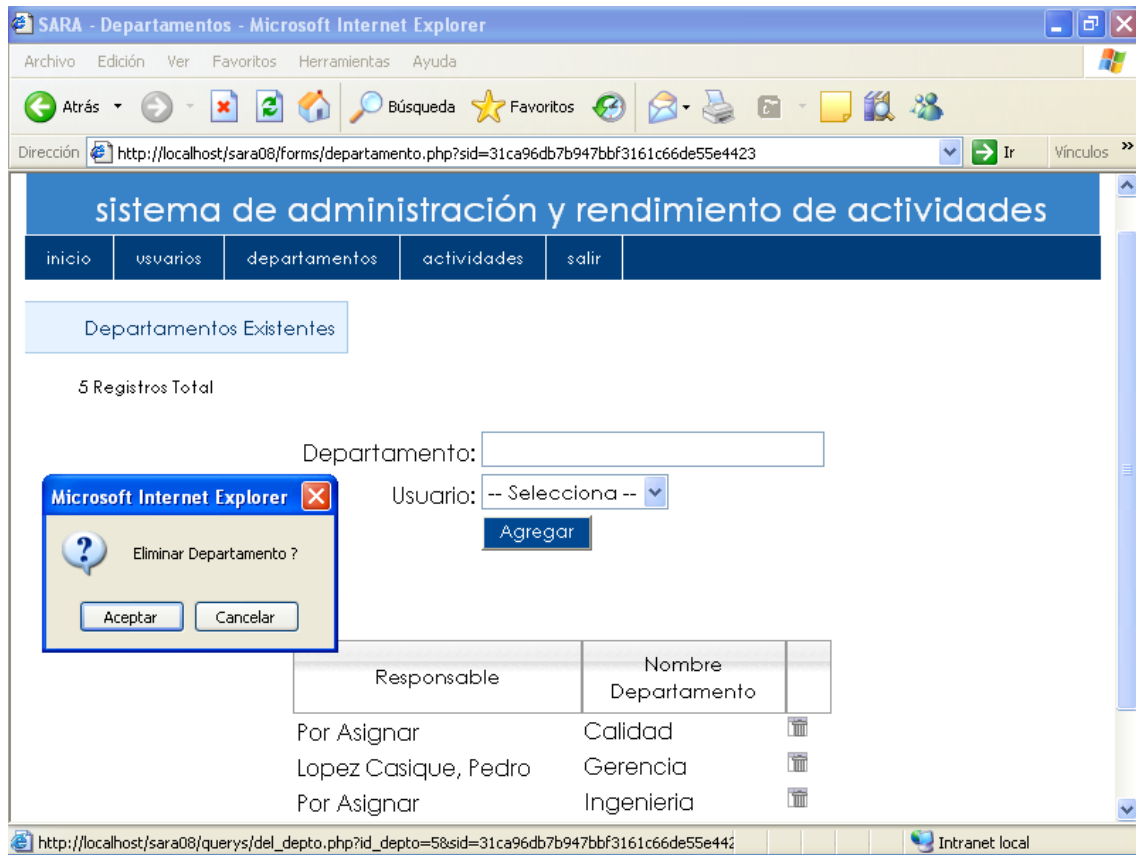


Figura 21. Mensaje de confirmación al querer eliminar un departamento.

Actividades y Rendimiento.

Esta es la parte principal del sistema SARA, se trata del concentrado de las actividades registradas así como del porcentaje de rendimiento e cada uno de ellas.

Este es un listado en estado inicial, es decir, aun no hay fechas de término, por lo que no es posible calcular todavía el porcentaje de rendimiento.

Mas adelante se muestran las mismas actividades pero ya con sus códigos de colores correspondientes a sus porcentajes.

Usuario	Titulo	Actividad	Fecha Inicial	Fecha Final	Fecha Exacta	Porcentaje
0001	Nuevo troquel AS-87	Diseño y creación de la nueva pieza para VW	12-05-2008	00-00-0000	?	?
0002	Planos de Diseño	Generar los planos para la máquina FC-X-09	20-08-2008	17-09-2008	?	?
0002	Manual de Calidad	Reporte del manual de calidad para Auditoria	12-08-2008	26-08-2008	?	?
0001	Asignacion Equipo	Junta con Zurisadai para entregarle equipo de meteorologia	11-08-2008	11-08-2008	?	?
0002	Solicitud de Cotizacion	Cotizar al proveedor Simplex el costo de 1000 piezas NM-09-PL	23-08-2008	23-08-2008	?	?

Figura 22. Listado de actividades, de todos los usuarios.

Porcentaje de las actividades de los diferentes usuarios.

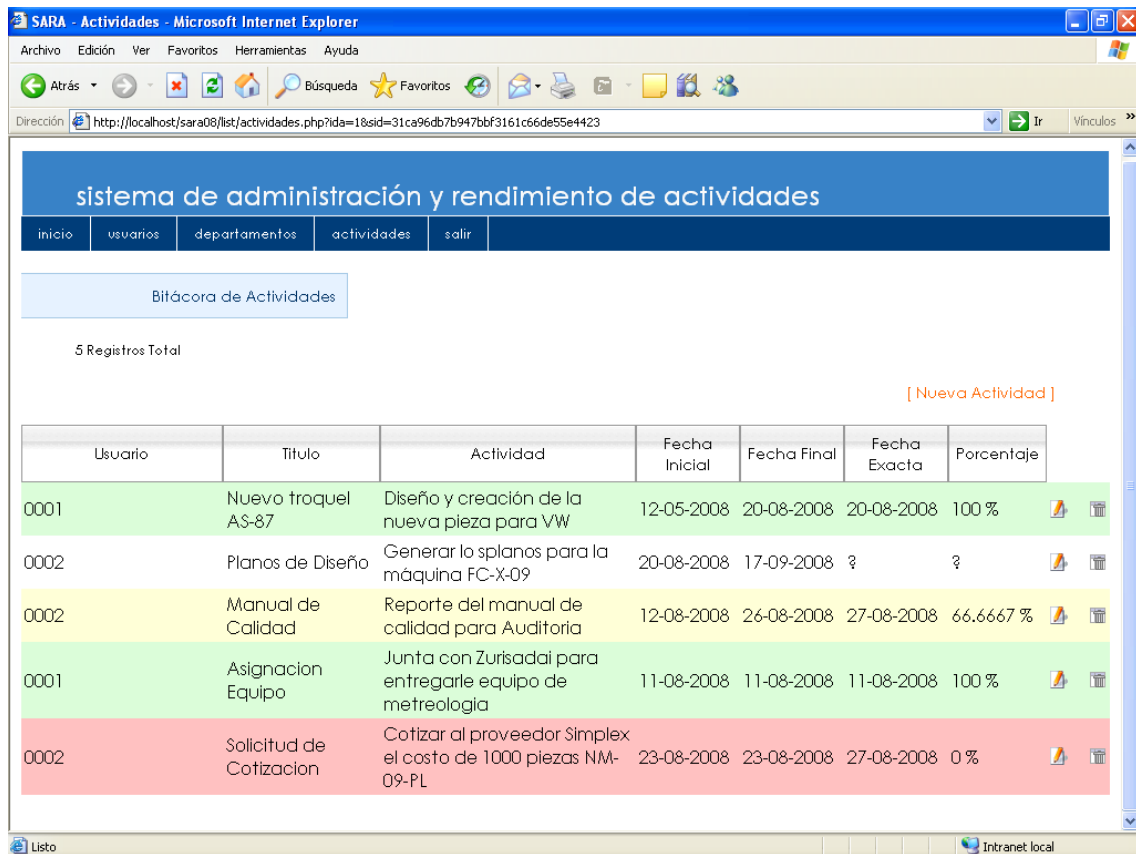
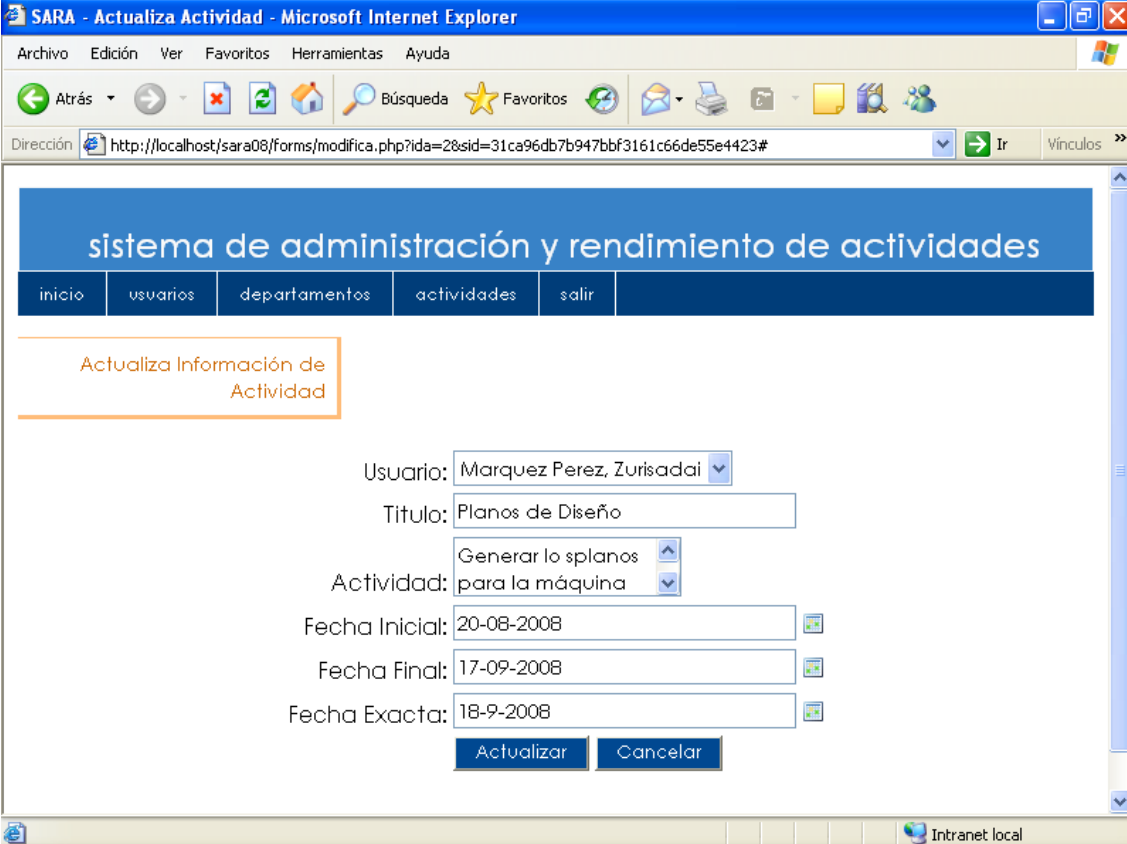


Figura 23. Resumen de actividades por código de colores.

Actualización de Actividad

Formulario que actualiza la información de alguna actividad.



The screenshot shows a Microsoft Internet Explorer browser window titled "SARA - Actualiza Actividad". The address bar displays the URL: `http://localhost/sara08/forms/modifica.php?ida=2&sid=31ca96db7b947bbf3161c66de55e4423#`. The page content includes a navigation menu with options: inicio, usuarios, departamentos, actividades, and salir. A highlighted section titled "Actualiza Información de Actividad" contains the following form fields:

- Usuario: Marquez Perez, Zurisadai (dropdown menu)
- Título: Planos de Diseño (text input)
- Actividad: Generar lo splanos para la máquina (dropdown menu)
- Fecha Inicial: 20-08-2008 (calendar icon)
- Fecha Final: 17-09-2008 (calendar icon)
- Fecha Exacta: 18-9-2008 (calendar icon)

At the bottom of the form are two buttons: "Actualizar" and "Cancelar". The browser's status bar at the bottom right shows "Intranet local".

Figura 24. Formulario de actualización para una actividad seleccionada.

Eliminación de Actividades.

Nuevamente, el usuario Administrador es quien puede eliminar cualquiera de las actividades ya registradas.

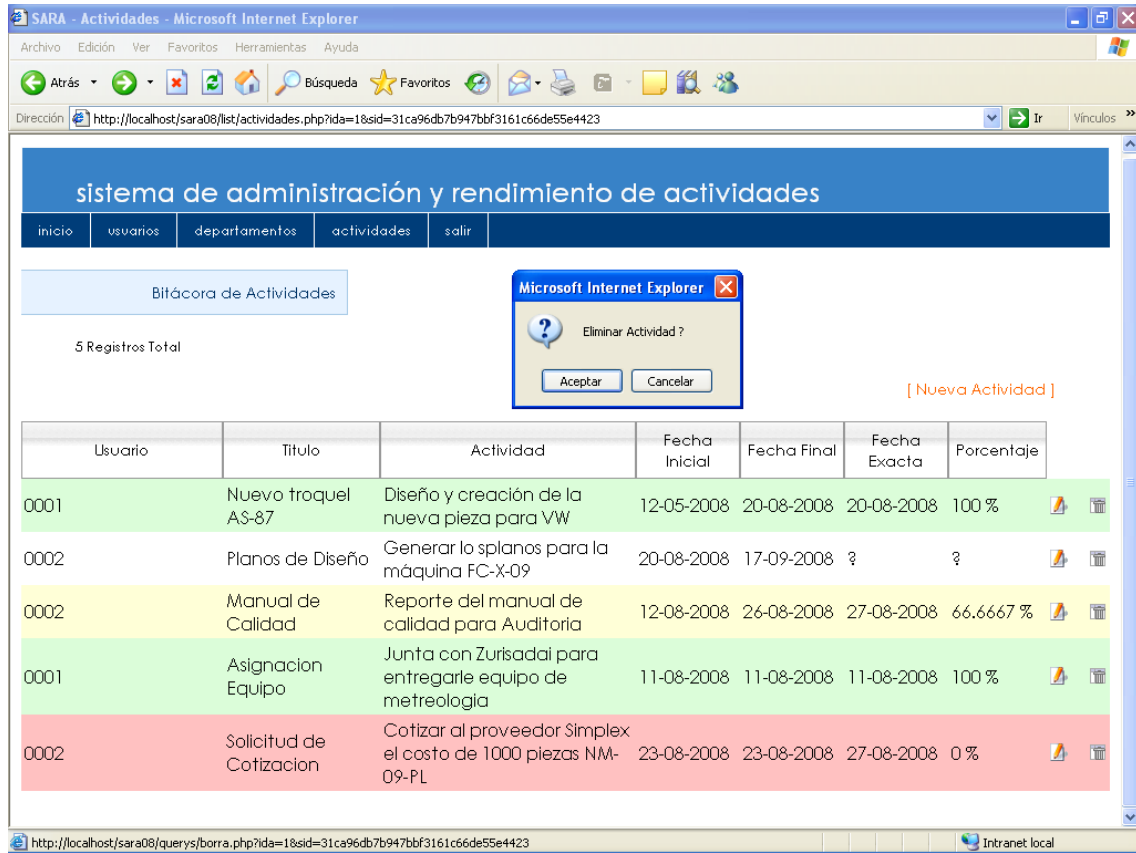


Figura 25. Mensaje de confirmación al querer eliminar una actividad.

Registro de Actividades.

Tanto el usuario Administrador como el Trabajador pueden crear nuevas actividades por medio de este formulario.

SARA - Nueva Actividad - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost/sara08/forms/actividad.php?sid=31ca96db7b947bbf3161c66de55e4423#> Ir Vinculos >>

sistema de administración y rendimiento de actividades

inicio usuarios departamentos actividades salir

Nueva Actividad

Usuario: Marquez Perez, Zurisadai

Titulo: Inventario

Actividad: Inventario de Metrologia

Fecha Inicio: 18-12-2008

Fecha Final: 25-12-2008

Registrar Cancelar

Intranet local

Figura 26. Formulario de captura de una nueva actividad.

Cierre de Sesión.

Hay una confirmación al querer abandonar el sistema.

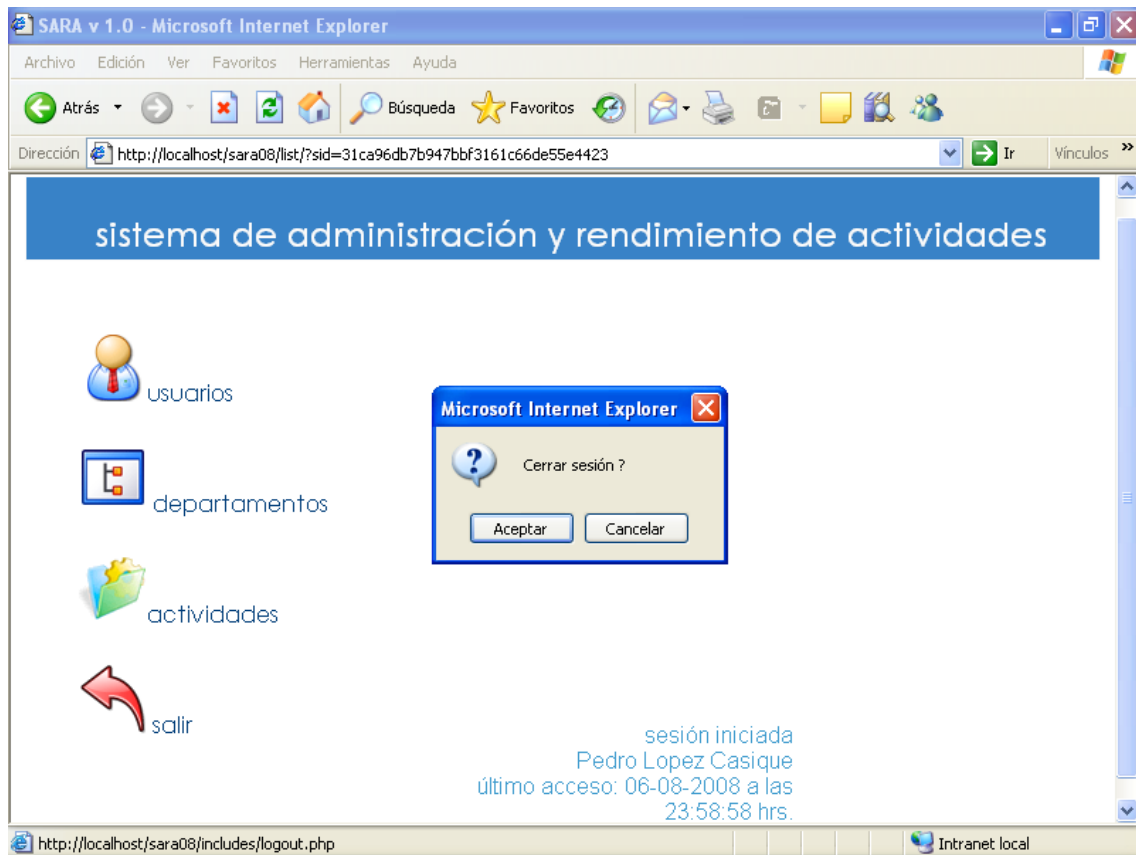


Figura 27. Mensaje de confirmación al salir del sistema.

Usuario Administrador y Usuario Trabajador

A cada uno de ellos se le presenta un menú diferente como ya se ha visto anteriormente.



Figura 28. Menú para el usuario logeado como Trabajador.



Figura 29. Menú para el usuario logeado como Administrador.

Para el usuario Trabajador solo cuenta con la opción de Actividades.

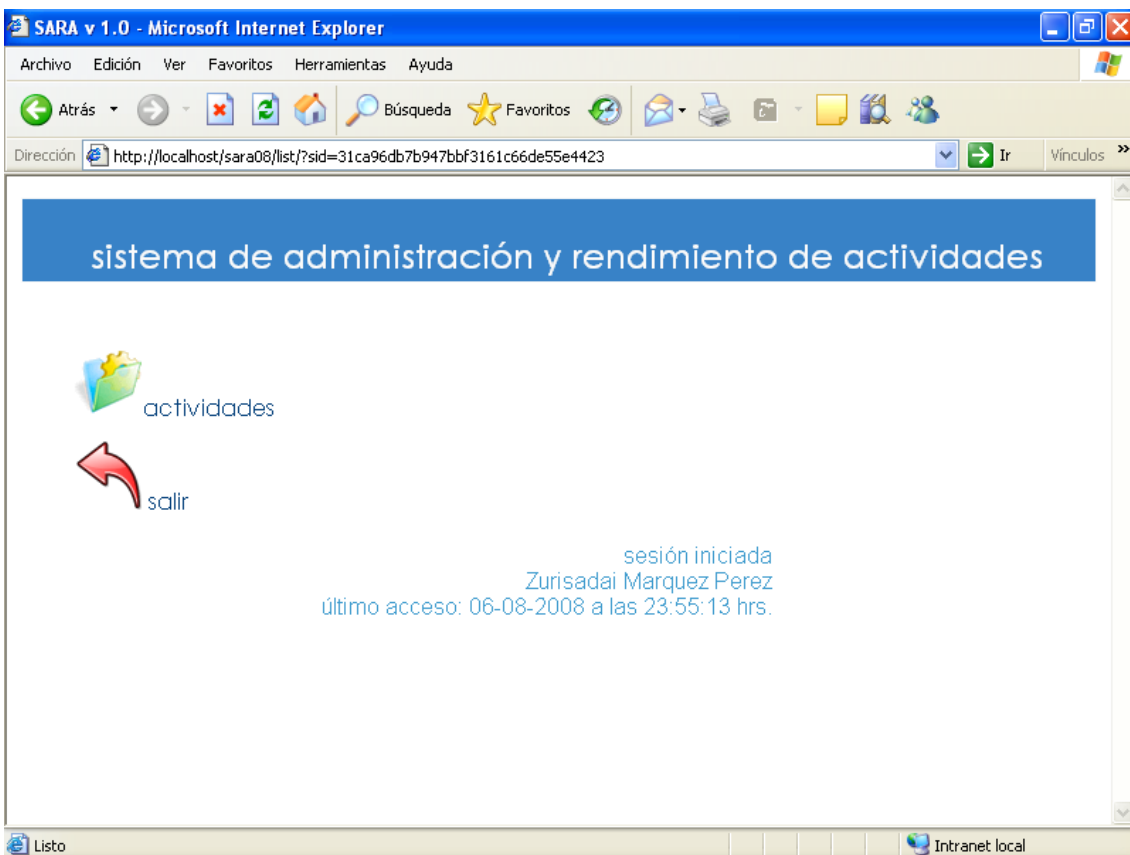


Figura 30. Bienvenida al usuario con rol Trabajador.

Actividades para Trabajador.

Básicamente el contenido de cada uno de los scripts se presenta de manera similar tanto para el usuario Administrador como para el Trabajador; las diferencias consisten en algunas herramientas de administración.



Figura 31. Listado de actividades que corresponden al usuario Trabajador.

Sesión

Se lleva un historial de acceso de cada uno de los usuarios, fechas y horas.

sesión iniciada
Pedro Lopez Casique
último acceso: 07-08-2008 a las
23:59:51 hrs.

Figura 32. Fecha y hora de la última sesión registrada en el sistema para el usuario logeado.

Misceláneos.

Están son las descripciones los diferentes mensajes y botones del sistema.

Nombre : Zurisdai Marquez Perez Dirección : 26 poniente 987 - C Código Postal : 72000 Estado : Veracruz Teléfono : 2345672 Celular : 2221976530 Sexo : Femenino Estado Civil : Soltero (a) Fecha de Nacimiento : 16-06-1982 Email : zuri@yahoo.com.mx	Mantener visible la informacion
--	---

Figura 33. Ficha informativa de los datos generales del usuario.



Figura 34. Mensaje de confirmación en eliminación de usuario.



Figura 35. Mensaje de confirmación en la eliminación de un departamento.



Figura 36. Mensaje de confirmación en la eliminación de una actividad.



Figura 37. Mensaje de confirmación al abandonar el sistema y terminar la sesión.



Figura 38. Botón para visualizar los usuarios registrados en el sistema.



Figura 39. Botón para mostrar los departamentos existentes en la empresa.



Figura 40. Botón que muestra el resumen de actividades a realizar dentro de la empresa.



Figura 41. Botón para terminar la sesión y salir del sistema.

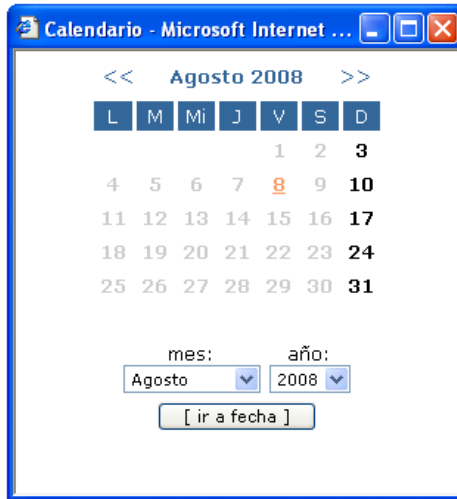


Figura 42. Calendario para los formularios.

4.4. ANALISIS DE RESULTADOS

Los resultados obtenidos fueron satisfactorios; la interfaz de usuario es intuitiva para el usuario final, se puede acceder al menú en cualquier momento, cuenta con la suficiente información para que el usuario pueda trabajar sin dificultad y sobre todo sin duda alguna; el usuario Administrador tiene el control absoluto de la información, puede manejarla libremente y los resultados se muestran en tiempo real; mientras tanto el usuario Trabajador solo puede consultar su propia información, y por ningún motivo tiene acceso a información confidencial; los tiempos de respuesta son inmediatos a las peticiones; los resultados esperados de los cálculos internos son precisos.

El sistema es una herramienta de gran ayuda en la administración de actividades, así como en el cálculo de rendimientos por parte de los usuarios involucrados en proyectos. Esta herramienta es de gran ayuda para el control de tiempos en la administración de proyectos.

Un punto importante que hay que destacar, es el hecho del mantenimiento del sistema; por tratarse de un desarrollo mediante un lenguaje de código abierto es posible darle mantenimiento sin dificultad, e incluso hacer el sistema mas robusto.

La arquitectura empleada, Cliente / Servidor, permite al sistema ser portable, es decir, puede trabajar tanto en un ambiente Windows como Linux, y esta es una gran ventaja que distingue este software de otros que se encuentran en el mercado.

CONCLUSIONES

Las aplicaciones Web están teniendo gran auge en el desarrollo de soluciones informáticas. Estas soluciones pueden desarrollarse en cualquier sector, ya sea el gubernamental, el comercial, el sector salud o el industrial.

Las tecnologías empleadas en este proyecto, fueron tecnologías de código abierto, y mediante un arquitectura cliente / servidor, se pudo obtener una herramienta que de solución a la necesidad encontrada en el análisis de requerimientos.

Otro aspecto importante es el buen diseño de una base de datos, gracias a esto podemos evitar redundancia de información, inconsistencia de los datos y el tiempo de ejecución de las consultas SQL son mas rápidas.

En particular, SARA es capaz de arrojar resultados confiables, en tiempo real, y siempre se mantiene la confidencialidad de la información, aun frente a gente especializada.

Puedo resumir este proyecto en 3 puntos:

1. **Identificación del Problema.** La garantía de que un sistema funcione se basa 100% en la identificación del problema a resolver, y de aquí poder brindar una solución, la más óptima.
2. **Solución a Ejecutar.** La propuesta generada, como resultado en el paso anterior, se lleva a la fase de análisis, y a su posterior implementación, que mediante ingeniería de software y un buen diseño de la base de datos obtendremos los resultados esperados.
3. **Arquitectura y Herramientas de Desarrollo.** Nuestro papel como desarrolladores no finaliza en la implementación, no basta con programar solamente, necesitamos conocer muy bien el ambiente sobre el cual trabajará nuestro software, y seleccionar las mejores herramientas de desarrollo, para brindar un sistema confiable, portable, seguro, exacto, y de bajo coste.

Hoy en día, los bancos e incluso el gobierno, tienen sus aplicaciones trabajando en Web, lo que nos muestra la gran eficacia que brindan las herramientas tecnológicas actuales, como Apache, PHP y MySQL, y también la seguridad en ellas.

BIBLIOGRAFIA

1. Hugh E. Williams, David Lane. Web Database Applications with PHP & MySQL 2nd Edition, 2004.
2. Sascha Schumann; Deepak Veliath, Harish Rawat, and Jesus Castagnetto. Professional PHP Programming, 1999.
3. Angel Cobo, Patricia Gómez. PHP y MySQL - Tecnologías Para El Desarrollo de Aplicaciones Web, 2005.

REFERENCIAS ELECTRONICAS

1. <http://www.apache.org>
2. <http://www.php.net>
3. <http://www.mysql.com>
4. http://cq-pan.cqu.edu.au/david-jones/Reading/Web_Engineering/
5. <http://download255.mediafire.com/jkim9i9dl5zg/mjymskttjx2/redes+de+ordenadores+-+andrew+s+tanenbaum.pdf>