



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**“SISTEMA DE SOPORTE TÉCNICO DESARROLLADO EN JSP  
PARA LA INTRANET DE LA DIRECCIÓN GENERAL DE  
RECURSOS HUMANOS DE LA SECRETARÍA DE HACIENDA Y  
CRÉDITO PÚBLICO”**

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

**PRESENTA:  
RUIZ RODRÍGUEZ DANIEL OCTAVIO**

**ASESOR:  
ING. JUAN JOSÉ CARREON GRANADOS**



**MÉXICO D.F.**

**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.

## **Dedicatorias.**

Quiero dedicar esta tesis con todo cariño a:

Mi gran amigo y maestro, **DIOS**, que me dio la vida y que está en cada momento de mi vida. Todo contigo, nada sin ti.

A mi madre, Magdalena, la más linda del mundo, que me permitió nacer y que gracias a sus consejos, amor y comprensión nos ha llevado a mi y a mis hermanos por el camino del bien. Te adoro "jechu".

A mi padre, Octavio, por apoyarme en todo momento por difícil que este sea, por darme ese espíritu de superación constante y por ser mi gran amigo.

A Martina, mi pequeña hija, que siempre está en mis pensamientos y que es mi principal motivación en esta vida para existir y ser mejor. Te amo.

A mis hermanos Talina y Luis, gracias por existir y por ser ese gran complemento de mi vida. Sin ustedes, que aburrida hubiera sido la vida. Los quiero mucho.

A mis segundas madres, mi tía Socorro y mi tía Rufina, que DIOS las bendiga por siempre.

A mis abuelos, Octavio, Blanca, Delfino y Adelaida, que aunque estén ya con DIOS, siempre los recordamos con mucho cariño.

A Ángel Castillo, por ser el mejor amigo que uno puede tener. Como te quiero condenadote.

A Erich "Von" Monroy y a Alfonso "Lancelot" Cabello, por su gran corazón, por ser mis gurús y porque gracias a ustedes, puedo decir que la amistad verdadera existe.

A Mariana Zaragoza, por tu cariño y apoyo incondicional. Eres una persona excepcional a la que quiero mucho. Sigue adelante y no te detengas.

A Vanessa Montaña, por todo lo que aprendí contigo, por todo lo que pasamos y porque siempre me diste palabras de aliento en los momentos que más lo necesité. Gracias de verdad, desde México hasta la Paz, B.C.S.

A todos mis maestros de la Facultad de Ingeniería, en especial al Ing. Juan José Carreón Granados, al Ing. Carlos Román Zamitis, al Fis. Raymundo Hugo Rangel, al Ing. Orlando Zaldívar, a la Ing. Laura Sandoval, al Ing. Torres H. y al Dr. Abel Herrera. Gracias a ellos, a sus enseñanzas y sus consejos, he podido desempeñarme de una manera exitosa en el campo profesional.

A todo el colectivo Aztekinetix, en especial a mi "super brother", Sergio Rodríguez "Pache" y a Omar Pérez, "Peyo". Dos personas con un gran corazón, de las cuáles me siento muy feliz de haberlas conocido.

A la Lic. Carmelina Olmos y al Mat. Manuel Villasana, por su cariño, por las experiencias compartidas, y por haber sido el mejor ejemplo de cómo un líder debe ser.

A Sandra Pulido, mi gran compañera profesional y amiga, con la cual compartí el proyecto de Intranet de donde se desprende este trabajo de tesis, y que siempre me alentó a terminar con la carrera.

A Ximena Quintero y Adriana Carboney, mis grandes comadres y confidentes. Las quiero mucho.

Al Ing. Boris Franco, a Vicente, Miriam, Carlos y Jorge (The Kong Family), de la AAADAM, por confiar en mí, por permitirme pertenecer a su equipo de trabajo y por el apoyo brindado para la culminación del presente trabajo.

A todos mis amigos: Carlos Mendoza (DGSCA), Eunice Romero y Olmos, Carlos Hernández, Isai López, Israel Muñoz, Chaken, Polo, Fernando "Fercho" Martínez, Roberto Terán, Rocío López, Berenice Hernández, Areli Rodríguez, Dalia Romero, Ulises Osorio, Julieta Bandaccari, Omar Chirino, Daniel Hernández, a FARVAC, a TZOALLI, al Gabo, al Crispy; y a todos los que han participado como actores en la película de mi vida.

"Que la paz esté con ustedes"  
(Jesucristo)

## ÍNDICE.

<b>INTRODUCCIÓN</b>	<b>2</b>
<b>CAPÍTULO 1. INTRANET</b>	<b>4</b>
1.1 Intranet	5
1.2 Intranet de la Dirección General de Recursos Humanos de la SHCP	6
1.3 Objetivos de la Intranet dentro de la DGRH	9
1.4 Tecnología utilizada para la creación de la Intranet de la DGRH de la SHCP	10
<b>CAPÍTULO 2. SERVIDORES DE APLICACIÓN</b>	<b>12</b>
2.1 Antecedentes	13
2.2 Servidor de aplicación	15
2.3 Arquitectura de tres capas	16
2.4 Principales lenguajes utilizados por los servidores de aplicación	17
2.5 Servidor de Aplicaciones Tomcat	20
<b>CAPÍTULO 3. JAVA SERVER PAGES</b>	<b>21</b>
3.1 Lenguaje de programación Java	22
3.2 JAVA y las JSP	24
3.3 JSP y las bases de datos	30
<b>CAPÍTULO 4. ANÁLISIS DEL SISTEMA DE SOPORTE TÉCNICO</b>	<b>36</b>
4.1 Antecedentes	37
4.2 Etapa de Análisis de Requerimientos	40
4.3 Análisis de Requerimientos aplicado al Sistema de Soporte Técnico	42
4.4 Selección del software	46
4.5 Alternativas para el Sistema de Base de Datos	47
4.6 Alternativas de servidores para la capa del medio (middle tier)	48
4.7 Proceso para la generación de un reporte de usuario	50
4.8 Proceso para la gestión de los reportes (parte administrativa)	51
4.9 Diseño del esquema de la base de datos	54
4.10 Diseño de las interfaces de usuario	62
<b>CAPÍTULO 5. DESARROLLO DEL CÓDIGO EN JSP</b>	<b>77</b>
5.1 Antecedentes	78
5.2 Levantamiento de Reportes por los usuarios	79
5.2 Parte Administrativa y Gestión de los Reporte	89
<b>CAPÍTULO 6. PRUEBAS CON EL USUARIO FINAL</b>	<b>122</b>
<b>CONCLUSIONES</b>	<b>125</b>
<b>BIBLIOGRAFÍA</b>	<b>130</b>
<b>ANEXO A. MANUAL TÉCNICO Y MANUAL DE USUARIO</b>	

## Introducción

El presente trabajo de tesis muestra el desarrollo de un sistema para la atención de reportes, correspondientes a un área de soporte técnico, utilizando como medio de acceso a la Intranet de la Dirección General de Recursos Humanos de la Secretaría de Hacienda y Crédito Público.

El desarrollo de este sistema se llevó a cabo durante los meses de enero y marzo del 2003, con la finalidad de llevar un mejor control sobre los reportes de fallas, que los usuarios hacen al área de soporte técnico. Antes de la puesta en marcha del presente sistema, el levantamiento de un reporte se efectuaba mediante papel y llamadas telefónicas; por lo cual si los datos iban a ser procesados con fines estadísticos, tenían que ser capturados en una hoja de cálculo ó una base de datos como Access.

El presente trabajo consta de 6 capítulos, los cuales van desde la definición de conceptos clave utilizados para la creación de este sistema, hasta el desarrollo y puesta en marcha del mismo.

A manera de resumen, los capítulos se encuentran distribuidos de la siguiente manera:

- a) En el capítulo 1, se presentan el concepto de Intranet y los objetivos principales de la misma, dentro de la Dirección General de Recursos Humanos.
- b) El capítulo 2, habla sobre los servidores de aplicación y su funcionalidad en una arquitectura de tres capas, como una solución para la creación de páginas web dinámicas.

- c) En el capítulo 3, se presenta la tecnología Java orientada al web conocida como "Java Server Pages" (JSP), así como un método para acceder a bases de datos relacionales por medio de la misma.
  
- d) Los capítulos 4, 5 y 6, tratan sobre el análisis y desarrollo del sistema de soporte técnico, desde su planteamiento inicial hasta las pruebas con el usuario final.
  
- e) Por último se presenta un apartado con las conclusiones obtenidas.

## **CAPÍTULO 1**

### **I N T R A N E T**

## 1.1 Intranet

De acuerdo con "GUIDE TO INTERNET TERMS" ([www.getnetwise.org](http://www.getnetwise.org)), se puede entender una intranet como "Una red privada dentro de una compañía y organización, la cual utiliza tecnología basada en internet. Esta red es únicamente para uso interno, por lo tanto, el acceso público no es permitido. Las compañías utilizan intranets para manejar proyectos, proveer información del empleado, distribuir datos e información, etc."

En mi experiencia, puedo definir a una intranet como un conjunto de aplicaciones basadas en tecnología web que permite la colaboración de individuos, grupos y/o empresas desde cualquier computadora que tenga conexión a Internet o a una red con protocolo de comunicaciones "Transfer Communication Protocol / Internet Protocol" (TCP/IP). Una intranet puede resolver, por ejemplo, el problema de la distribución de información para todos los empleados, pues se pueden publicar manuales, planes de trabajo, procedimientos, material de formación, anuncios, promociones etc. Estas publicaciones son accesibles para el usuario de la Intranet de forma inmediata y generan un ahorro considerable respecto a los métodos clásicos: panfletos, circulares, notas informativas, etc. Además, se tiene la ventaja de que cualquier actualización de datos es inmediata, no supone ninguna carga para la empresa como los métodos tradicionales (generalmente impresos) y puede estar disponible los 7 días de la semana las 24 horas.

Uno de los factores más importantes que buscan las empresas o dependencias al construir una intranet, es obtener mayor comodidad en el trabajo sin incrementar ni perder tiempo en localizar su información. Una Intranet define su grupo de trabajo como una entidad con capacidad de comunicar objetivos, proyectos, e incluso fomentar la cultura empresarial.

## 1.2 Intranet de la Dirección General de Recursos Humanos de la SHCP.

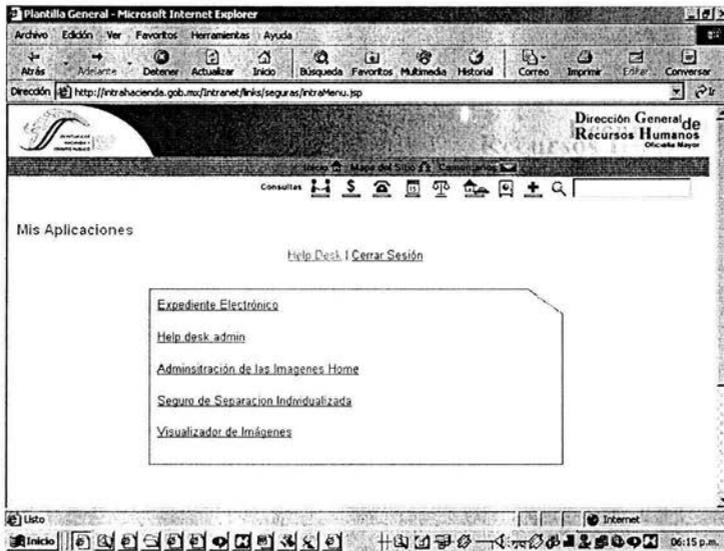
La intranet de la DGRH de la SHCP, nace como proyecto en julio del 2001 y comienza su materialización en septiembre del mismo año. El objetivo de esta intranet es informar y aumentar la colaboración entre todo el personal de la SHCP, tomando en cuenta las funciones de cada uno de los individuos que lo conforman. Con base en lo anterior, actualmente los usuarios de la intranet tienen acceso a diversos servicios e información, por ejemplo, pueden solicitar y consultar el estado de algún trámite, los descuentos en tiendas y espectáculos a los que tienen derecho por ser trabajadores de la SHCP, calendario de pagos, el directorio telefónico de los funcionarios, manejo de contenidos automatizado, observar el estado de algún proyecto, entre otros.



La imagen anterior, muestra la página principal de intranet (también conocida como "Home Site"). Como podemos observar, en la parte inferior derecha de la imagen, se muestran dos campos de texto, que tienen como etiquetas "usuario" y "password". Ahí es donde cualquier empleado de la Secretaría de Hacienda y Crédito Público, coloca sus datos de identificación

para acceder a las aplicaciones a las que tiene derecho de acuerdo a las políticas de seguridad implantadas. Es decir, existe una parte pública a la que cualquier persona que se dirija al sitio puede ver su contenido, y una parte privada que, de acuerdo a las políticas establecidas y a las funciones que cada persona desempeñe, le mostrará las aplicaciones o la información que tiene designadas.

En la imagen que se presenta a continuación, se muestra la página que se obtiene como resultado de una identificación con el sistema de manera exitosa (autenticación). En ella se observan las aplicaciones a las que un usuario en particular puede tener asociadas a su cuenta.



Se observa también, en la imagen anterior, una liga (hipervínculo o "link") que dice "Help Desk", y una más dentro del cuadro de texto sombreado que dice "Help Desk admin". Estos

hipervínculos son los que hacen referencia y que dirigen al usuario al sistema tratado en el presente trabajo.

### 1.3 Objetivos de la intranet dentro de la DGRH

Dentro de los objetivos principales, por lo cuáles se desarrolló esta intranet se tienen:

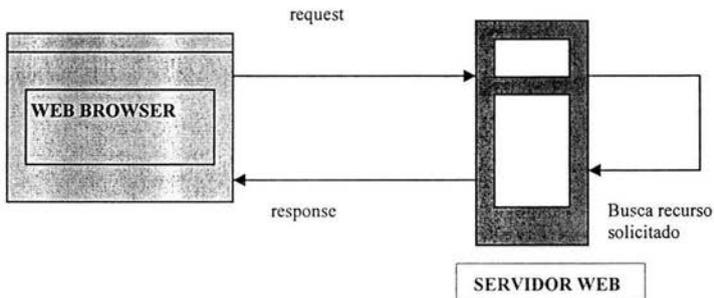
- Creación, publicación y difusión de información.
- Comunicación y colaboración en tareas en tiempo real.
- Elaboración e integración de bases de datos privadas y seguras.
- Foros de discusión, tele-conferencias e interacciones varias.
- Navegación y búsqueda de información dentro y fuera de la red interna.
- Formación de grupos de trabajos virtuales y dinámicos.
- Desarrollo de sistemas de "e-learning" para potenciar el uso de las Nuevas tecnologías aplicadas a la formación.

De acuerdo con lo anterior, se pretende, al tener una intranet dentro de la DGRH, disminuir los costos de desplazamiento de un lugar a otro para el intercambio de información o solicitud de servicios, acelerar y facilitar la comunicación entre áreas, compartir recursos a distancia, fomentar la transparencia en el acceso y uso de la información, así como aumentar la colaboración formal e informal entre el personal de la dependencia.

## 1.4 Tecnología utilizada para la creación de la intranet de la DGRH.

Como se dijo, una intranet esta basada en tecnología web. Esta tecnología tiene dos participantes principales, un explorador de Internet "web browser" y un servidor web con el que se comunicará. En el entorno de sistemas, este tipo de relación es llamada modelo "request/response".

El cliente, en este caso el explorador de Internet, hace la petición "request" de un recurso en específico (por ejemplo una página web) y el servidor responde "response" con el envío del recurso solicitado, obviamente, si este se encuentra disponible en el servidor. En caso de no encontrarse el recurso solicitado, se envía una respuesta con error producido.



Este modelo "request/response", utiliza el protocolo de comunicaciones "Hypertext Transfer Protocol" conocido como "http". Un protocolo es aquel que maneja, de alguna manera, el comportamiento de los participantes en su relación con los demás; un protocolo en el ámbito de redes, define como un dispositivo o programa se va a comunicar con otro (las famosas reglas de negocio). Http por lo tanto, define como se va a comunicar el explorador de Internet con el servidor web y viceversa.

El esquema "request/response" es útil cuando el contenido de las páginas en el servidor son de contenido estático, esto es, cuando están desarrolladas en lenguaje HTML y tecnologías que son ejecutadas en el lado del cliente como JavaScript, VBScript o las "Cascade Style Sheets". Estas páginas consisten en texto, imágenes y ligas hacia otros documentos.

Ahora bien, el internet y la intranet actualmente están orientados a las aplicaciones, que en su mayoría incorporan bases de datos. Por lo tanto, el contenido a desplegar dentro de las páginas debe ser generado, en gran parte, dinámicamente, pues el resultado a devolverse en el "response" dependerá de los datos enviados en el "request" y/o a los eventos que el usuario realice con el sistema (por ejemplo un clic en un menú). Aquí es donde los lenguajes de programación interpretados o compilados por un servidor de aplicaciones hacen su aparición y justifican su existencia.

La intranet de la DGRH fue desarrollada en el lenguaje de programación Java orientado al web conocido como Java Server Pages "JSP", utilizando como servidor de aplicaciones "Tomcat " en su versión 4.0. Este servidor de aplicaciones utiliza al servidor "Apache" para la publicación de código HTML.

## **CAPÍTULO 2**

### **SERVIDORES DE APLICACIÓN**

## 2.1 Antecedentes

Las necesidades de crear contenido dinámico basado en las peticiones y datos enviados por el cliente a un servidor web, hacen que los servidores de aplicación existan. Estos sitios con contenido dinámico son construidos mediante una combinación de lenguajes y tecnologías, por lo tanto, se consideran sistemas híbridos.

Podemos dividir a las tecnologías útiles para desplegar contenido dinámico en dos grupos: aquella que se ejecuta del lado del cliente "client-side technology" y tecnología que se ejecuta del lado del servidor "server-side technology".

Dentro de las tecnologías que se ejecutan del lado del cliente encontramos:

- a) Active X
- b) Applets de Java
- c) Scripts para ejecutarse por el explorador web (ej. JavaScript).
- d) HTML.
- e) XML (algunos navegadores ya soportan XML).

Dentro de las tecnologías que se ejecutan del lado del servidor tenemos:

- a) CGI's (Common Gateway Interfase)
- b) API propietarios de algún web server como ISAPI y NSAPI
- c) Active Server Pages (ASP).
- d) Java Server Pages (JSP) y Java Servlets
- e) JavaScript Server-Side
- f) PHP
- g) Cold Fusion

h) Server-side HTML y tecnologías similares.

Las tecnologías del lado del cliente son útiles cuando no hay que procesar información del servidor. Los objetos Active X, código javascript y applets de Java pueden generar contenido dinámico pero requieren de instalación del lado del cliente y a veces el tiempo de carga de los mismos puede ser bastante elevado. Además como estos últimos son instalados en la máquina del cliente, existen registros de que son potencialmente inseguros a pesar de que sean registrados por una entidad verificadora.

Algunos años atrás, la única solución para generar páginas dinámicas en el Web fue algo llamado Common Gateway Interfase (CGI). Los programas CGI proveían una forma relativamente simple de crear aplicaciones web que permitieran entrada de datos, consultas a bases de datos, y regresar algunos resultados de vuelta al explorador de Internet. Lo malo es que utilizan bastantes recursos del servidor (memoria y procesador) donde residen y se ejecutan.

Posteriormente Netscape y Microsoft desarrollaron API's propietarias (API es la abreviatura de "Aplication Programming Interfase"), capaces de generar código para responder a las peticiones vía web . Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador, por ejemplo, imprimir un caracter en pantalla, leer el teclado, escribir en un fichero de disco, etc.

La tecnología "server-side" que ha prevalecido en materia web son los llamados servidores de aplicación que interpretan ó compilan los lenguajes ASP, JSP, PHP, ColdFusion, etc.

## 2.2 Servidor de Aplicación

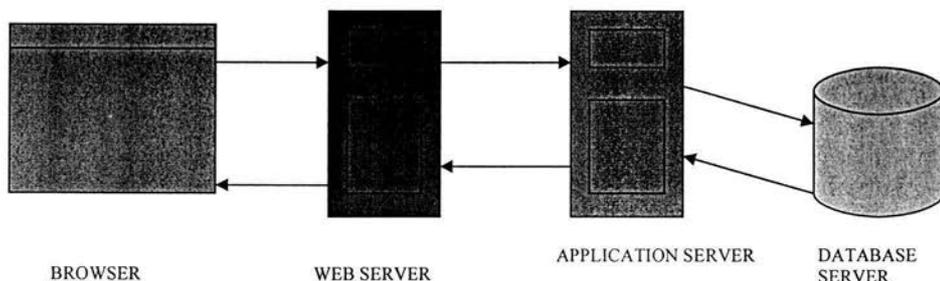
Un servidor de aplicación, en el sentido más básico, es aquel software dentro de una red distribuida que provee la lógica de negocio "business logic" para una aplicación. El servidor de aplicaciones, es frecuentemente visto, como un elemento dentro de una aplicación de tres o más capas, la cual consta de una interfase gráfica de usuario, un servidor de aplicaciones y un servidor de datos empresarial (comúnmente una base de datos).

El servidor de aplicaciones que utiliza la tecnología internet, es aquel que interpreta o compila un lenguaje, para dar como respuesta una página en HTML o en algún formato de visualización estándar (XML, WML, etc.) El servidor de aplicaciones al que haremos referencia en este trabajo, es aquel que nos genera respuestas HTML para poder ser publicadas mediante el protocolo de comunicaciones "Hyper Text Transfer Protocol" (http).

## 2.3 Arquitectura de Tres Capas.

A continuación se presenta una arquitectura de tres capas. Este tipo de arquitectura es comúnmente utilizada para la creación de sitios con contenido dinámico, y en ella aparecen los servidores de aplicación web.

En la imagen que se presenta a continuación, se muestra el cambio del esquema tradicional "request/response" para publicación de páginas estáticas, planteado anteriormente.



La forma en la que funciona este esquema es el siguiente: el cliente envía la petición al servidor web, éste busca el recurso solicitado, si lo encuentra, le envía al servidor de aplicaciones los datos enviados en el "request" para que la respuesta sea generada, en este caso, en HTML. Si esta respuesta requiere de alguna consulta a alguna base de datos ó de algún otro medio de almacenamiento, entonces se abrirá una conexión con ellos para obtener los datos requeridos que después serán manipulados y enviados en el formato que la respuesta lo requiera. Para fines de demostración, en la imagen anterior, fue separada la segunda capa en un servidor web y en un servidor de aplicaciones, pero en la práctica son vistos como una sola entidad.

## 2.4 Principales lenguajes utilizados por los servidores de aplicación.

Existen muchos lenguajes que son interpretados por servidores de aplicación, aquí nombraré a los líderes en el mercado comercial para el año 2003 en el desarrollo de aplicaciones web.

### **Microsoft Active Server Pages – ASP.**

Las Active Server Pages mejor conocidas como ASP's, son páginas que combinan lenguaje HTML, scripts (Javascript-Visual Basic Script) y componentes de tipo server-side. El lenguaje nativo de esta tecnología es "Visual Basic Script" que se ejecuta en el servidor.

Una ventaja de las ASP, es que puede utilizar componentes server-side que implementen una interfase de tipo COM (Especificación de componentes de Microsoft). Una desventaja de las Active Server Pages es que solo pueden ser utilizadas por el servidor de web/aplicaciones, "Internet Information Server" (IIS) de Microsoft, bajo un sistema operativo "Windows NT" o similar (Windows 2000 server, por ejemplo).

Existen emuladores para otros sistemas operativos y web servers pero al no utilizar a los objetos COM nativamente, se ve reducida la efectividad de estos.

### **PHP: Hypertext Preprocessor.**

PHP trabaja de manera similar a ASP. Contienen código HTML, scripts y código a ser interpretado. Al contrario que ASP, PHP es portable, pues existen versiones para Windows, Unix, Linux y para un gran número de servidores web como Apache e Internet Information

Server. Esta tecnología es libre y de código fuente abierto (open source). Es una buena opción como plataforma de desarrollo ya que a nivel mundial, está teniendo bastante apoyo por parte de los programadores.

### **Java Server Pages – JSP.**

Las Java Server Pages o JSP's, son la tecnología server-side que utilizan a Java como lenguaje de programación. Esta tecnología es similar a las anteriores, en el procedimiento de generación de páginas dinámicas. Una página JSP contiene código html, javascript, estilos css y código Java (beans, classes, scriptlets, etc.).

El motor de las páginas JSP está basado en los servlets de Java (programas en Java destinados a ejecutarse en el servidor), con la salvedad de que JSP es más fácil de comprender en su programación que los servlets, por lo que son más atractivas a nuevos programadores.

En JSP creamos páginas de manera parecida a como se crean en ASP o PHP. Generamos archivos con extensión .jsp que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor.

Antes de que un archivo .jsp sea funcional, el motor JSP lleva a cabo una fase de traducción de esa página en un servlet, implementado en un archivo class (Byte codes de Java). Esta fase de traducción se lleva a cabo habitualmente, cuando se recibe la primera solicitud de la página .jsp, aunque existe la opción de precompilar para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

La Intranet de la Dirección General de Recursos Humanos de la SHCP está desarrollada con tecnología JSP, por lo tanto la aplicación del sistema de soporte técnico también fue programada en esta tecnología para no caer en incompatibilidades. Cabe señalar que, de no haber sido este lenguaje especificado anteriormente por las políticas de la organización, hubiera existido la posibilidad de utilizar PHP para el desarrollo de este sistema.

## 2.5 Servidor de Aplicaciones Tomcat.

La Intranet de la DGRH reside en un servidor de aplicaciones que soporta JSP's llamado "Tomcat", en la versión 4.0.

"Tomcat" es un contenedor de servlets ("servlet container") que cumple con la Referencia Oficial de Implementación para Java Servlets 2.3 y Java Server Pages 1.2 desarrolladas por Sun (empresa que creó Java).

"Tomcat" es capaz de compilar Servlets y JSP, para que respondan a peticiones bajo el protocolo http y generar respuestas dinámicas que puedan ser interpretadas por clientes, como los navegadores de internet.

"Tomcat" está desarrollado bajo el esquema de código abierto ("Open Source") y tiene licencia de uso gratuita, gracias a la fundación Apache en su proyecto conocido como "Jakarta".

Mayor información se puede encontrar en el sitio de la organización Apache, <http://www.apache.org>.

## **CAPÍTULO 3**

### **JAVA SERVER PAGES**

### 3.1 Lenguaje de programación Java

El lenguaje de programación Java, fue lanzado por Sun Microsystems en 1995 e instantáneamente creó una nueva percepción de las posibilidades interactivas de la red. Java es un lenguaje de programación, que dicho por sus creadores, fue expresamente diseñado para usarse en el entorno de Internet y de redes distribuidas. Se diseñó para que se "pareciera" al lenguaje C++, pero que fuera más sencillo de usar, lo que implica una visión de la programación completamente orientada a objetos.

La gama de soluciones que se tienen con Java, es bastante amplia. Puede usarse para crear aplicaciones completas que corran en un sólo ordenador, o bien, aplicaciones que se distribuyan entre servidores y clientes en una red.

Las principales características de Java, expresados por SUN, son:

**Portabilidad.** Los programas creados son portátiles dentro de una red. El programa se compila en un código de bytes que puede ejecutarse en cualquier servidor o cliente que tenga una máquina virtual de Java. La máquina virtual de Java interpreta el código de bytes y lo transforma en código que puede se puede ejecutar en la plataforma donde reside. Esto significa que las diferencias individuales de las plataformas, como la longitud de las instrucciones, pueden ser reconocidas y adaptadas localmente mientras el programa se está ejecutando.

**Compatibilidad.** Ya no se necesitan versiones específicas de nuestros programas para cada plataforma, debido a que la Máquina Virtual de Java y el Java Runtime Environment (JRE) hacen este trabajo.

**Seguridad.** El código es "robusto", lo que significa que, a diferencia de los programas escritos en C++ y quizá algunos otros lenguajes, los objetos Java no pueden tener referencias a datos externos a ellos mismos. Esto asegura que una instrucción no pueda tener la dirección de almacenamiento de datos en otra aplicación o en el sistema operativo mismo; casos que provocarían que el programa y quizá el sistema operativo se cerraran o cayeran (lo que se conoce como "crash"). La máquina virtual de Java realiza una serie de verificaciones de cada objeto para garantizar su integridad.

**Objetos.** Java está orientado a objetos, lo que significa que, entre otras características, objetos similares pueden aprovechar el hecho de que son parte de la misma clase y heredar código común. Se piensa en los objetos como "sujetos" gramaticales, con los que un usuario puede relacionarse y no sólo los tradicionales "verbos" de procedimiento de la programación estructurada. Este lenguaje cumple con la herencia, el encapsulamiento y el polimorfismo que caracteriza a los lenguajes orientados a objetos.

**Facilidad.** Comparado con el C++, Java es más fácil de aprender y comprender. (Sin embargo, no es un lenguaje que podamos dominar en una noche).

Actualmente, los principales navegadores en el mercado incluyen una máquina virtual de Java, y casi todos los principales desarrolladores de sistemas operativos como IBM, Red Hat y SUN, han añadido compiladores de Java como parte de su oferta de productos.

El script de Java o JavaScript no debe confundirse con Java. El JavaScript, que se originó en Netscape, se interpreta a un nivel más alto y es más fácil de aprender que Java, pero carece de parte de la portabilidad de Java y la velocidad del código de bytes.

## 3.2 JAVA y las JSP.

Una página JSP es un archivo de texto simple, que consiste en contenido xhtml con elementos Java.

Los elementos que componen una página JSP son los siguientes:

**1. Directivas:** Dan información global de la página, por ejemplo, importación de clases, la página que maneja los errores, cuando la página forma parte de una sesión, etc.

**2. Declaraciones:** Sirven para declarar métodos y variables.

**3. Scripts de JSP:** Es el código Java incluido en la página; delimitado por "<% %>"

**4. Expresiones de JSP:** Formatea las expresiones como cadenas para incluirlas en la página de salida.

Estos elementos siguen una sintaxis muy parecida al xml, así se puede mantener la presentación totalmente separada de la lógica programada. Un buen ejemplo es la etiqueta con la que se crea una instancia de un java bean `<jsp:useBean .../>`.

Una directiva de JSP es una sentencia que proporciona la información del motor de JSP para la página que la pide. Su sintaxis general es `<%@ directiva {atributo ="valor"} %>` dónde la directiva debe tener un número de atributos. Algunas directivas en JSP 1.0 son:

Page: Información para la página.

Include: Incluye archivos completos palabra por palabra.

Taglib: La dirección de la librería de etiquetas ("tags") que se usará en la página.

La directiva Page es importante y como tal posee varios atributos como:

- *language="java"* Comunica al servidor, el lenguaje que va a ser utilizado en la página.
- *extends="package.class"* La variable extends, define la clase padre del servlet generado. Normalmente no es necesario utilizar otras que no sean las clases base del proveedor.
- *import="package.\*,package.class"* Sirve para especificar los paquetes y clases que se quieran utilizar.
- *session="true|false"* Por defecto "session" vale "true", manteniendo los datos de la sesión para la página.
- *isThreadSafe="true|false"* Por defecto vale "true", le hace señales al motor de JSP para que múltiples pedidos del cliente puedan ser tomadas como uno sólo.
- *info="text"* Información en la página a la que puede accederse a través del método Servlet.getServletInfo()
- *errorPage="pagina\_error"* Nombre de la página que cachará y manejará las excepciones generadas por errores.
- *isErrorPage="true|false"* Marca a la página que contiene esta directiva como una página que puede manipular excepciones.

Las declaraciones en JSP pueden definirse como una definición de variables y métodos, a nivel de clase, que son usados en la página. Un bloque de declaraciones típico sería `<%! declaración %>`.

Un ejemplo de declaración de script sería el siguiente:

```
<HTML>
<HEAD>
<TITLE>Página simple JSP</TITLE>
</HEAD>
<BODY>
<%! String strCadena = "x";
int intContador = 0;
%>
</BODY>
</HTML>
```

Se puede ver que se están declarando dos variables a utilizarse dentro de la página: una de tipo String de nombre strCadena inicializada con un valor de "x"; y la otra es una variable de tipo int (entero) con un valor inicial de 0.

Los scripts son bloques de código Java que residen entre los delimitadores `<%` y `%>`, y pueden acceder a cualquier variable o JavaBean que haya sido instanciado previamente. Existen también algunos objetos implícitos disponibles para los scripts desde entorno del Jsp-Servlet. Como se señala a continuación:

*Request:* Representa a la petición del cliente a través de un requerimiento por el protocolo HTTP. Es normalmente una subclase de la clase `HttpServletRequest`.

*Response:* Es la página JSP de respuesta generada por el request y es una subclase de `HttpServletResponse`.

*PageContext*: Los atributos de la página y los objetos implícitos necesitan ser accesibles a través del API, para permitir al motor de JSP compilar la página. Pero cada servidor tiene implementaciones específicas de cada uno de esos atributos y objetos. Para solucionar este problema, el motor de JSP utilizar la clase Factory para devolver la implementación de clase PageContext del servidor. Esta clase PageContext es inicializada con los objetos response y request y algunos atributos de la directiva de la página (errorpage, session ,buffer and autoflush) y facilita los otros objetos implícitos para la pagina de petición.

*Session*: El objeto de sesión HTTP asociado a la petición, utilizado cuando es necesario un almacenamiento temporal.

*Application*: El objeto que representa la salida de texto por pantalla.

*config* El objeto ServletConfig de la página.

*Page*: Es la forma que tiene la página para referirse a si misma. Se usa como alternativa al objeto this.

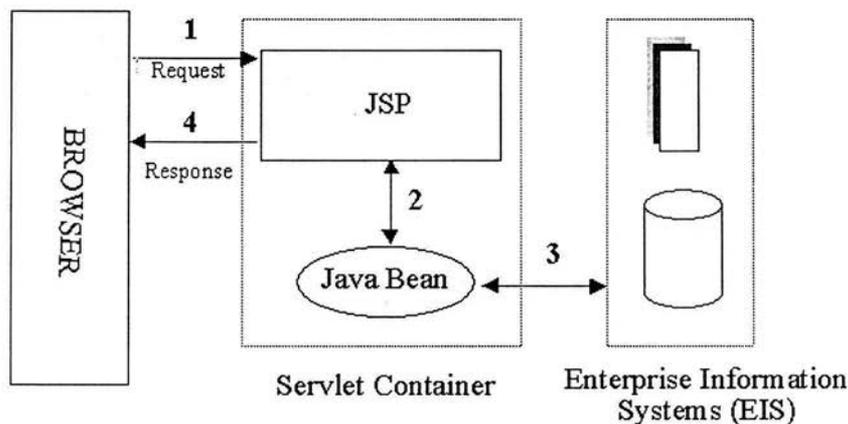
*Exception*: Es una subclase libre de Throwable que es pasada a la página que maneja los errores.

El siguiente fragmento de código, muestra como obtener el valor de un parámetro mediante el objeto request y como pasarlo a un objeto de tipo cadena para mostrarlo en pantalla; la página recibe un parámetro llamado "nombre" por medio de la URL ("Uniform Resource Locator"), en la forma: `http://dominio/mipagina.jsp?nombre=Daniel`.

```
<%  
String strNombre = request.getParameter("nombre");  
out.println(strNombre);  
%>
```

En el código anterior, se puede ver que el parámetro que espera recibir se llama "nombre", y este pudo ser enviado a través de un método get o post mediante un formulario en html. Este parámetro se almacena en una variable de nombre "strNombre" y la manera en la que se va a mostrar el valor de esta variable en el código de respuesta, es a través del método "out.println". Será común ver este tipo de código cuando se muestre el código utilizado en el desarrollo del sistema.

A continuación se muestra el mismo esquema de tres capas, señalado anteriormente, pero ahora desde la visión de JSP.



Entre las ventajas que JSP tiene ante sus competidores, SUN plantea las siguientes:

- Separación del contenido estático de contenido dinámico: Con los servlets, la lógica para la generación del contenido dinámico, es una parte intrínseca de los mismos y puede ligarse fácilmente con plantillas elaboradas estáticamente para el mostrado de los datos. La lógica para la generación del contenido dinámico se mantiene separada de la parte estática encapsulando esta en componentes Java.
- Escribe una vez y ejecútalo donde quieras: Las páginas JSP pueden ser portadas de una plataforma a otra fácilmente.
- El contenido dinámico puede ser generado en varios formatos: Nada obliga a que una página JSP deba tener un cierto formato. Consecuentemente, JSP puede generar contenido para los navegadores convencionales de HTML/DHTML, para dispositivos de mano como los PDA usando WML, o generar código XML para aplicaciones "business to business" (B2B).

Para poder comprender mejor el lenguaje de JSP, se sugiere la siguiente dirección de internet, donde se encuentran tutoriales, aplicaciones y ejemplos: [www.jspin.com](http://www.jspin.com)

### 3.3 JSP y las bases de datos.

Gran parte de la potencia de las páginas que son manejadas por un servidor de aplicaciones JSP, radica en la interacción que tienen estas con sistemas de bases de datos. Java contiene un API para la programación con bases de datos llamada JDBC ("Java Database Connectivity").

JDBC es una API pura de Java que se usa para ejecutar comandos de "Structured Query Language" (SQL). Suministra una serie de clases e interfases que permiten al desarrollador escribir aplicaciones que gestionen bases de datos.

La interacción típica con una base de datos como una primera aproximación de JDBC, consta de los siguientes cuatro pasos:

- Abrir la conexión a la base de datos
- Ejecutar consultas contra la base de datos
- Procesar los resultados
- Cerrar la conexión a la base de datos

Esto en código java, equivaldría a lo siguiente:

#### **Paso 1. Abrir la conexión a la base de datos.**

```
Connection conexion = DriverManager.getConnection("jdbc:odbc:Nombre_ODBC","usuario","password");
```

#### **Paso 2. Ejecutar consultas a la base de datos.**

```
Statement st = conexion.createStatement();
```

```
ResultSet rs = st.executeQuery("select dni,nombre,apellidos,edad from agenda");
```

#### **Paso 3. Procesar los resultados. En este caso los muestra en pantalla.**

```
while (rs.next()) { //mientras existan datos en el Resultset
```

```
out.println("DNI ->" + rs.getString("dni")); //Imprime DNI
```

```
out.println("NOMBRE ->" + rs.getString("nombre")); //Imprime Nombre
out.println("APELLIDOS ->" + rs.getString("apellidos")); //Imprime Apellidos
out.println("EDAD ->" + rs.getInt("edad")); //imprime Edad
}
```

#### **Paso 4 . Cerrar la conexión a la base de datos.**

```
rs.close(); //Cierro el objeto Results
st.close(); //Cierro el objeto Statement
conexion.close(); //Cierro el objeto Connection
```

El servidor de base de datos que existe en la SHCP está en una plataforma ORACLE. Esta base de datos tiene una instancia de nombre "avii" y se encuentra en un servidor cuya dirección ip es 10.145.0.218 en el puerto 1521.

Los parámetros para establecer la conexión son tres, uno es la URL dónde se encuentra la base de datos, el segundo es el nombre de usuario y tercero es el password para poder acceder a la base de datos.

Por lo tanto, la conexión para este caso en particular, se realizó de la siguiente manera:

```
String dbURL = "jdbc:oracle:thin:@10.145.0.218:1521:AVII";
String usuario = "user";
String pwd = "pass";
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection conexion = DriverManager.getConnection(dbURL,usuario,pwd);
```

Estas instrucciones requieren de la instalación del driver jdbc para bases de datos Oracle en el ambiente donde van a ser procesadas, en este caso, en los servidores de aplicación para desarrollo y producción.

Para ejecutar cualquier instrucción SQL contra la base de datos se utiliza el objeto Statement. Para crearlo tiene que llamarse al método de la superclase Connection llamada createStatement(). Este devolverá un objeto de tipo Statement que sirve para ejecutar las consultas en la base de datos. Con la siguiente instrucción, se observa como se crea un Statement.

```
Statement st = conexion.createStatement();
```

El paso siguiente es ejecutar estas consultas, pudiendo ser cualquiera que ejecute una acción para gestionar la base de datos. Por ejemplo, se pueden ejecutar instrucciones como UPDATE o DELETE utilizando el método "executeUpdate(SQLQuery)". Para el caso en particular de la instrucción SELECT, se necesita un objeto de tipo Resultset, en el cual se almacenarán en memoria los resultados obtenidos:

```
ResultSet rs=Statement.executeQuery(sentenciaSQL)
```

Una vez obtenido el Resultset, entonces se puede "navegar" secuencialmente por cada registro del conjunto de resultados obtenido, utilizando el método next().

Para obtener un valor del registro donde el cursor del Resultset se encuentre posicionado, se pueden utilizar diversos métodos, uno de ellos es getString("nombreCampo"). Este método nos devuelve un objeto de tipo String con el valor de lo que se encuentre en el campo "nombreCampo". Por ejemplo, rs.getString("nombre"), nos devolvería una cadena con lo que se encuentra en el campo "nombre" del Resultset de nombre "rs".

Algo bien importante, es cerrar todos los objetos que ya no se utilicen, pues, de no hacerlo, es probable que la memoria se sature, lo que haría que el sistema operativo donde se ejecute la

aplicación, pueda tener problemas serios. Los objetos de tipo Connection, Statement y ResultSet, todos tienen un método close() con el cuál pueden ser cerrados.

A continuación veremos un ejemplo en el cual se creará una tabla en una base de datos, utilizando sentencias SQL ejecutadas mediante JDBC.

```
import java.sql.*;

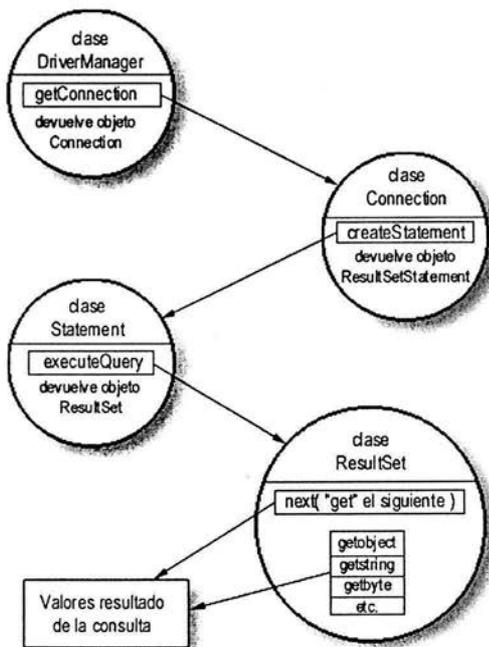
public class CrearTablaAgenda{
    public void CrearTablas(){
        Connection conexion = null;
        String strSQL = "";

        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            conexion = DriverManager("jdbc:odbc:NombreODBC","nombre_usuario","clave");
            Statement Estamento = conexion.createStatement();
            strSQL = "CREATE TABLE AGENDA (DNI VARCHAR(5),NOMBRE VARCHAR(30),APELLIDOS VARCHAR(30))";
            Estamento.executeUpdate(strSQL);
        } catch (SQLException excepcion_SQL) {
            System.err.println(excepcion_SQL.getMessage());
        } catch (ClassNotFoundException excepcion_ClassForName) {
            System.err.println(excepcion_ClassForName.getMessage());
        } catch (Exception excepcion_general) {
            System.err.println(excepcion_general.getMessage());
        }
        finally {

        try {
            if (conexion != null) {
                conexion.close();
            }
        }
    }
}
```

```
catch (SQLException sql_excepcion) {  
    System.err.println(sql_excepcion.getMessage());  
}  
}  
}  
  
public static void main(String[] argumentos) {  
    CrearTablaAgenda tablas_agenda = new CrearTablaAgenda(); //instanciamos el Objeto  
    tablas_agenda.CrearTablas(); //Creamos la tabla con el método CrearTablas()  
}  
}
```

La imagen que se presenta a continuación, muestra de manera gráfica el proceso que sigue la primera aproximación de JDBC para la gestión de bases de datos.



Como lo podemos notar, es relativamente sencillo el interactuar con bases de datos utilizando Java. Mediante el uso de JDBC es como podemos hacer consultas a una base de datos y mostrar la respuesta de estas consultas en el formato que se requiera, que en nuestro caso, el formato es una plantilla en html.

El mundo de JDBC es extenso como todo el mundo de java. Aquí sólo se mostró lo básico para poder interactuar con una base de datos. Para mayor información, se sugiere la siguiente página donde se encuentra un buen tutorial de JDBC:

<http://www.itapizaco.edu.mx/paginas/JavaTut/froufe/parte21/cap21-1.html>

## **CAPÍTULO 4**

### **ANÁLISIS DEL SISTEMA DE SOPORTE TÉCNICO**

## 4.1 Antecedentes

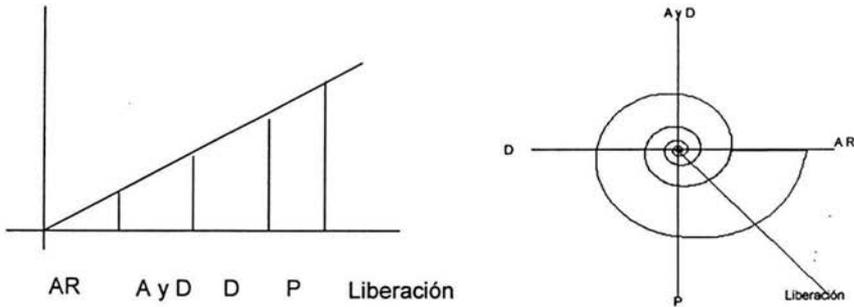
En los capítulos anteriores, se mostraron las bases tecnológicas que se tomaron en cuenta antes del desarrollo de este sistema. A continuación se expondrán las partes del análisis y del desarrollo empleados para la construcción de este sistema.

Este sistema fue desarrollado, en las siguientes etapas:

- a) Análisis de Requerimientos
- b) Diseño del Sistema
- c) Desarrollo del Sistema
- d) Pruebas con el usuario final
- e) Liberación.

Existen dos prácticas de desarrollo de sistemas, una que es de tendencia lineal y la otra que es en forma de espiral.

En la tendencia lineal, cada una de las etapas se realiza una sola vez para generar el producto, en este caso el sistema. En la tendencia en espiral se realizan el análisis de requerimientos, diseño, desarrollo, pruebas con el usuario, todas las veces necesarias hasta satisfacer sus requerimientos.



En el desarrollo con tendencia lineal, se le entrevista al usuario para saber lo que requiere y lo firma en una sola ocasión, lo cual es uno de los errores que comúnmente se tienen, ya que no se tiene un contacto frecuente con el usuario final hasta llegar a la etapa de pruebas. El problema existente con esta práctica es que el cliente no queda, en ocasiones, totalmente satisfecho ya que no fue revisando cada una de las fases del desarrollo.

En la práctica donde el desarrollo es de tipo espiral, se busca que el usuario final este revisando constantemente los avances del desarrollo desde su inicio hasta su fin, así en la etapa de liberación tendrá un producto conforme, que cubra totalmente sus expectativas.

El sistema presente, fue desarrollado utilizando la práctica en espiral para el desarrollo de sistemas, donde cada semana se fueron revisando los avances del proyecto y recabando firmas de conformidad.

El proyecto tuvo una duración de 4 semanas, por lo que al menos 4 veces se tuvo contacto con el usuario final antes de la puesta en marcha.

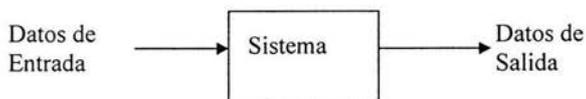
A continuación se hablará sobre cada una de las etapas del sistema, dando una breve introducción de cada una de ellas, y mostrando la aplicación de las mismas, dentro del desarrollo del sistema.

## 4.2 Etapa de Análisis de Requerimientos

Este es el primer paso en el desarrollo de un sistema, en este proceso el encargado de hacer el análisis se reúne con el cliente y/o usuario, e identifican las metas globales, se analizan las perspectivas del cliente, sus necesidades, sus requerimientos y otros puntos que puedan ayudar a la identificación y desarrollo del proyecto.

Antes de la reunión donde se expondrán los requerimientos, el cliente puede preparar un documento conceptual del proyecto, el cual deberá ser complementado con los resultados obtenidos por el analista en la entrevista con el cliente.

Cuando se quiere visualizar de una alguna manera, lo que vamos a construir en el caso de edificios, herramientas, aviones, máquinas, se crea un modelo idéntico, pero en menor escala. Sin embargo, cuando aquello que se construirá es un software, el modelo debe tomar una forma diferente. Los modelos se concentran en lo que debe hacer el sistema no en como lo hace, estos modelos pueden incluir notación gráfica, información y comportamiento del sistema. Todos los sistemas de software pueden modelarse como una transformación de información empleando una arquitectura entrada y salida (caja negra), como se muestra a continuación



El resultado de la entrevista con el cliente nos va a dar las especificaciones del sistema, las cuales serán concentradas en un documento que servirá como fundamento para la ingeniería

en el hardware, software y bases de datos. Este documento describirá la función y rendimiento del sistema, así como las dificultades que se pueden presentar en el desarrollo del mismo.

Un aspecto importante en la realización de un sistema, además de las especificaciones, es la viabilidad de la realización del mismo. La viabilidad y el análisis de riesgos en la producción de un sistema, están relacionados de muchas maneras, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce, por ello, se deben tomar en cuenta estas tres áreas:

1. **Viabilidad económica.**- Una evaluación de los costos de desarrollo, comparados con los ingresos netos o beneficios a obtener por el producto o sistema desarrollado.
2. **Viabilidad Técnica.**- Un estudio de funciones, rendimiento y restricciones que puedan afectar la realización de un sistema.
3. **Viabilidad Legal.**- Es determinar cualquier posibilidad de infracción, violación o responsabilidad legal en que se podría incurrir al desarrollar el sistema.

Este estudio de la viabilidad puede documentarse como un informe para los mandos superiores o los clientes, y sirve como respaldo ante contingencias.

### 4.3 Análisis de Requerimientos aplicado al Sistema de Soporte Técnico.

Para el desarrollo de este sistema fue necesario conocer la manera en la cual se llevaba a cabo el proceso de levantamiento y seguimiento de reportes. A través de la entrevista con el cliente, el proceso de atención de reportes de soporte técnico se efectuaba de la manera siguiente:

- a) El usuario que tenía un problema con su equipo de cómputo, se comunicaba por teléfono al área de soporte técnico.
- b) El área tomaba sus datos y los registraba en un formato en papel, anotando el nombre del usuario, ubicación del usuario y el problema que tenía.
- c) Si había algún especialista disponible, inmediatamente se hacía cargo del reporte generado para su atención, si no era así se mantenía en la cola de espera hasta que alguno de los especialistas lo estuviera.
- d) Una vez identificada la falla señalada en el reporte, se procedía a la reparación de la misma.
- e) Reparada la falla, el usuario firmaba la conformidad del servicio en el formato donde se anotaron previamente los datos del reporte, y a su vez, le daba una calificación a la calidad en el servicio obtenido.

Al analizar este proceso se detectaron ciertas oportunidades de mejora, como las que se muestran a continuación:

- a) No se efectúa un registro de las fallas más comunes. Al almacenarse estas fallas en alguna base de datos, se puede tener una base de conocimientos sobre fallas y posibles soluciones. Con ello se pueden reducir los tiempos de solución a un problema, dando un mejor servicio.

- b) No se lleva un control estricto de los reportes, lo cual llevaba a que el usuario reportara su falla muchas veces si el reporte inicial se extraviaba o traspapelaba. Al tener un monitor de estado de todos los reportes, este problema podía resolverse.
  
- c) No se lleva un control de los reportes que ya habían sido atendidos pero que todavía no habían sido concluidos, lo que provocaba que la reparación de alguna falla tardara mucho más tiempo del contemplado. El monitor de estado de reportes, también resuelve este problema.
  
- d) No se asigna un responsable para la atención de la falla, es decir, el reporte podía ser iniciado por alguien y concluido por otra persona sin que ninguno de los dos tuviera la responsabilidad de darle el debido seguimiento. Con la asignación a través del sistema se puede saber quién es el responsable de la finalización de un reporte.
  
- e) No se lleva un control de tiempo de cuanto tarda un especialista en reparar una falla. Al contar los días, los especialistas tardarán menos tiempo en resolver una falla ya que sabrán que su tiempo esta siendo contabilizado y pondrá en evidencia su efectividad.

Una vez analizado el proceso y después de informar lo anterior a los mandos superiores, se autorizó la realización de un sistema a mi cargo, que mejorara la gestión de los reportes de soporte técnico a través de la Intranet corporativa.

De la entrevista con el usuario final se llegaron a las siguientes conclusiones que conforman los requerimientos del sistema:

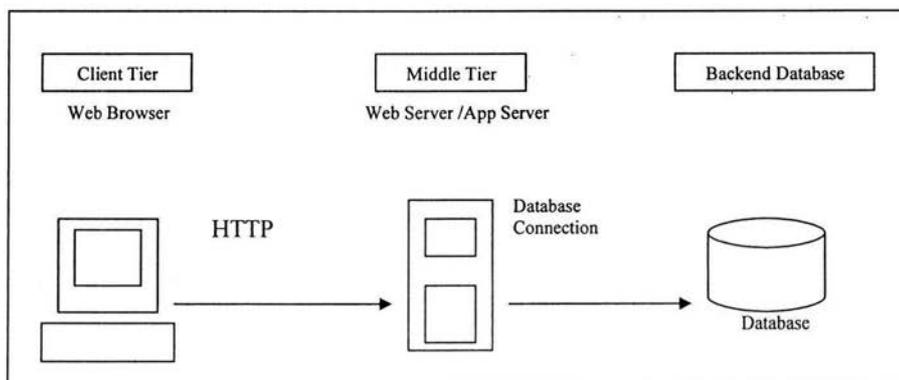
- a) Los usuarios deberán de levantar su reporte a través de la Intranet. Esto quiere decir que desde cualquier equipo conectado en la red local o VPN "Virtual Private Network", se puede levantar el reporte de cualquier equipo que no funcione correctamente.
  
- b) Los usuarios pueden tener uno o varios equipos registrados, por lo que ellos deberán tener la libertad de escoger entre cual de todos ellos desea generar su reporte.
  
- c) Todo el personal de soporte técnico puede ser elegible para atender y finalizar un reporte, esto incluye a Jefes de Departamento y Enlaces.
  
- d) En el área existen 3 niveles jerárquicos: jefatura, enlaces y operativos. La jefatura será el administrador y el encargado de asignar los reportes que van generándose a través del sistema. Todos los miembros del área de soporte técnico son elegibles para atender un reporte. El administrador puede hacer una asignación directa a cualquier miembro del área para la atención de un reporte, así como delegar reportes a los enlaces para que ellos puedan asignar al personal a su cargo, o bien, a ellos mismos para la atención de un reporte.
  
- e) Los reportes deberán pasar por cuatro estados, el primero es "inicio" que se da cuando el usuario genera su reporte, el segundo estado es "en proceso" cuando este reporte ya ha sido asignado a algún técnico y este ha empezado a atenderlo. El tercer estado se trata de "finalizado" que se da cuando el usuario esta conforme con la reparación de su falla y firma un reporte que deberá generar el sistema; y el último estado es "concluido" que este estado lo dará el administrador del sistema (jefatura o mandos superiores) cuando corrobore el papel firmado con lo reportado por el sistema.

- f) El sistema deberá contar los días que tarda un reporte desde su inicio hasta su finalización.
  
- g) El sistema será capaz de guardar una descripción de cómo fue solucionada la falla para contar en un futuro con una base de datos de conocimientos.

Estos requerimientos le fueron mostrados al usuario solicitante para su aprobación, y ya con ello, proceder al desarrollo del sistema.

#### 4.4 Selección del software para el desarrollo del Sistema.

Para poder tener un ambiente propicio tanto de desarrollo como de producción, la aplicación deberá contar con una base de datos en la capa final del sistema "backend" para almacenar todas las operaciones tanto de los usuarios como de los administradores. Además deberá contar en la capa del medio "middle tier" con un servidor web y uno de aplicación (que puede ser el mismo) para procesar las peticiones hechas por el explorador.



Antes de evaluar las alternativas de software en cada una de las capas descritas, cabe señalar que al momento de empezar el desarrollo, ya se contaba con una red corporativa de datos TCP/IP en la dependencia, por lo que no había que preocuparse por cuestiones de conectividad.

#### 4.5 Alternativas para el Sistema de Base de Datos.

Existen varias alternativas para almacenar la información del sistema "Enterprise Information Backend", entre las cuáles destacan:

- a) Archivos de texto con formato para almacenamiento de datos por ejemplo XML.
- b) Bases de datos relacionales como Oracle, Sybase, MySQL, etc.

La primera opción fue descartada, porque su implementación requiere de un diseño de cierta complejidad. Se tendría que diseñar como se almacenarán los datos en los archivos, la estructura en la que se organizarán y como serán utilizados en un futuro. Todas estas características ya se encuentran disponibles en las bases de datos relacionales.

Para este sistema fue seleccionada la base de datos relacional Oracle 9i, primero porque ya se contaba con una licencia en la organización y segundo, por ser uno de las más poderosos sistemas de bases de datos que existen en el mercado. Las características de esta base de datos en particular, proporcionará el soporte necesario al sistema en el "back end".

Para mayor información sobre esta base de datos y características, se sugiere el siguiente sitio de internet: <http://www.oracle.com>

#### **4.6 Alternativas de servidores para la capa del medio "middle tier".**

La capa del medio, será la encargada de crear en tiempo de ejecución, páginas codificadas en html como resultado de un proceso en esta misma capa. Por ejemplo, para obtener el listado de los reportes creados en cierta fecha, la capa del medio ejecutará la consulta correspondiente hacia la base de datos, obtendrá los resultados y generará el código html siguiendo una estructura específica, la cuál está definida por la interfaz de usuario correspondiente.

Las alternativas analizadas para la capa del medio fueron las siguientes:

- a) CGI's: Estos programas CGI pueden acceder a la base de datos usando los API's de cada lenguaje para el manejo de la misma.
  
- b) Lenguajes para servidores de aplicación "Server-Side Scripting Language": En este rubro se encuentran lenguajes como ASP, JSP y PHP. Estos lenguajes soportan objetos para acceder a casi todas las bases de datos relacionales en el mercado.

La opción elegida fue la de utilizar un lenguaje de tipo Server-Side Scripting Lenguaje, debido a que el sistema debe residir en el mismo servidor de aplicaciones de la intranet. Por ello, fue elegido JSP como lenguaje de desarrollo del sistema utilizando el mismo servidor "Tomcat" de la intranet, como servidor de aplicaciones.

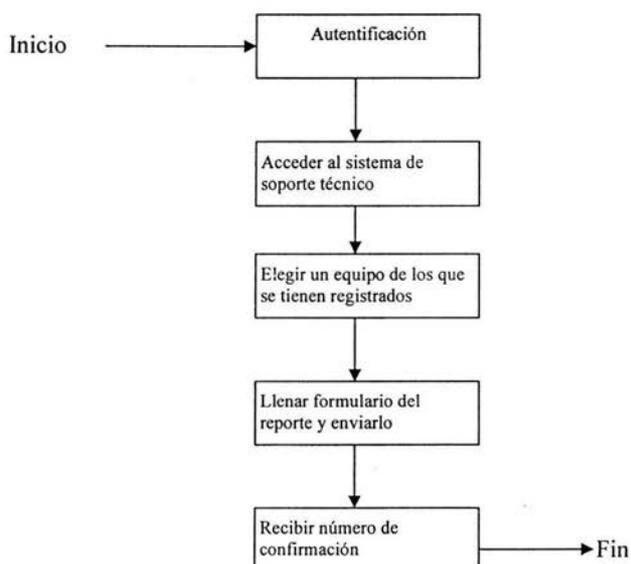
En cuanto al sistema operativo no se tuvo ninguna limitación ya que todas las versiones de software elegidas, están disponibles en todos los principales sistemas operativos (UNIX, WINDOWS, MAC OS, LINUX, etc.), pero por cuestiones de licenciamiento, se optó por utilizar

LINUX SUSE versión 7.3, que es el sistema operativo sobre el cual esta instalada la intranet de la DGRH..

Una vez definido el conjunto de software sobre el cual estará basado el sistema, se analizarán los procesos que se llevarán a cabo para el levantamiento de reportes y la gestión de los mismos.

#### 4.7 Proceso para la generación de un reporte de usuario.

Los usuarios para poder levantar un reporte por alguna falla en su equipo de cómputo, deberán autenticarse en la página principal de la intranet mediante nombre de usuario y contraseña. Si la autenticación es exitosa, los usuarios encontrarán una página donde se encuentra una liga con la leyenda "Help Desk". Al momento de hacer clic sobre esta liga, se le mostrarán al usuario los equipos que tiene registrados para que elija sobre cual desea levantar el reporte. Una vez elegido el equipo, se le presentará el formulario a llenar para declarar la falla o las fallas presentadas. Una vez completo el formulario, se almacenará el reporte y se le dará un número de confirmación al usuario para efectos de seguimiento o reclamaciones. El siguiente diagrama ilustra de manera gráfica lo expuesto en el párrafo anterior:



#### **4.8 Proceso para la gestión de los reportes del sistema de soporte técnico.**

Esta parte es la que se conoce como administrativa del sistema. Primero se definirán los roles que se encargarán de la gestión de los reportes.

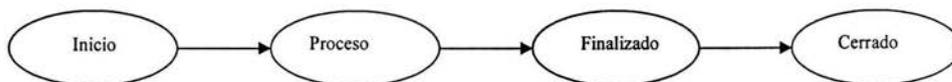
Existirán tres roles con sus respectivos privilegios listados a continuación:

- a) **ADMINISTRADOR:** Es el administrador del sistema de soporte técnico, tiene la función de dar mantenimiento a la base de datos de usuarios, de equipos y de técnicos. A su vez, podrá asignar reportes a verificadores o a cualquier técnico directamente para su atención. También podrá verificar el estado en el que se encuentren los reportes y concluir los mismos. En caso de no querer hacer asignación directa, el administrador podrá delegar reportes a sus verificadores para que ellos a su vez, asignen estos reportes al personal a su cargo. Hereda las características del rol OPERADOR.
  
- b) **VERIFICADOR:** Los usuarios con esta función podrán ver que reportes tienen asignados, para que ellos a su vez, asignen a un técnico a su cargo, como responsable de la atención de cada uno de ellos. También podrán ver que reportes tienen asignados para realizarlos como responsables finales de su atención. Así mismo, podrán visualizar el estado de sus reportes asignados. Hereda las características del rol OPERADOR.
  
- c) **OPERADOR:** Los usuarios con este rol, sólo pueden ver que reportes les fueron asignados, visualizar el estado de sus reportes y cambiarlos de estado.

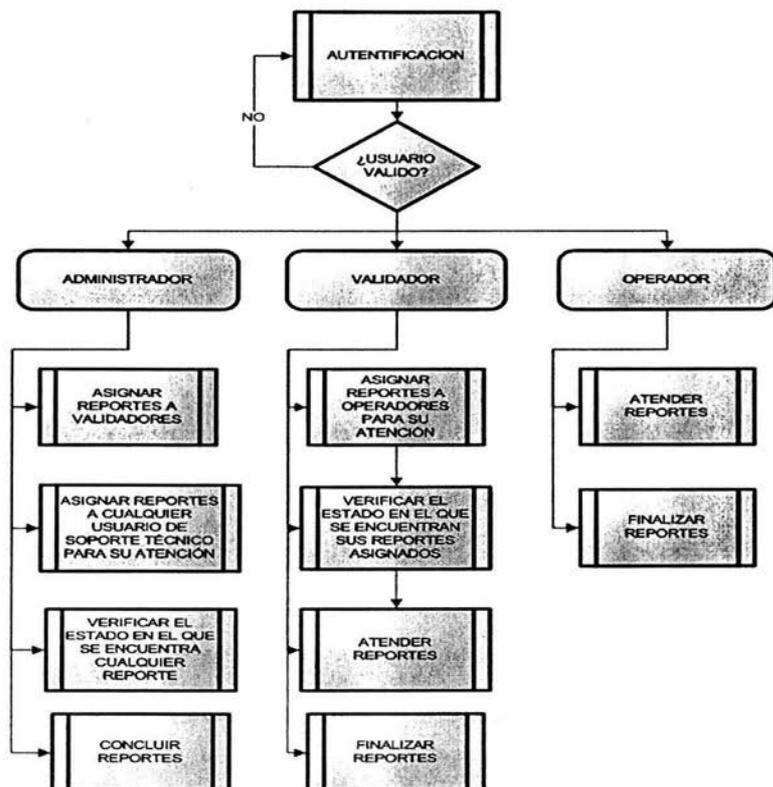
Los posibles estados en los que se puede encontrar un reporte son los siguientes: Inicio, Proceso, Finalización y Conclusión. Cabe señalar que no es posible que cambien de un estado

superior a uno menor, es decir, si un reporte esta en el estado de "Proceso", por ejemplo, no podrá regresar al estado de "Inicio", únicamente podrá tomar el valor inmediato superior que es "Finalización". Este comportamiento ocurre en todos los estados en los que puede estar un reporte.

#### Ciclo de Vida de un Reporte



A grandes rasgos, el proceso para la gestión de los reportes podemos mostrarlo gráficamente de esta forma:



Más adelante, se mostrará la codificación en JSP de cada uno de estos procesos.

#### 4.9 Diseño del esquema de la base de datos.

Para poder llevar a cabo todas las tareas y cumplir con los requerimientos del cliente, fue necesario el diseño y la creación de una base de datos como la que veremos a continuación.

El esquema de base de datos que utilizará este sistema de soporte técnico está compuesto de las siguientes tablas:

- a) HD\_CAT\_TECNICOS
- b) HD\_REL\_EQUIPO\_USUARIO
- c) HD\_REPORTES

La tabla HD\_CAT\_TECNICOS es el catálogo de todo el personal administrativo de este sistema y que puede atender a los reportes de los usuarios. En esta tabla también se almacenan los roles de cada técnico.

<b>HD_RFC_TECNICO</b>	Registro Federal de Contribuyentes y llave primaria del técnico	PK, NOT NULL VARCHAR2(15)
<b>HD_APE_PAT_TECNICO</b>	Apellido Paterno del Técnico	VARCHAR2(20)
<b>HD_APE_MAT_TECNICO</b>	Apellido Materno del Técnico	VARCHAR2(20)
<b>HD_NOMBRE_TECNICO</b>	Nombre del Técnico	VARCHAR2(20)

<b>HD_FUNCION</b>	Privilegios dentro de la parte administrativa del sistema, pueden ser: ADMIN, VALIDA y OPERA	VARCHAR2(10)
-------------------	--	--------------

La tabla HD\_REL\_EQUIPO\_USUARIO es donde están almacenados todos los equipos de la DGRH con sus respectivos responsables. De aquí se tomarán los datos para que el usuario elija el equipo sobre el cuál basará su reporte.

<b>HD_REL_INVENTARIO</b>	Número de inventario del equipo.	NOT NULL VARCHAR2(100)
<b>HD_REL_RFC</b>	RFC del usuario que tiene registrado el equipo y es una parte de la llave primaria compuesta.	PK NOT NULL VARCHAR2(12)
<b>HD_REL_SERIE</b>	Número de serie del equipo y segunda parte de la llave primaria compuesta.	PK VARCHAR2(100)
<b>HD_REL_MARCA</b>	Marca del equipo	VARCHAR2(100)
<b>HD_REL_MODELO</b>	Modelo del equipo	VARCHAR2(100)

La tabla HD\_REPORTES es donde quedarán registrados todos los datos correspondientes a los reportes de los usuarios.

<b>HD_CLAVE_REPORTES</b>	Clave del reporte generado por el sistema y llave primaria.	PK, NOT NULL NUMBER
<b>HD_HORA_REPORTES</b>	Hora en la que se generó el reporte.	NOT NULL VARCHAR2(15)

<b>HD_FECHA_REPORTE</b>	Día en que se generó el reporte.	NOT NULL VARCHAR2(15)
<b>HD_RFC_REPORTE</b>	RFC de la persona que generó el reporte. Llave foránea de la tabla HD_REL_EQUIPO_USUARIO.	NOT NULL VARCHAR2(15)
<b>HD_ESTADO_REPORTE</b>	Estado en el que se encuentra el reporte: INI-INICIADO; PRO-PROCESO; FIN-FINALIZADO; CON-CONCLUIDO.	NOT NULL VARCHAR2(5)
<b>HD_FECHA_INI_SERVICIO</b>	Fecha en la que se inició la atención del reporte.	VARCHAR2(15)
<b>HD_HORA_INI_SERVICIO</b>	Hora en la que se inició la atención del reporte.	VARCHAR2(15)
<b>HD_FECHA_FIN_SERVICIO</b>	Fecha de Finalización del reporte.	VARCHAR2(15)
<b>HD_HORA_FIN_SERVICIO</b>	Hora de Finalización del reporte.	VARCHAR2(15)
<b>HD_COMENTARIOS_USUARIO</b>	Comentarios del usuario acerca de cómo fue atendido su reporte.	VARCHAR2(2000)
<b>HD_CALIFICACION_SERVICIO</b>	Calificación que le da al usuario al servicio y a la atención por parte del técnico.	VARCHAR2(1)

<b>HD_VERIFICADO_POR</b>	RFC del técnico que fue asignado como verificador.	VARCHAR2(12)
<b>HD_CONCLUIDO_POR</b>	RFC de la persona que validó que estuviera bien concluido el reporte.	VARCHAR2(15)
<b>HD_FECHA_CONCLUSION</b>	Fecha en que se concluyó el reporte.	VARCHAR2(15)
<b>HD_DATOS_ADI</b>	Campo donde se almacenan datos adicionales por parte del que levanta el reporte.	VARCHAR2(2000)
<b>HD_CLAVE_TECNICO</b>	RFC del técnico que fue asignado para atender el reporte. Es llave foránea de HD_CAT_TÉCNICOS.	VARCHAR2(12)
<b>HD_IP</b>	Dirección IP de la máquina donde fue levantado el reporte a través de la Intranet.	VARCHAR2(20)
<b>SERV_TONER</b>	Se guarda 1 si requiere servicio de cambio de tóner en una impresora..	VARCHAR2(1)
<b>SERV_ALTACORREO</b>	Se guarda 1 si requiere alta de correo electrónico	VARCHAR2(1)
<b>SERV_CONFIMPRESORA</b>	Se guarda 1 si requiere configuración de impresora.	VARCHAR2(1)

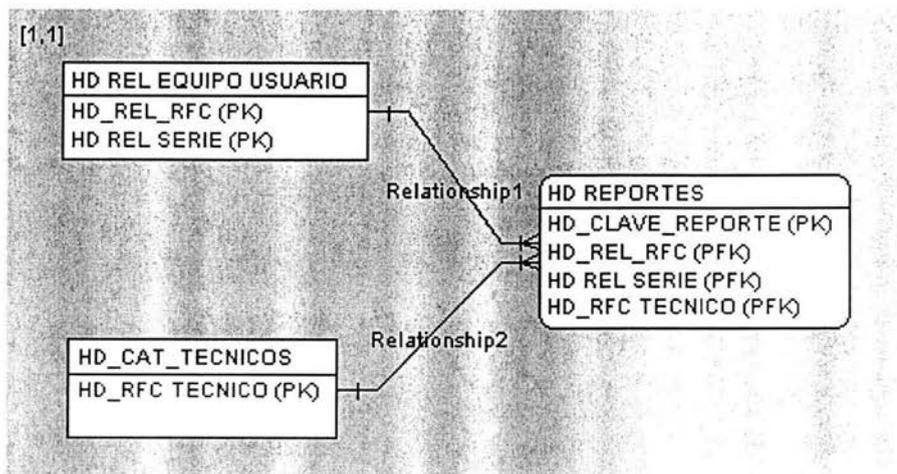
<b>SERV_MANTIMPRESORA</b>	Se guarda 1 si requiere mantenimiento alguna impresora.	VARCHAR2(1)
<b>SERV_INSTNODO</b>	Se guarda 1 si requiere instalación de un nodo de red.	VARCHAR2(1)
<b>SERV_INSTARJETARED</b>	Se guarda 1 si requiere instalación de tarjeta de red.	VARCHAR2(1)
<b>SERV_CONFTELNET</b>	Se guarda 1 si requiere configuración de una aplicación basada en telnet.	VARCHAR2(1)
<b>SERV_CONFENTORNORED</b>	Se guarda 1 si requiere configuración del entorno de red.	VARCHAR2(1)
<b>SERV_COMPARCHVOS</b>	Se guarda 1 si requiere compartir archivos a través de la red.	VARCHAR2(1)
<b>SERV_INSTSOFT</b>	Se guarda 1 si requiere instalación de algún software.	VARCHAR2(1)
<b>SERV_RESTSOFT</b>	Se guarda 1 si requiere restauración de algún software dañado.	VARCHAR2(1)
<b>SERV_CONFSOFT</b>	Se guarda 1 si requiere configuración de algún software.	VARCHAR2(1)

<b>SERV_DESCSOFT</b>	Se guarda la descripción del software que desea ser instalado, restaurado o configurado.	VARCHAR2(50)
<b>SERV_MANTPC</b>	Se guarda 1 si requiere mantenimiento de equipo de cómputo.	VARCHAR2(1)
<b>SERV_FALLATECLADO</b>	Se guarda 1 si existe una falla en el teclado.	VARCHAR2(1)
<b>SERV_FALLAMOUSE</b>	Se guarda 1 si existe una falla en el mouse.	VARCHAR2(1)
<b>SERV_FALLAAUDIO</b>	Se guarda 1 si existe una falla en el audio.	VARCHAR2(1)
<b>SERV_FALLAFLOPPY</b>	Se guarda 1 si existe una falla en el drive para discos de 3.5 pulgadas.	VARCHAR2(1)
<b>SERV_FALLACDROM</b>	Se guarda 1 si existe una falla en el lector CD-ROM.	VARCHAR2(1)
<b>SERV_FALLABOCINAS</b>	Se guarda 1 si existe una falla en las bocinas del equipo.	VARCHAR2(1)
<b>SERV_FALLAMONITOR</b>	Se guarda 1 si existe una falla en el monitor.	VARCHAR2(1)
<b>SERV_DESCOTRO</b>	Se guarda la descripción de otro problema que no este dentro del listado de opciones.	VARCHAR2(200)

<b>REPORTE_ANTERIOR</b>	Se almacena el número del reporte anterior en caso de que la falla haya sido causa de éste.	NUMBER(5)
<b>HD_SERIE_REPORT</b>	Se almacena el número de serie del equipo del que se hace el reporte. Es llave foránea de HD_REL_EQUIPO_USUARIO.	VARCHAR2(100)
<b>SERV_NOENCIENDE</b>	Se guarda 1 si el equipo no enciende.	VARCHAR2(1)
<b>HD_TECNICOS_EXTRA</b>	Se almacenan los RFC, si existen más técnicos asignados al reporte como auxiliares al titular.	VARCHAR2(300)
<b>HD_SOLUCION</b>	Se almacena como fue solucionado este reporte como base de datos de conocimientos.	VARCHAR2(400)
<b>HD_FECHA_RESPONSABLE</b>	Se almacena la fecha en la que se asignó el técnico responsable.	VARCHAR2(15)
<b>HD_HORA_RESPONSABLE</b>	Se almacena la hora en que se asignó al técnico responsable.	VARCHAR2(15)

<b>HD_CAP_ASESOR</b>	Se guarda 1 si requiere capacitación o asesoría en cuestiones de cómputo.	VARCHAR2(1)
----------------------	---	-------------

Las relaciones entre las columnas de las tablas son mostradas en la siguiente figura.



Es necesario señalar que en la tabla HD\_REL\_EQUIPO\_USUARIO se creó una llave primaria compuesta de dos campos, que son HD\_SERIE\_REPORTE y HD\_RFC\_REPORTE, debido a que existe la posibilidad de que un usuario pueda tener más de un equipo registrado por lo cual la combinación que nos puede dar una llave única es la combinación de estas dos columnas del registro.

Una vez diseñadas las tablas, mediante una herramienta llamada CASE STUDIO, se generó el archivo de exportación de datos DMP para ejecutarse en el entorno de la base de datos ORACLE 9i y así crear las tablas de manera física.

#### 4.10 Diseño de las interfases de usuario.

El tener una idea de cómo se verá el sistema funcionando, da una mejor orientación a la hora de programar el código. Por ello en este apartado veremos las interfases propuestas para este sistema, que en su totalidad, fueron desarrolladas con lenguaje HTML y Javascript en la validación de formularios. Estos lenguajes no causan conflicto con el servidor de aplicaciones ya que al momento de procesar una página JSP, separa el contenido dinámico del contenido estático. HTML y Javascript se ejecutan del lado del cliente en el explorador de Internet y JSP (código java) se ejecuta en el servidor de aplicaciones.

Como se dijo, el sistema se divide en dos partes principales: la parte de los usuarios para levantamiento de reportes y la parte administrativa utilizada por el personal que brinda el soporte técnico.

Las interfases de usuario para la parte del levantamiento de reportes, son las siguientes:

- a) La interfaz inicial de autenticación utilizada para la entrada al sistema. En este caso es la misma de la Intranet, que ya cuenta con un sistema de validación de usuarios en la parte inferior derecha.



- b) Interfaz para la elección del Sistema de Soporte Técnico. Esta interfaz será la misma que utiliza la intranet para mostrar las aplicaciones permitidas como resultado de una autenticación exitosa. Como podemos observar, aparecen dos hipervínculos, uno que dice "Help Desk" y el otro que dice "Help Desk Admin". El primero es el que aparece a todos los usuarios de la Intranet dentro de sus aplicaciones permitidas y que dando clic sobre él, nos permitirá levantar un reporte de soporte técnico. El segundo es el hipervínculo que aparece sólo a los usuarios que tienen permitida la parte administrativa de este sistema.



- c) Interfaz de inicio para el levantamiento de reportes. En esta interfaz al dar clic en "Continuar" nos llevará a la interfaz donde aparecerán todos los equipos que tiene el usuario registrados.

[Cerrar Sesión](#)

### Reporte de SERVICIO

Pulse en continuar para levantar un reporte de servicio al área de Soporte Técnico de la Dirección General de Recursos Humanos.

CANCELAR

CONTINUAR

Si desea consultar el estado de un reporte efectuado, [de un click aqui](#)

### VER ARTICULOS SOBRE TECNOLOGIA

- d) Interfaz donde aparecen los equipos registrados. Aquí aparecerán todos los equipos registrados para este usuario y dará clic en el botón de "Elegir" que aparezca del lado derecho de la descripción del equipo sobre el cual levantará el reporte.

The screenshot shows a navigation bar with the logo of the Secretaría de Hacienda y Crédito Público and the text "Dirección General de Recursos Humanos". Below the navigation bar is a search bar and a "Cerrar Sesión" link. The main content area displays the following information:

Filiación: RURD76123101  
Usuario: Daniel Ruiz Rodriguez  
Area: ADMINISTRACION DE PERSONAL

Usted tiene asignado el siguiente equipo, oprima ELEGIR en el que desea basar su reporte:

Marca	Lanix
Serie	110003552
Modelo:	Brain
Inventario	3724245

[ELEGIR](#)

- e) Interfaz para declarar las fallas o problemas existentes en el equipo elegido en la interfaz anterior. En esta interfaz se le solicitará al usuario que elija entre las opciones que se le presentan, aquellas que describan los problemas que el equipo en cuestión esté

presentando. Existe la opción de otros en caso de que ninguna de las opciones describa realmente el problema que tiene su equipo.

The screenshot shows a web form titled "Recursos Humanos" with a search bar and a "Cerrar Sesión" link. The form contains the following sections:

- Verifique que sus datos sean correctos:**
  - Rfc: RURD76123101
  - Nombre Completo: Daniel Ruiz Rodriguez
  - Marca del Equipo: Lanix
  - Serie Equipo: 110003552
  - Número Inventario: 3724245
  - Ubicación: EDIF. D P.B.
- Seleccione la(s) fallas o servicios aplicables al equipo:**
  - Capacitación y asesoría:
    - Requiero capacitación y/o asesoría
  - Correo Electrónico:
    - Alta de Correo Electrónico Interno/Externo
  - Impresoras:
    - Cambio de Tóner/Cartucho de Impresión
    - Configuración de Impresora (No puedo imprimir)
    - Mantenimiento de Impresora
  - Servicio de Red de Comunicaciones:
    - Instalación de nodo
    - Instalación de tarjeta de Red
    - Configuración del servicio de Telnet
    - Configuración de entorno de red
    - Compartir archivos, unidades de disco, etc.
  - Software (Programas):
    - Instalación de software
    - Restauración de software
    - Configuración de software
- Que software es:

- f) Interfaz de confirmación. En esta interfaz se notificará al usuario el alta de su reporte y se le otorgará un número de reporte para contacto o aclaraciones con el área de soporte técnico.

The screenshot shows a confirmation message in a Microsoft Internet Explorer browser window. The message text is:

Plantilla General - Microsoft Internet Explorer  
Su reporte ha sido enviado, a la brevedad uno de nuestros especialistas lo atenderá.  
El número de reporte es el 89, le recomendamos conservar esta clave.  
Cualquier duda, aclaración o sugerencia a la extensión 5382 y 7093 con Mario Martínez Medina.  
Cerrar Sesión

Con esta última, concluimos con las interfases de la parte a la que tiene acceso cualquier usuario registrado en la Intranet. Se observa que cada una de ellas hace referencia a un proceso dentro del diagrama presentado previamente.

Ahora, para la parte administrativa donde se llevará a cabo la gestión de los reportes, se tendrán las siguientes interfases:

- a) Interfaz de inicio después de la autenticación. Una vez que un usuario de la parte administrativa dio clic en el hipervínculo "HELP DESK ADMIN" (imagen de la interfaz b de las interfases de usuario), le aparecerá esta interfaz con el menú de opciones disponibles para el administrador, verificador u operador, según sea el caso. En la imagen que se presenta a continuación, se muestran las opciones de un administrador de este sistema. Los otros perfiles de usuario pueden reducir su número de opciones presentadas, pero no aumentarlas.

De click en la opción que desea:

**Alta reporte de usuario**  
**Ver todos los reportes**  
**Reportes generados no atendidos para asignación de verificadores**  
**Asignación directa de personal para reportes**  
**Verificación de finalización de reporte y cerrado del mismo**  
**Dar de alta artículo**

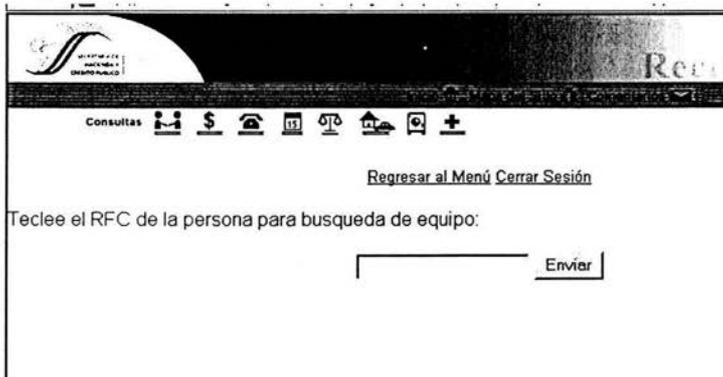
**Mantenimiento de la información del personal**  
**Mantenimiento de la información de equipos**

**Reportes en proceso que asigné al personal**  
**Mis reportes asignados**  
**Mis reportes asignados en proceso**

**ASIGNACION DE PASSWORDS**  
**Cerrar Sesión**

- b) Interfases para el alta de reporte de usuario. En el menú aparece como primera opción "Alta de reporte de usuario". Estas interfases hacen posible levantar un reporte, en caso

de que el usuario no llegara a contar con un equipo para levantar su reporte por la intranet y éste sea realizado por medio telefónico o presencial.



Aquí sólo presentamos la interfaz principal, debido a que las interfases subsecuentes son idénticas a las presentadas en la interfaz para el levantamiento de reportes por los usuarios, con la diferencia de que estas regresan al menú administrativo y no al menú común de la intranet, al confirmar la operación.

- c) Como segunda opción dentro del menú administrativo, encontramos "Ver todos los reportes". En esta interfaz el administrador podrá ver que reportes se han levantado, quiénes lo están atendiendo, el estado en el que se encuentran y cuántos días hábiles han tardado desde su creación hasta la fecha actual.

Fecha y Hora de Reporte	Fecha y hora de asignación de Responsable	Clave Reporte	Nombre del Usuario	Serie Equipo	Entregado por	Atendido por	Detalles	Estado	Usuario ocupado
240303 14:25	240303 16:56	21	Mario Antonio Martínez Medina	110003515	MAMM7/1012801	MAMM7/1012801	<a href="#">detalles</a>	FIN	91
240303 14:32	240303 14:33	23	Mario Antonio Martínez Medina	110003515	MAMM7/1012801	MAMM7/1012801	<a href="#">detalles</a>	FIN	91
240303 16:56	240303 16:57	24	Mario Antonio Martínez Medina	110003515	MAMM7/1012801	MAMM7/1012801	<a href="#">detalles</a>	FIN	91
240303 21:25	240303 21:27	25	Mario Antonio Martínez Medina	110003515	MAMM7/1012801	MAMM7/1012801	<a href="#">detalles</a>	FIN	91
D10403 14:54		27	Juana Castillo Cabrera	110002344			<a href="#">detalles</a>	INI	35
D10403 15:27		28	MARTINEZ MEDINA MARIO ANTONIO	77351			<a href="#">detalles</a>	INI	35

Como se pudo notar en la imagen anterior, existe un hipervínculo con la leyenda "detalles", al dar clic sobre esta palabra, aparecen los detalles del reporte elegido de la siguiente manera:

Clave reporte: 89  
 Fecha: 04/07/03 Hora: 15:21  
 Usuario: Daniel Ruiz Rodriguez Puesto:  
 Ubicación: EDIF. D P.B.  
 Equipo: Lanix Serie: 110003552 Inventario: 3724245  
 Fallas reportadas:

Alta de correo electrónico Instalación de Software Software:Real Player Falla en el teclado Falla en bocinas	Descripción de otro servicio:
---	-------------------------------

Datos adicionales: [Imprimir esta página](#)

Técnicos adicionales:

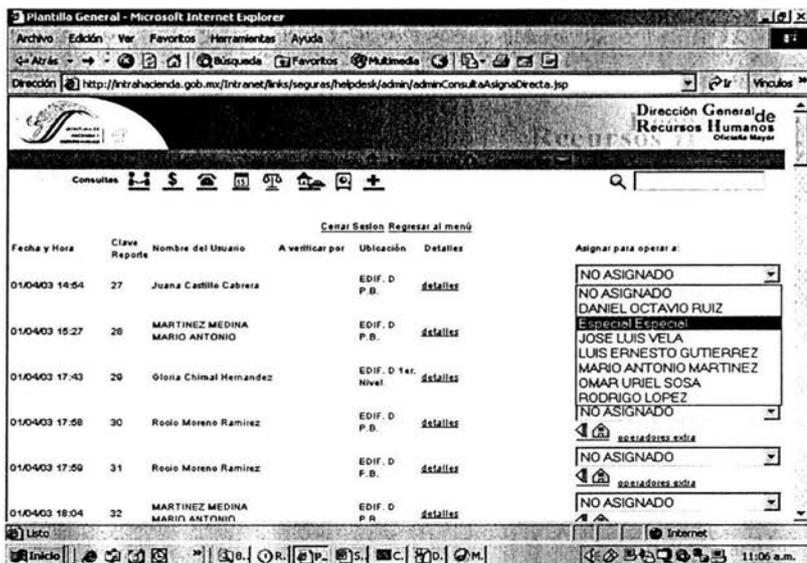
- d) La siguiente opción en el menú administrativo dice "Reportes generados no atendidos para la asignación de verificadores". En esta opción, que es sólo para aquellos que tiene privilegio "ADMIN", aparecen los reportes que aún no tienen una persona asignada

como responsable de la culminación del mismo y que tampoco tienen a una persona verificadora asignada como responsable del seguimiento de este reporte. En el ejemplo mostrado en la siguiente imagen, sólo se tienen a dos verificadores.

Fecha y Hora	Clave Reporte	Nombre del Usuario	Ubicación	Detalles	Responsable
01/04/03 14:54	27	Juana Castillo Cabrera	EDIF. D P.B.	detalles	NO ASIGNADO NO ASIGNADO JOSE LUIS VELA LUIS ERNESTO GUTIERREZ
01/04/03 15:27	28	MARTINEZ MEDINA MARIO ANTONIO	EDIF. D P.B.	detalles	NO ASIGNADO
01/04/03 17:43	29	Gloria Chimal Hernandez	EDIF. D 1er. Nivel.	detalles	NO ASIGNADO
01/04/03 17:58	30	Rocio Moreno Ramirez	EDIF. D P.B.	detalles	NO ASIGNADO
01/04/03 17:50	31	Rocio Moreno Ramirez	EDIF. D P.B.	detalles	NO ASIGNADO
01/04/03 18:04	32	MARTINEZ MEDINA MARIO ANTONIO	EDIF. D P.B.	detalles	NO ASIGNADO

Una vez asignado este reporte a un verificador, este puede asignar a su vez a un técnico como responsable para dar atención al mismo.

- e) La cuarta opción con la leyenda "Asignación directa de personal para reportes" es sólo válida para aquellos con privilegio de administrador. En esta interfaz, el administrador tiene la capacidad de asignar un responsable directamente a un reporte sin tener que pasar por un verificador. En este caso quedará asentado como verificador al administrador que hizo la asignación directa. Como podemos ver, a diferencia de la interfaz anterior, en esta aparece listado todo el personal de soporte técnico al que se le puede asignar un reporte.



- f) En la siguiente opción del menú “Verificación de finalización de reporte y cerrado del mismo”, el administrador puede observar que reportes ya han sido finalizados por el responsable, corroborar esto con la impresión del reporte firmado y así poder concluirlo dando clic en el botón con la leyenda “Concluir”. Opción válida para aquellos con privilegio de administrador.

Fecha y Hora de Inicio	Fecha y Hora de Finalización	Clave Reporte	Nombre del Usuario	Ubicación Atendido	Calificación Servicio	Comentarios del Usuario	Detalles	Concluir
240303 17:01	240303 17:03	21	Mario Antonio Martínez Medina	EDIF. D P.B. MARIO ANTONIO MARTINEZ	I		detalles	Concluir
240303 17:01	240303 17:03	24	Antonio Martínez Medina	EDIF. D P.B. MARIO ANTONIO MARTINEZ	E		detalles	Concluir
240303 21:27	240303 21:30	25	Mario Antonio Martínez Medina	EDIF. D P.B. MARIO ANTONIO MARTINEZ	A		detalles	Concluir

- g) Interfaz para la opción "Mantenimiento de la información del personal". En ella es posible dar de alta, actualizar o dar de baja registros dentro del catálogo del personal perteneciente al área de soporte técnico (HD\_CAT\_TECNICOS). Podemos ver que hay dos íconos en la parte derecha de la interfaz, ahí es donde el usuario al darle clic, procederá a eliminar un registro ó pasará a la interfaz para agregar uno nuevo mediante un formulario de entrada.

Consultas

[Agregar Tecnico](#) [Regresar al Menu](#)

APPELLIDO PATERNO	APPELLIDO MATERNO	NOMBRE TECNICO	NO FUNCION	RFC TECNICO	EDITAR/ELIMINAR
GUTIERREZ	DELGADO	LUIS ERNESTO	VALIDA	9UDL78000401	
LOPEZ	MENDEZ	RODRIGO	OPERA	LOMR78032901	
MARTINEZ	MEDIHA	MARIO ANTONIO	ADMIN	MAMM71012601	
RUIZ	RODRIGUEZ	DANIEL OCTAVIO	OPERA	RUR078123101	
Especial	Especial	Especial	ADMIN	SEK81072801	
SOSA	CASTRO	OMAR URIEL	OPERA	SOC078101001	
VELA	COLORADO	JOSE LUIS	VALIDA	VECL09070701	

- h) En la parte de "Mantenimiento de Equipos" es donde se pueden dar de alta, actualizar o dar de baja equipos del catálogo. El funcionamiento es similar a la interfaz anterior, con la diferencia de que en la primera tenemos el listado de todos los técnicos, en la segunda tenemos que buscar por número de serie al equipo que queremos eliminar ó editar.

- i) La opción con la leyenda "Reportes que asigné al personal", será la interfaz donde el administrador puede ver los reportes que ha asignado al personal de soporte técnico y ver en que estado se encuentran. Funciona como un monitor de reportes.

Dirección <http://intrahacienda.gob.mx/Intranet/links/seguros/helpdesk/admin/adminConsultaReportesPro.jsp>

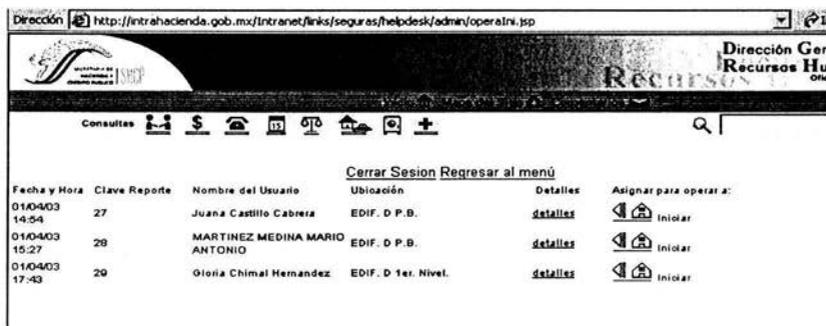
[Cerrar Sesión](#) [Regresar al menú](#)

Fecha y Hora de Asignación de Responsable	Clave Reporte	Nombre del Usuario	Atendido por:	Estado del reporte
08/01/03 14:14	1	Daniel Ruiz Rodriguez	LUIS ERNESTO GUTIERREZ	CON
09/01/03 14:13	3	Daniel Ruiz Rodriguez	JOSE LUIS VELA	CON
24/03/03 15:57	5	Sandra Pulido Galvan	MARIO ANTONIO MARTINEZ	CON
24/03/03 15:57	8	Daniel Ruiz Rodriguez	MARIO ANTONIO MARTINEZ	CON
24/03/03 15:58	10	Mario Antonio Martinez Medina	MARIO ANTONIO MARTINEZ	CON
24/03/03 20:44	11	Daniel Ruiz Rodriguez	MARIO ANTONIO MARTINEZ	CON
24/03/03 20:59	12	Daniel Ruiz Rodriguez	MARIO ANTONIO MARTINEZ	CON
24/03/03 20:59	13	Daniel Ruiz Rodriguez	MARIO ANTONIO MARTINEZ	CON
24/03/03 15:58	14	Daniel Ruiz Rodriguez	MARIO ANTONIO MARTINEZ	CON
17/03/03 20:51	15	Daniel Ruiz Rodriguez	MARIO ANTONIO MARTINEZ	CON
17/03/03 21:04	16	Juan Antonio Guevez Zurita	MARIO ANTONIO MARTINEZ	CON
24/03/03 15:58	17	Jose Luis Vela Colorado	MARIO ANTONIO MARTINEZ	CON
18/03/03 20:55	19	Mario Antonio Martinez Medina	MARIO ANTONIO MARTINEZ	CON
18/03/03 21:08	20	Mario Antonio Martinez Medina	MARIO ANTONIO MARTINEZ	CON
24/03/03 14:33	22	Mario Antonio Martinez Medina	MARIO ANTONIO MARTINEZ	CON
24/03/03 14:33	23	Mario Antonio Martinez Medina	MARIO ANTONIO MARTINEZ	CON
25/03/03 19:43	25	Mario Antonio Martinez Medina	MARIO ANTONIO MARTINEZ	CON

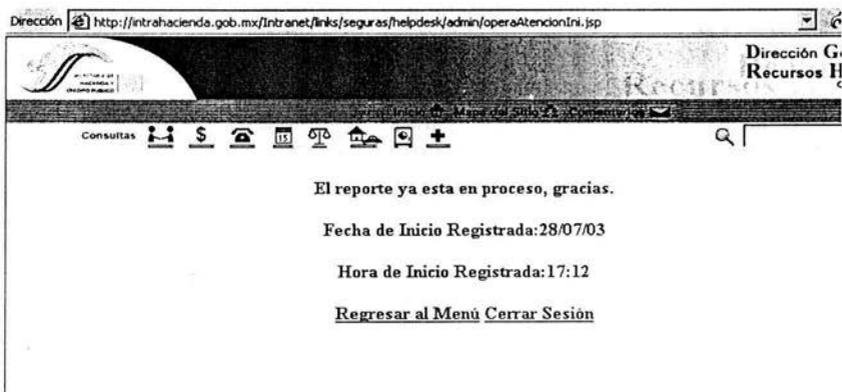
Las opciones que se indican a continuación, son válidas para cualquier tipo de privilegio.

- j) En la opción de "Mis reportes asignados", cualquier usuario de soporte técnico inscrito en el sistema, podrá ver que reportes le fueron asignados por el administrador ó bien por

el verificador. En ese momento él puede iniciar con la atención del reporte, dando clic en donde dice "Iniciar".



Después de dar clic en "Iniciar", le aparecerá la siguiente interfaz señalando que el reporte se encuentra en proceso.



- k) La siguiente opción con la leyenda "Mis reportes asignados en proceso", es donde, una vez iniciado un reporte, los usuarios de soporte técnico podrán ver los reportes que tiene en proceso, teniendo la posibilidad de concluir el reporte desde esta interfaz con solo dar clic en "concluir" como se señala en la siguiente imagen.





Con lo anterior, terminamos con las interfases diseñadas en un inicio con la finalidad de visualizar la estructura sobre la cual se distribuirán los datos, generados dinámicamente, por la capa del medio.

## **CAPÍTULO 5**

### **DESARROLLO DEL CÓDIGO EN JSP**

## 5.1 Antecedentes

Ahora que hemos visto la parte donde va a interactuar el usuario con el sistema, veremos lo que habrá detrás de estas interfases para que el sistema pueda funcionar con base en los requerimientos del cliente. Esto se conoce como la lógica de negocio.

Es preciso señalar que la técnica empleada en este caso para el mostrado de datos, fue la generación de tablas y formularios HTML dinámicos. Es decir al momento de tener un objeto Resultset como resultado de una consulta SQL vía JDBC, se genera una tabla (<table></table>) o un formulario (<form></form>), teniendo como base los resultados contenidos en el Resultset. En el caso de las tablas, tenemos la primera fila (<TR>) con el título de las columnas que vamos a mostrar, y las demás filas son generadas de acuerdo con la cantidad de registros en el Resultset.

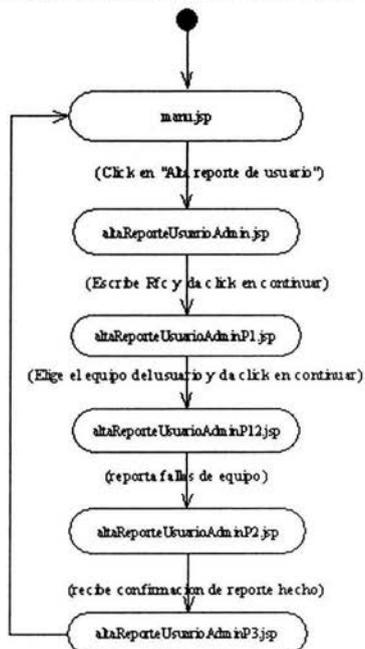
La utilización de la instrucción "while (ResultSet.next(){ //instrucciones})" , define el número de líneas a escribir dinámicamente en la tabla, pues no dejará de ejecutar las instrucciones que estén dentro de sus llaves, hasta llegar al final del objeto Resultset.

Otro punto importante, es señalar que Javascript se utilizó en algunas interfases para la validación de datos, es decir, al momento de enviar los datos de una página a otra por medio de un formulario, se verificaba que algunos de los datos estuvieran en el formato necesario por medio de rutinas escritas en este lenguaje. Como el sistema no depende totalmente de este tipo de validaciones, no se profundiza más en ello, pues sólo es un complemento de la lógica del sistema, no la parte fundamental del mismo.

## 5.2 Levantamiento de Reportes por los usuarios.

Para el desarrollo del código que soportará la lógica del sistema, se utilizaron como base, los diagramas mostrados en la parte del diseño. En este momento, se mostrará la primera etapa del proyecto y la más sencilla, que es el levantamiento de reportes por parte del usuario.

### ALTA REPORTE DE USUARIO



Siguiendo el diagrama, se observa que como primer paso, tenemos la autenticación del usuario. La autenticación de un usuario se hace a través de un módulo general de la intranet. Este módulo recibe los datos del usuario (login y password), hace una consulta sobre una base de datos de usuarios registrados y regresa una variable de sesión de nombre "usuario", que es válida mientras el usuario se encuentre activo dentro del sistema.

Esta variable de sesión almacena una clave con la cual se validarán las consultas en la base de datos, para obtener sólo la información que esté permitida a este usuario en particular.

Este módulo, retorna también, una variable de sesión llamada "flag" que al validar al usuario, toma el valor de "green" y que es evaluada en todas las páginas del sistema. Con ello cualquier usuario que no tenga la variable de sesión "flag" con el valor de "green" será dirigido a la página de autenticación de usuarios.

Para que la aplicación sólo deje acceder a usuarios autenticados, se incluye esta instrucción en todas las páginas:

```
<%@include file="../seguro1.jsp" %>
```

Esta instrucción quiere decir que la página incluirá el código de la página llamada "seguro1.jsp" en lugar donde se encuentre esta instrucción, que en este caso, nos es útil al principio de cada una de las páginas JSP que conforman el sistema.

La página de nombre "seguro1.jsp" contiene las siguientes líneas de código:

```
<%  
if(!((String)session.getAttribute("flag")).equals("green")){  
    response.sendRedirect("../index.jsp");  
    session.invalidate();  
}  
%>
```

Lo anterior indica, que al convertir la variable de sesión "flag" a un objeto de tipo String y al compararlo con la cadena de caracteres "green", esta devuelve un valor verdadero "true", entonces este usuario puede acceder al contenido de esta página; si por el contrario, devuelve falso "false", este usuario será enviado a la página de autenticación que en este caso es "index.jsp".

Una vez completo el proceso de autenticación, es posible acceder a la parte de generación de reportes de soporte técnico mediante la liga mostrada en la interfaz que se refiere al menú de aplicaciones permitidas (<a href="altaReporte.jsp">Help Desk </a>). Con esto cumplimos con el segundo paso en el diagrama.

El tercer paso en el diagrama consiste en elegir un equipo de los que el usuario tiene registrados. Para poder hacer esto se escribió el siguiente código:

```
<%  
  
// Generamos la conexión con la base de datos ORACLE  
  
boolean Recordset1_hasData=false;  
  
Driver DriverRecordset1 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();  
  
Connection ConnRecordset1 =  
  
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);  
  
//Generamos el objeto Statement con la instrucción SQL a ejecutarse sobre el objeto //Connection  
previamente creado.  
  
PreparedStatement StatementRecordset1 = ConnRecordset1.prepareStatement("SELECT Distinct  
HD_USUARIO_FILIACION, HD_USUARIO_NOMBRECORTO, HD_USUARIO_PUESTO,  
HD_EQUIPO_MARCA, HD_EQUIPO_SERIE, HD_EQUIPO_INVENTARIO, HD_USUARIO_AREA,  
UBICACION_USUARIOS, HD_EQUIPO_MODELO, HD_USUARIO_DESC_PUESTO, CLAVE_AREA,  
hd_ID,DESCRIPCION FROM DAP_SPG.HD_CAT_USUARIOS, DAP_SPG.HD_CAT_AREAS WHERE  
HD_USUARIO_FILIACION='"+session.getAttribute("usuario")+"' AND CLAVE_AREA=HD_USUARIO_AREA");
```

**//Generamos el objeto ResultSet ejecutando la consulta sql del Objeto Statement.**

```
ResultSet Recordset1 = StatementRecordset1.executeQuery();
```

```
boolean Recordset1_isEmpty = !Recordset1.next();
```

```
Recordset1_hasData = !Recordset1_isEmpty;
```

```
Object Recordset1_data;
```

```
int Recordset1_numRows = 0;
```

```
%>
```

```
<%
```

```
int Repeat1__numRows = -1;
```

```
int Repeat1__index = 0;
```

```
Recordset1_numRows += Repeat1__numRows;
```

```
%>
```

Como se puede observar, en este código primero se abre una conexión a la base de datos y después se genera un objeto Statement donde se almacena el texto de la consulta SQL. Al utilizar el método "executeQuery " del objeto Statement, se genera un objeto de tipo ResultSet que es el que guardará los resultados obtenidos por la consulta SQL.

El texto que corresponde a la consulta SQL, regresa como resultado, los datos registrados del usuario en la base de datos de soporte técnico. Estos datos son mostrados en una tabla de html generada dinámicamente con base en los registros almacenados en el objeto ResultSet llamado Recordset1. Para hacer esto se utilizó el código mostrado a continuación:

```
<% //mientras el objeto Recordset1 tenga datos
```

```
while (Recordset1_hasData) {
```

```
//genero formulario dinámicamente %>
```

```
    <form action="altareporteP2.jsp" method="post">
```

```
        <tr>    <td height="24" width="13%" valign="top"><b><font size="2" face="Arial, Helvetica, sans-
```

```
serif"><strong>Marca</strong></font></b></td>
```

```
        <td valign="top"><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
```

```
Recordset1.getObject("HD_EQUIPO_MARCA"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
```

```

        <td valign="top">&nbsp;</td>
        <td valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td height="24" width="13%" valign="top"><b><font size="2" face="Arial, Helvetica, sans-serif"><strong>Serie</strong></font></b></td>
        <td valign="top"><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data = Recordset1.getObject("HD_EQUIPO_SERIE"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
        <td valign="top">&nbsp;</td>
        <td valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td height="24" width="13%" valign="top"><b><font face="Arial, Helvetica, sans-serif" size="2">Modelo:</font></b></td>
        <td valign="top"><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data = Recordset1.getObject("HD_EQUIPO_MODELO"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
        <td valign="top">&nbsp;</td>
        <td valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td height="2" width="13%" valign="top"><b><font size="2" face="Arial, Helvetica, sans-serif"><strong>Inventario</strong></font></b></td>
        <td width="57%" height="2" valign="top"><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data = Recordset1.getObject("HD_EQUIPO_INVENTARIO"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
        <td width="15%" height="2" valign="top">
            <div align="left"><font size="2" face="Arial, Helvetica, sans-serif">
                <input type="submit" value="ELEGIR" name="submit">
                <input name="id" type="hidden" id="id" value="<%=(((Recordset1_data = Recordset1.getObject("HD_ID"))==null || Recordset1.isNull())?"":Recordset1_data)%>">
            </font></div>
        </td>
        <td width="15%" height="2" valign="top">&nbsp;</td>
    </tr>
    <tr>

```

```
<td height="24" width="13%"><b><font size="2" face="Arial, Helvetica, sans-serif"></font></b></td>
<td colspan="3"><font size="2" face="Arial, Helvetica, sans-serif">
<br>

</font></td>
</tr>
<tr>
<td height="2" colspan="4"><font size="2" face="Arial, Helvetica, sans-serif"></font></td>
</tr>
<tr>
<td height="2" colspan="4">&nbsp;</td>
</tr>
</form>
<%
Repeat1__index++;
Recordset1_hasData = Recordset1.next();
}
%> </table>
```

El código anterior genera una tabla como la que se observa en la siguiente figura, con los siguientes datos:

Filiacion: RURD76123101  
Usuario: Daniel Ruiz Rodriguez  
Area: ADMINISTRACION DE PERSONAL

Usted tiene asignado el siguiente equipo, oprima ELEGIR en el que desea basar su reporte:

Marca Lanix  
Serie 110003552  
Modelo: Brain  
Inventario 3724245

ELEGIR

Si el usuario tuviera más equipos registrados, con la estructura de control "while", se asegura que se muestren todos los equipos que el usuario tenga registrados y por ello contenidos en el objeto Resultset.

En el código se observan líneas con la siguiente instrucción, en varias ocasiones:

```
"<%=Recordset1_data = Recordset1.getObject("HD_EQUIPO_INVENTARIO")==null ||  
Recordset1.isNull()?"";Recordset1_data)%>".
```

Esta instrucción es utilizada para mostrar el contenido de un objeto Resultset en un campo o columna determinado, como una cadena de caracteres (String). En este caso se observa que se quiere mostrar el contenido del objeto Resultset en el campo "HD\_EQUIPO\_INVENTARIO". La instrucción indica que el Recordset1\_data va a ser igual a nulo, si el Resultset o el campo son nulos; si el Resultset en ese campo no está vacío, el objeto Recordset1\_data tomará el valor que éste contenga para ser mostrado. La notación <%=variable%> se utiliza en JSP para escribir el valor de una variable o de un objeto dentro de html.

El cuarto paso en el diagrama consiste en el llenado del formulario de reporte y su envío. Este paso fue cubierto mediante la generación de un formulario "form" en html formado en su mayoría por casillas de verificación conocidas como "checkboxes" y por campos de texto, como lo muestra la interfaz de usuario marcada con la letra b.

A continuación se mostrará un fragmento de código utilizado para la selección de fallas por parte del usuario:

```
<form action='recibeDatos.jsp' method=post>  
  <input type="checkbox" name="fallaFloppy">
```

```
<input type="text" name="otraFalla" value="">
<input type="submit" name="botonEnviar" value="Enviar">
</form>
```

En este código, se muestra un formulario "form" de html con dos opciones, una que es un cuadro de verificación "checkbox" y otro es un campo de texto. Existe un botón de nombre "botonEnviar" que es el encargado de ejecutar la acción de envío de datos hacia la página recibeDatos.jsp declarada en la propiedad "action" del formulario. No se colocaron todas las opciones que conforman el formulario, debido a que el código ocuparía demasiado espacio y el código las opciones restantes, en todos los casos es similar.

El quinto y último paso en el diagrama del levantamiento de reportes es el almacenamiento del reporte y envío de número de confirmación al usuario. En esta ocasión se hizo una inserción dentro de la tabla HD\_REPORTES y al final se le muestra al usuario el número de reporte generado al hacer su inserción. El código utilizado fue el siguiente:

```
<%
//Obtenemos el Driver
Driver DriverRecordset1 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();
Connection ConnRecordset1 =
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);
//Creamos el Statement con la sentencia SQL donde seleccionamos el número maximo anterior.
PreparedStatement StatementRecordset1 = ConnRecordset1.prepareStatement("SELECT MAX(
HD_CLAVE_REPORTE) FROM DAP_SPG.HD_REPORTES");
//Ejecutamos el Statement
ResultSet Recordset1 = StatementRecordset1.executeQuery();
boolean Recordset1_isEmpty = !Recordset1.next();
boolean Recordset1_hasData = !Recordset1_isEmpty;
Object Recordset1_data;
int Recordset1_numRows = 0;
```

```
%>

<%
if(Recordset1.getObject("MAX(HD_CLAVE_REPORTE)")==null)
    clave_next="0";
else
    clave_next=Recordset1.getString("MAX(HD_CLAVE_REPORTE)");

//if(clave_next.equals(""))
    //clave_next="0";
%>

<%
Driver DriverInsertaReporte = (Driver)Class.forName(MM_avii_DRIVER).newInstance();
Connection ConnInsertaReporte =
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);

String SQL="INSERT INTO DAP_SPG.HD_REPORTES (HD_CLAVE_REPORTE, HD_HORA_REPORTE,
HD_FECHA_REPORTE, HD_RFC_REPORTE,HD_ESTADO_REPORTE, HD_DATOS_ADI,HD_IP,HD_SERIE_REPORTE,
serv_TONER, serv_ALTACORREO, serv_CONFIMPRESORA, serv_MANTIMPRESORA, serv_INSTNODO,
serv_INSTTARJETARED, serv_CONFTELNET, serv_CONFENTORNORED, serv_COMPARCHIVOS, serv_INSTSOFT,
serv_RESTSOFT, serv_CONFOSOFT, serv_DESCSOFT, serv_MANTPC, serv_FALLATECLADO, serv_FALLAMOUSE,
serv_FALLAAUDIO, serv_FALLAFLOPPY, serv_FALLACDROM, serv_FALLABOCINAS, serv_FALLAMONITOR,
serv_DESCOTRO, REPORTE_ANTERIOR, SERV_NOENCIENDE,HD_CAP_ASESOR) VALUES
("+Integer.parseInt(clave_next)+1) +"," + hora +"," +fecha+"," +rfc+","INI"," +datosadi+"," +
request.getRemoteAddr()+"," +request.getParameter("serie")+"," + serv_toner+"," + serv_altacorreos +"," + serv_confimpresora
+"," +",";
SQL+=serv_mantimpresora+"," +serv_instnodo+"," + serv_insttarjetared+"," +",";
SQL+=serv_confnet+"," +serv_confentornored+"," +serv_comparchivos+"," +",";
SQL+=serv_instsoft+"," + serv_restsoft+"," +serv_confsoft+"," +",";
SQL+=serv_descsoft+"," + serv_mantpc+"," +serv_fallateclado+"," +",";
SQL+=serv_fallamouse+"," + serv_fallaudio+"," +serv_fallafloppy+"," +serv_fallacdrom+"," +",";
SQL+=serv_fallabocinas+"," +serv_fallamonitor+"," + serv_descotro+"," +",";
SQL+=reporte_anterior+"," +serv_noenciende+"," +hd_cap_asesor+"," +",";

```

```
//Ejecutamos la inserción del reporte
```

```
PreparedStatement InsertaReporte = ConnInsertaReporte.prepareStatement(SQL);
```

```
InsertaReporte.executeUpdate();
```

```
//Mostramos confirmación al usuario%>
```

```
<p align="center"><font size="2"><strong><font face="Arial, Helvetica, sans-serif">El número de reporte es el  
<%=Integer.parseInt(clave_next)+1%>, le recomendamos conservar esta clave.</font></strong></font></p>
```

El resultado generado con el código anterior se puede ver en la interfaz de usuario marcada con la letra "f" en la parte de levantamiento de reportes.

Con esto cubrimos lo marcado en el diagrama inicial, terminando así con la parte del levantamiento de reportes por los usuarios del sistema.

## 5.2 Parte Administrativa y Gestión de los Reportes.

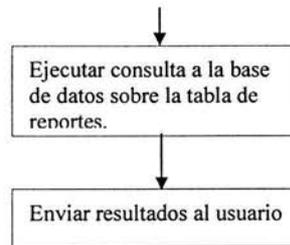
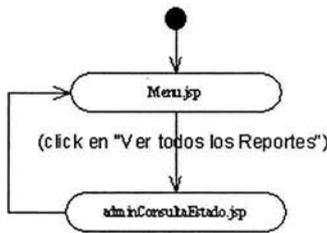
El menú de la parte administrativa del sistema de soporte técnico, está compuesto por las siguientes opciones:

1. Alta reporte de usuario
2. Ver todos los reportes
3. Reportes generados no atendidos para asignación de verificadores
4. Asignación directa de personal para reportes
5. Verificación de finalización de reporte y cerrado del mismo
6. Mantenimiento de la información de equipos
7. Reportes en proceso que asigné al personal
8. Mis reportes asignados
9. Mis reportes asignados en proceso

La opción número uno, correspondiente a "Alta reporte de usuario", contiene los mismo s pasos, interfases y código, que el levantamiento de reportes por los usuarios, con la única variación de que el administrador introduce el número clave del usuario para buscar sus equipos registrados, mientras que en el levantamiento normal, esta clave proviene de una variable de sesión como se explicó anteriormente.

La opción con el número dos en el menú que dice "Ver todos los reportes", está basado en los siguientes diagramas:

VER TODOS LOS REPORTE (OPCION ADMINISTRATIVA).



Para llevar al cabo estas tareas, se escribieron las siguientes líneas de código:

```

<%
Driver DriverRecordset1 = (Driver) Class.forName(MM_avii_DRIVER).newInstance();
Connection ConnRecordset1 =
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);
PreparedStatement StatementRecordset1 = ConnRecordset1.prepareStatement("SELECT * FROM
DAP_SPG.HD_REPORTES WHERE HD_ESTADO_REPORTES IN ('INI','PRO','FIN');
ResultSet Recordset1 = StatementRecordset1.executeQuery();
boolean Recordset1_isEmpty = !Recordset1.next();
boolean Recordset1_hasData = !Recordset1_isEmpty;
Object Recordset1_data;
int Recordset1_numRows = 0;
%>
<% while ((Recordset1_hasData)&&(Repeat1__numRows-- != 0)) { %>

<tr bgcolor="<%=color%>">
<td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_CLAVE_REPORTES"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_HORA_REPORTES"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_FECHA_REPORTES"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =

```

```
Recordset1.getObject("HD_RFC_REPORTE")==null || Recordset1.isNull()?"":Recordset1_data)%></font></td>
        <td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_ESTADO_REPORTE"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
        <td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_FECHA_INI_SERVICIO"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
        <td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_HORA_INI_SERVICIO"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
        <td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_VERIFICADO_POR"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
        <td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_CLAVE_TECNICO"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
        <td><font size="2" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("REPORTE_ANTERIOR"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
    </tr>
    <%
Repeat1__index++;
Recordset1_hasData = Recordset1.next();
}
%> </table>
```

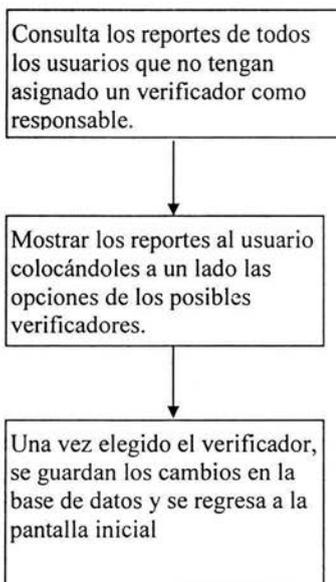
El código anterior nos muestra que primero se hace una consulta a la base de datos y después se genera dinámicamente la tabla en html con los datos obtenidos como resultado. A nivel de diseño se observará algo como lo que se muestra en la siguiente imagen:

The screenshot shows a web application interface for 'Recursos Humanos' (Human Resources). At the top, there is a header with the logo and the text 'Recursos Humanos Oficina Mayor'. Below the header is a navigation bar with a search bar and several icons. The main content area displays a table with the following columns: Clave, Hora de reporte, Fecha de reporte, Rfc del reporte, Estado, Fecha inicio de servicio, Hora de inicio, RFC Responsable, RFC Tecnico, and Reporte Anterior. The table contains one row of data with empty values in parentheses.

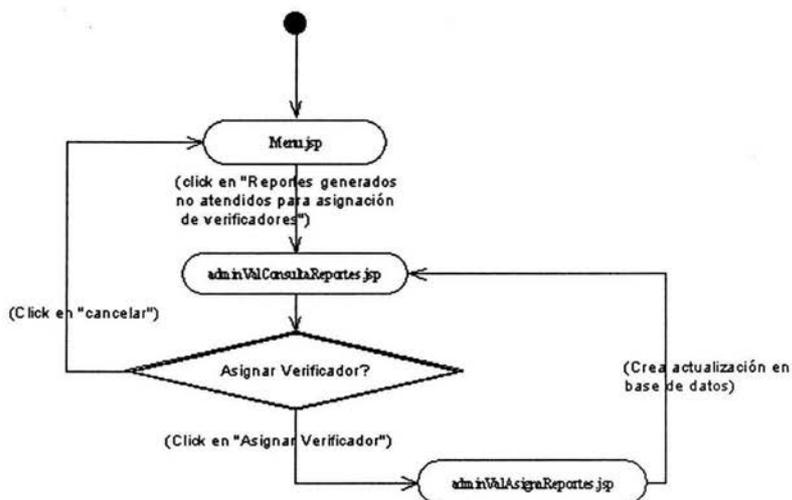
Clave	Hora de reporte	Fecha de reporte	Rfc del reporte	Estado	Fecha inicio de servicio	Hora de inicio	RFC Responsable	RFC Tecnico	Reporte Anterior
()	()	()	()	()	()	()	()	()	()

Al procesarse este código en el servidor de aplicaciones, este dará como resultado la interfaz de usuario con los datos obtenidos.

En la opción número 3 de título "Reportes generados no atendidos para la asignación de verificadores" se tienen los siguientes diagramas como base:



Reportes generados, no atendidos, para asignación de verificadores



Para consultar los reportes que no tienen verificador y para mostrar la lista de verificadores disponibles se utilizó el siguiente código:

```

<%
if(session.getAttribute("funcion").equals("OPERA"))
{ response.sendRedirect("../intraMenu.jsp"); }
%>
<%!
String sql="";
%>
<!-- Página observada por admin, concluye.valida-->
<%
if(session.getAttribute("funcion").equals("VERIFICA"))
sql="SELECT DISTINCT * FROM DAP_SPG.HD_REPORTES, DAP_SPG.HD_CAT_USUARIOS WHERE
HD_ESTADO_REPORTES='INI' AND HD_VERIFICADO_POR='"+ session.getAttribute("usuario")+"' AND
HD_EQUIPO_SERIE=HD_SERIE_REPORTES ORDER BY HD_CLAVE_REPORTES, HD_FECHA_REPORTES,

```

```
HD_HORA_REPORTE";
else
sql="SELECT DISTINCT * FROM DAP_SPG.HD_REPORTES, DAP_SPG.HD_CAT_USUARIOS WHERE
HD_ESTADO_REPORTE='INI' AND HD_EQUIPO_SERIE=HD_SERIE_REPORTE ORDER BY
HD_CLAVE_REPORTE, HD_FECHA_REPORTE, HD_HORA_REPORTE";
%>
<%
Driver DriverRecordset1 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();
Connection ConnRecordset1 =
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);
PreparedStatement StatementRecordset1 = ConnRecordset1.prepareStatement(sql);
ResultSet Recordset1 = StatementRecordset1.executeQuery();
boolean Recordset1_isEmpty = !Recordset1.next();
boolean Recordset1_hasData = !Recordset1_isEmpty;
Object Recordset1_data;
int Recordset1_numRows = 0;
%>

<%
Driver DriverRecordset2 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();
Connection ConnRecordset2 =
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);
PreparedStatement StatementRecordset2 = ConnRecordset2.prepareStatement("SELECT DISTINCT
HD_RFC_TECNICO, HD_NOMBRE_TECNICO||' '|| HD_APE_PAT_TECNICO FROM
DAP_SPG.HD_CAT_TECNICOS WHERE HD_FUNCION<>'SUPER'");
ResultSet Recordset2 = StatementRecordset2.executeQuery();
boolean Recordset2_isEmpty = !Recordset2.next();
boolean Recordset2_hasData = !Recordset2_isEmpty;
Object Recordset2_data;
int Recordset2_numRows = 0;
%>
```

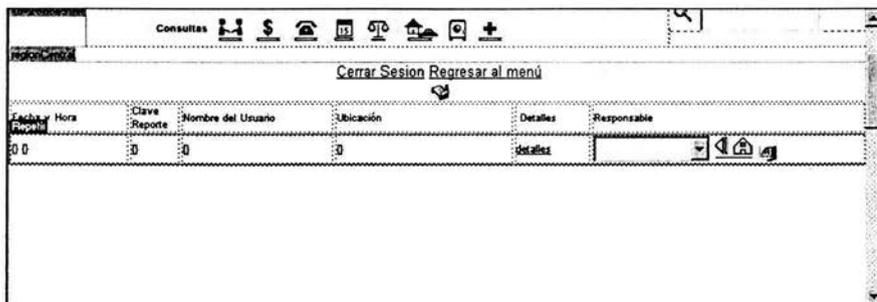
Para lograr que aparezcan los reportes con la opción para asignar verificador se utilizó el siguiente código:

```
<table width="772" border="0" cellspacing="0" cellpadding="0" height="99%">
<tr>
<td height="731" valign="top" bgcolor="#FFFFFF"> <table width="100%" border="0">
<tr>
<td width="5%"><font size="1" face="Arial, Helvetica, sans-serif">Fecha
y Hora</font></td>
<td width="5%"><font size="1" face="Arial, Helvetica, sans-serif">Clave
Reporte</font></td>
<td width="8%"><font size="1" face="Arial, Helvetica, sans-serif">Nombre
del Usuario</font></td>
<td width="6%"><font size="1" face="Arial, Helvetica, sans-serif">Ubicaci&oacute;n</font></td>
<td width="7%"><font size="1" face="Arial, Helvetica, sans-serif">Descripci&oacute;n</font></td>
<td width="14%"><font size="1" face="Arial, Helvetica, sans-serif">&nbsp;Datos
Adicionales </font></td>
<td width="14%"><font size="1" face="Arial, Helvetica, sans-serif">Detalles</font></td>
<td width="39%"><font size="1" face="Arial, Helvetica, sans-serif">Asignar
a</font></td>
</tr>
<% while ((Recordset1_hasData)&&(Repeat1__numRows-- != 0)) { %>
<tr>
<form action="asignaTecnico.jsp" method="post">
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_FECHA_REPORTE"))==null || Recordset1.isNull())?"":Recordset1_data)%>
<%=(((Recordset1_data = Recordset1.getObject("HD_HORA_REPORTE"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
```

```
Recordset1.getObject("HD_CLAVE_REPORTE")==null || Recordset1.isNull()?"":Recordset1_data)%></font></td>
    <td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_USUARIO_NOMBRECORTO"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
    <td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("UBICACION_USUARIOS"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
    <td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_DESCRIPCION_SERVICIO"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
    <td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_DATOS_ADI"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
    <td><font size="1" face="Arial, Helvetica, sans-serif"><a
href=javascript:abreDetalles(<%=Recordset1.getString("HD_CLAVE_REPORTE")%>);>detalles</a></font></td>
    <td>
        <font size="1" face="Arial, Helvetica, sans-serif">
        <select name="tecnico" id="tecnico">
            <option value="" <%=("".toString().equals((((((((Recordset1_data =
Recordset1.getObject("HD_CLAVE_TECNICO"))==null ||
Recordset1.isNull())?"":Recordset1_data)).toString()).toString()).toString()).toString())?"SELECTED":"")%>>Sin
            Asignar</option>
            <%
while (Recordset2_hasData) {
%>
            <option
value="" <%=(((Recordset2.getObject("HD_RFC_TECNICO")!=null)?Recordset2.getObject("HD_RFC_TECNICO"): "")%
>" <%=(((Recordset2.getObject("HD_RFC_TECNICO")).toString().equals((((((((Recordset1_data =
Recordset1.getObject("HD_CLAVE_TECNICO"))==null ||
Recordset1.isNull())?"":Recordset1_data)).toString()).toString()).toString()).toString()).toString())?"SELECTED":"")
%>
><%=(((Recordset2.getObject("HD_NOMBRE_TECNICO||||HD_APE_PAT_TECNICO")!=null)?Recordset2.getObject("
```

```
HD_NOMBRE_TECNICO|||HD_APE_PAT_TECNICO").""}%></option>
    <%
Recordset2_hasData = Recordset2.next();
}
Recordset2.close();
Recordset2 = StatementRecordset2.executeQuery();
Recordset2_hasData = Recordset2.next();
Recordset2_isEmpty = !Recordset2_hasData;
%>
    </select>
    <%if(session.getAttribute("funcion").equals("VALIDA")){%>
    <input type="submit" value="Asignar">
    <%}%>
    <input name="clave_reporte" type="hidden" id="clave_reporte" value="<%=(((Recordset1_data =
Recordset1.getObject("HD_CLAVE_REPORTE"))==null || Recordset1.isNull())?"":Recordset1_data)%>">
    </font></td>
    </form>
</tr>
    <% Repeat1__index++;
Recordset1_hasData = Recordset1.next();
}
%>
```

Con este código se genera una tabla dinámica que generará formularios para cada uno de los registros encontrados. Todos estos formularios enviarán datos a la página "asignaTecnico.jsp" donde se actualizarán los registros con los datos del verificador. La vista a nivel de diseño del código anterior, mostrará algo similar a la siguiente figura:



Como se puede notar, existe un elemento de tipo menú con la leyenda "Responsable". Este menú es generado dinámicamente en la parte del código donde se hace referencia a un objeto Resultset de nombre Recordset2. Este Resultset contiene el resultado de una consulta de todos los posibles verificadores que existen dentro de la población de técnicos. Así, cada vez que se agregue o borre un registro de un verificador, se reflejará automáticamente el cambio en el menú. En la propiedad "value" de las opciones del menú, se escribe la clave del responsable, mientras que en la propiedad "text" de las opciones del menú, se escribe el nombre completo para ser mostrado. La propiedad "value" contiene el valor que vamos a enviar a través del formulario para que sea recibido por la página de actualización de registros "asignaTecnicos.jsp".

Para poder actualizar en la base de datos al reporte con su verificador, se escribió el siguiente código para la página "asignaTecnicos.jsp":

```
<%
Driver DriverPrepared1 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();
Connection ConnPrepared1 =
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);
PreparedStatement Prepared1 = ConnPrepared1.prepareStatement("UPDATE DAP_SPG.HD_REPORTES SET
HD_CLAVE_TECNICO="" + Prepared1__tecnico +
"",HD_FECHA_RESPONSABLE="" + fecha + "",HD_HORA_RESPONSABLE="" + hora + "" WHERE
```

```
HD_CLAVE_REPORTE="+ Prepared1__reporte + " ");
```

```
Prepared1.executeUpdate();
```

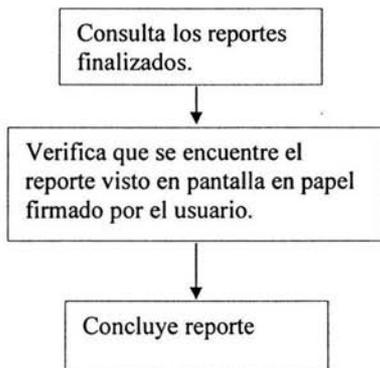
```
%>
```

En el código anterior, se puede notar, que la consulta cambió con respecto a las que se mostraron anteriormente. Ahora se utilizó la instrucción "UPDATE" de SQL, en lugar de la instrucción "SELECT".

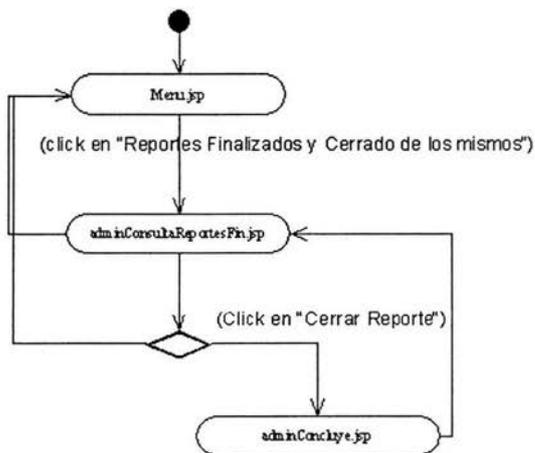
La siguiente opción en el menú administrativo, "Asignación directa de personal para reportes", es aquella donde el administrador puede asignar a un técnico directamente. El tratamiento de esta opción es similar a la opción anterior, únicamente que aquí se buscan los reportes que no tienen asignado a un técnico como responsable, y al momento de la actualización, se coloca la clave del administrador como verificador. Por otro lado, el menú en lugar de contener a los verificadores disponibles, tendrá un listado de todos los técnicos registrados en la base de datos. El código fuente es similar a la opción anterior.

En la opción de "Verificación de finalización de un reporte y cerrado", se consultan todos los reportes que el técnico ha señalado como finalizado ("FIN" en la base de datos). Estos reportes son lo que generan una hoja para ser firmada por el usuario que solicitó el servicio. Al administrador le llega esta hoja firmada y verifica que este reporte finalizado se encuentre registrado en el sistema. Si esto fue correcto, entonces dará clic en la opción de concluir que aparece al costado de ese reporte en particular en la interfaz de usuario correspondiente.

Los diagramas utilizados para esta opción, son los siguientes:



CONSULTA DE REPORTES FINALIZADOS Y CERRADO DE LOS MISMOS



El código que nos da como resultado cuáles reportes ya han sido finalizados, y están es espera de ser concluidos, es el siguiente:

```
<%
```

```
Driver DriverRecordset1 = (Driver)Class.forName(MM_avij_DRIVER).newInstance();
```

```
Connection ConnRecordset1 =
```

```
DriverManager.getConnection(MM_avij_STRING,MM_avij_USERNAME,MM_avij_PASSWORD);
```

```
PreparedStatement StatementRecordset1 = ConnRecordset1.prepareStatement("SELECT DISTINCT
HD_CLAVE_REPORTE, HD_RFC_REPORTE, HD_ESTADO_REPORTE, HD_FECHA_INI_SERVICIO,
HD_HORA_INI_SERVICIO, HD_FECHA_FIN_SERVICIO, HD_HORA_FIN_SERVICIO,
HD_COMENTARIOS_USUARIO, HD_CALIFICACION_SERVICIO, HD_USUARIO_NOMBRECORTO,
HD_APE_PAT_TECNICO, HD_NOMBRE_TECNICO, UBICACION_USUARIOS, HD_DESCRIPCION_SERVICIO
FROM DAP_SPG.HD_CAT_TECNICOS, DAP_SPG.HD_CAT_USUARIOS, DAP_SPG.HD_REPORTES WHERE
HD_ESTADO_REPORTE='FIN' AND
HD_REPORTES.HD_CLAVE_TECNICO=HD_CAT_TECNICOS.HD_RFC_TECNICO AND
HD_EQUIPO_SERIE=HD_SERIE_REPORTE");
ResultSet Recordset1 = StatementRecordset1.executeQuery();
boolean Recordset1_isEmpty = !Recordset1.next();
boolean Recordset1_hasData = !Recordset1_isEmpty;
Object Recordset1_data;
int Recordset1_numRows = 0;%>
```

Este código nos devuelve todos aquellos reportes que tienen en el campo de estado del reporte la palabra 'FIN', dispuestos en una tabla html. A nivel de diseño, lo anterior se verá de esta manera:

Fecha y Hora de Inicio Reporte	Fecha y Hora de Finalización	Clave Reporte	Nombre del Usuario	Ubicación	Rendimiento	Calificación Servicio	Comentarios del Usuario	Detalles	Concluir
00	00	0	0	0	00	0	0	detalles	Concluir

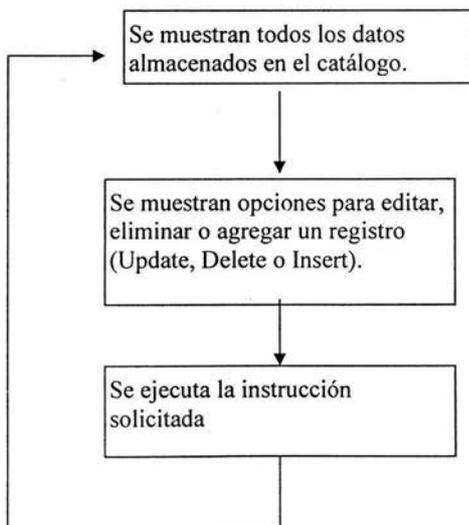
En este caso, el administrador corrobora que los reportes marcados como finalizados, estén respaldados por el reporte firmado por el usuario solicitante. Si este documento se encuentra,

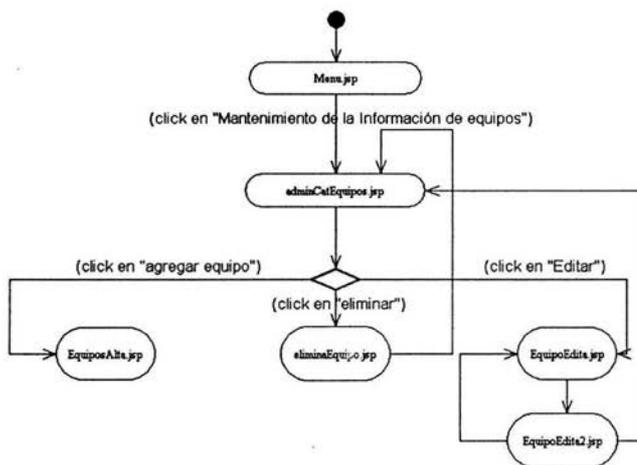
entonces se le dará a este reporte el estado de concluido y se cambiará la palabra 'FIN' en el estado del reporte por la palabra 'CON'. Para ello se escribió el siguiente segmento de código:

```
<%  
Driver DriverPrepared1 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();  
Connection ConnPrepared1 = DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);  
PreparedStatement Prepared1 = ConnPrepared1.prepareStatement("UPDATE DAP_SPG.HD_REPORTES SET  
HD_ESTADO_REPORTES ='CON', HD_CONCLUIDO_POR='"+ Prepared1__usuario + "',HD_FECHA_CONCLUSION='"+fecha+""  
WHERE HD_CLAVE_REPORTES ='"+ Prepared1__reporte + "'");  
Prepared1.executeUpdate();  
%>
```

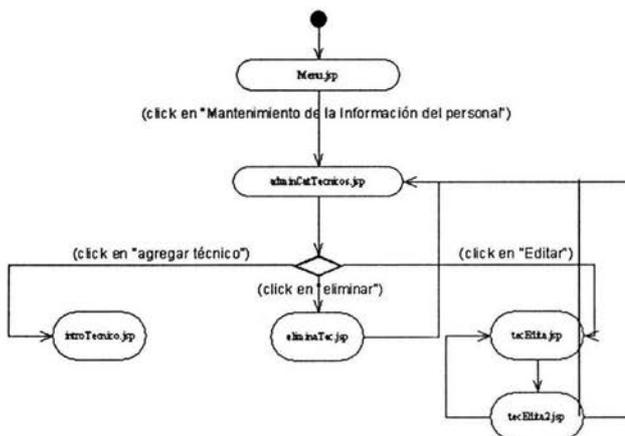
Las siguientes dos opciones en el menú administrativo son para el mantenimiento del catálogo de equipos y del catálogo de usuarios.

Como se indican, estas opciones constan de módulos para altas, bajas y cambios en la información de los catálogos de usuarios. Los diagramas genéricos de estos módulos, son los siguientes:





MANTENIMIENTO DEL CATALOGO DE TECNICOS



Se utilizaron las interfases mostradas para el mantenimiento del catálogo de usuarios y de equipos, mostradas en el capítulo anterior.

El código fuente para realizar las operaciones de inserción es el siguiente:

```
<%
// *** Insertar Record
if (request.getParameter("MM_insert") != null && request.getParameter("MM_insert").toString().equals("form1")) {

    MM_editDriver    = MM_sismed_DRIVER;
    MM_editConnection = MM_sismed_STRING;
    MM_editUserName  = MM_sismed_USERNAME;
    MM_editPassword  = MM_sismed_PASSWORD;
    MM_editTable     = "DAP_SPG.HD_CAT_USUARIOS";
    MM_editRedirectUrl = "";
    String MM_fieldsStr =
"HD_USUARIO_FILIACION|value|HD_USUARIO_NOMBRECORTO|value|HD_USUARIO_PUESTO|value|HD_EQUI
PO_MARCA|value|HD_EQUIPO_SERIE|value|HD_EQUIPO_INVENTARIO|value|HD_USUARIO_AREA|value|UBICA
CION_USUARIOS|value|HD_EQUIPO_MODELO|value|HD_USUARIO_DESC_PUESTO|value|HD_ID|value";
    String MM_columnsStr =
"HD_USUARIO_FILIACION|,none,"|HD_USUARIO_NOMBRECORTO|,none,"|HD_USUARIO_PUESTO|,none,"|HD
_EQUIPO_MARCA|,none,"|HD_EQUIPO_SERIE|,none,"|HD_EQUIPO_INVENTARIO|,none,"|HD_USUARIO_AREA
|none,none,NULL|UBICACION_USUARIOS|,none,"|HD_EQUIPO_MODELO|,none,"|HD_USUARIO_DESC_PUEST
O|,none,"|HD_ID|none,none,NULL";

    // create the MM_fields and MM_columns arrays
    java.util.StringTokenizer tokens = new java.util.StringTokenizer(MM_fieldsStr,"|");
    MM_fields = new String[tokens.countTokens()];
    for (int i=0; tokens.hasMoreTokens(); i++) MM_fields[i] = tokens.nextToken();

    tokens = new java.util.StringTokenizer(MM_columnsStr,"|");
    MM_columns = new String[tokens.countTokens()];
    for (int i=0; tokens.hasMoreTokens(); i++) MM_columns[i] = tokens.nextToken();

    // set the form values
    for (int i=0; i+1 < MM_fields.length; i+=2) {
```

```
MM_fields[i+1] = ((request.getParameter(MM_fields[i])!=null)?(String)request.getParameter(MM_fields[i]):"");
}

// append the query string to the redirect URL
if (MM_editRedirectUrl.length() != 0 && request.getQueryString() != null) {
    MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf("?") == -1)?"?":"&") + request.getQueryString();
}
}
%>
<%

// *** Insert Record: construct a sql insert statement and execute it

if (request.getParameter("MM_insert") != null) {

    // create the insert sql statement
    StringBuffer MM_tableValues = new StringBuffer(), MM_dbValues = new StringBuffer();
    for (int i=0; i+1 < MM_fields.length; i+=2) {
        String formVal = MM_fields[i+1];
        String elem;
        java.util.StringTokenizer tokens = new java.util.StringTokenizer(MM_columns[i+1], ",");
        String delim = ((elem = (String)tokens.nextToken()) != null && elem.compareTo("none")!=0)?elem:"";
        String altVal = ((elem = (String)tokens.nextToken()) != null && elem.compareTo("none")!=0)?elem:"";
        String emptyVal = ((elem = (String)tokens.nextToken()) != null && elem.compareTo("none")!=0)?elem:"";
        if (formVal.length() == 0) {
            formVal = emptyVal;
        } else {
            if (altVal.length() != 0) {
                formVal = altVal;
            } else if (delim.compareTo("") == 0) { // escape quotes
                StringBuffer escQuotes = new StringBuffer(formVal);
                for (int j=0; j < escQuotes.length(); j++)
```

```
        if (escQuotes.charAt(j) == "\"") escQuotes.insert(j++, "\"");
        formVal = "" + escQuotes + "";
    } else {
        formVal = delim + formVal + delim;
    }
}
MM_tableValues.append((i!=0)?", ":"").append(MM_columns[i]);
MM_dbValues.append((i!=0)?", ":"").append(formVal);
}
MM_editQuery = new StringBuffer("insert into " + MM_editTable);
MM_editQuery.append(" (").append(MM_tableValues.toString()).append(") values (");
MM_editQuery.append(MM_dbValues.toString()).append(")");

if (!MM_abortEdit) {
    // finish the sql and execute it
    Driver MM_driver = (Driver)Class.forName(MM_editDriver).newInstance();
    Connection MM_connection =
DriverManager.getConnection(MM_editConnection,MM_editUserName,MM_editPassword);
    PreparedStatement MM_editStatement = MM_connection.prepareStatement(MM_editQuery.toString());
    MM_editStatement.executeUpdate();
    MM_connection.close();

    // redirect with URL parameters
    if (MM_editRedirectUrl.length() != 0) {
        // response.sendRedirect(response.encodeRedirectURL(MM_editRedirectUrl));
        return;
    }
}
%>
```

Para el caso de la actualización se utilizó el siguiente código que utiliza la sentencia "UPDATE" de SQL:

```
<%
Driver DriverPrepared1 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();
Connection ConnPrepared1 = DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);
PreparedStatement Prepared1 = ConnPrepared1.prepareStatement("UPDATE DAP_SPG.HD_CAT_TECNICOS SET
HD_APE_PAT_TECNICO="+ Prepared1__apepat + ", HD_APE_MAT_TECNICO="+ Prepared1__apemat + ",
HD_NOMBRE_TECNICO="+ Prepared1__nombres + ", HD_FUNCION="+ Prepared1__funcion + ", HD_RFC_TECNICO="+
Prepared1__rfctecnico + " WHERE HD_CLAVE_TECNICO =" + Prepared1__clave + " ");
Prepared1.executeUpdate();
response.sendRedirect("tecEdita.jsp");
%>
```

Para la eliminación de registros se utilizó el siguiente código, donde utilizamos la instrucción "DELETE" de SQL:

```
<%
int borra=0;
borra=Integer.parseInt( request.getParameter("envio"));
Driver DriverPrepared1 = (Driver)Class.forName(MM_sismed_DRIVER).newInstance();
Connection ConnPrepared1 =
DriverManager.getConnection(MM_sismed_STRING,MM_sismed_USERNAME,MM_sismed_PASSWORD);
PreparedStatement Prepared1 = ConnPrepared1.prepareStatement("DELETE FROM
DAP_SPG.HD_CAT_TECNICOS WHERE HD_CLAVE_TECNICO =" +borra+""");
Prepared1.executeUpdate();
%>
```

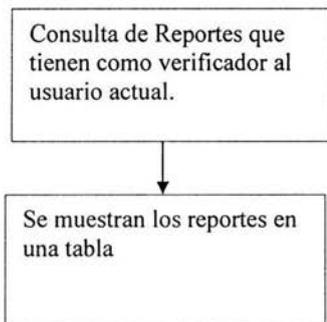
Sólo se expone este código una vez para los dos módulos, ya que el funcionamiento es similar en cuanto a instrucciones SQL a ejecutarse por el objeto Statement. Lo que varía son los nombres de las tablas y los campos que tienen que ver con la inserción, actualización y borrado

de registros, pues estos dependen del catálogo que se esté utilizando en ese momento (técnicos ó equipos).

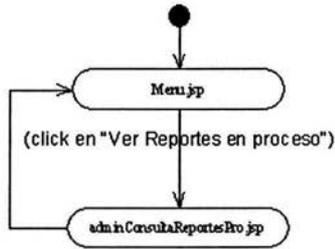
Nos quedan 3 opciones por definir, la primera de ellas es "Reportes en proceso que asigné al personal". Esta opción es para administradores y verificadores. En este punto, ellos pueden ver en que estado están los reportes que ellos asignaron a algún técnico para su atención.

Se hace una consulta a la base de datos donde se le pide que de como resultado todos los reportes y su estado, que tengan asignado al verificador que en ese momento hace la petición. Esto lo podemos saber por la variable de sesión "usuario" creada después de la autenticación.

Los diagramas para esta opción son los siguientes:



CONSULTA DE REPORTES EN PROCESO.



Como lo podemos notar, este módulo es utilizado sólo como monitoreo de reportes, pues no se lleva al cabo alguna otra operación más allá de la primera consulta.

El código utilizado para esta página es el siguiente:

```
<%  
boolean Recordset1_hasData=true;  
Driver DriverRecordset1 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();  
Connection ConnRecordset1 =  
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);  
PreparedStatement StatementRecordset1 = ConnRecordset1.prepareStatement("SELECT DISTINCT * FROM  
DAP_SPG.HD_REPORTES, HD_CAT_USUARIOS,HD_CAT_TECNICOS WHERE HD_VERIFICADO_POR="" +  
Recordset1__usuario + "" AND HD_SERIE_REPORTES=HD_EQUIPO_SERIE AND  
HD_REPORTES.HD_CLAVE_TECNICO=HD_CAT_TECNICOS.HD_RFC_TECNICO ORDER BY  
HD_ESTADO_REPORTES,HD_CLAVE_REPORTES, HD_FECHA_REPORTES, HD_HORA_REPORTES");  
ResultSet Recordset1 = StatementRecordset1.executeQuery();  
boolean Recordset1_isEmpty = !Recordset1.next();  
Recordset1_hasData = !Recordset1_isEmpty;  
Object Recordset1_data;  
int Recordset1_numRows = 0;  
%>  
<table width="100%" border="0" align="center">  
<tr>
```

```
<td width="7%" height="43"><font size="1" face="Arial, Helvetica, sans-serif">Fecha  
y Hora de Reporte</font></td>  
<td width="9%"><font size="1" face="Arial, Helvetica, sans-serif">Fecha  
y Hora de Asignaci&oacute;n de Responsable</font></td>  
<td width="7%"><font size="1" face="Arial, Helvetica, sans-serif">Clave  
Reporte</font></td>  
<td width="22%"><font size="1" face="Arial, Helvetica, sans-serif">Nombre  
del Usuario</font></td>  
<td width="7%"><font size="1" face="Arial, Helvetica, sans-serif">Serie  
Equipo</font></td>  
<td width="8%"><font size="1" face="Arial, Helvetica, sans-serif">Verificado  
por.</font></td>  
<td width="12%"><font size="1" face="Arial, Helvetica, sans-serif">Atendido  
por :</font></td>  
<td width="13%"><font size="1" face="Arial, Helvetica, sans-serif">Detalles:</font></td>  
<td width="7%"><font size="1" face="Arial, Helvetica, sans-serif">Estado</font></td>  
<td width="8%"><font size="1" face="Arial, Helvetica, sans-serif">D&iacute;a  
ocupados</font></td>  
</tr>
```

```
<% String color="";  
while ((Recordset1_hasData) ||(Recordset1.next())) {  
int fechaTemp=tiempo.getDays((String)Recordset1.getString("HD_FECHA_REPORTE"),fecha);  
//fecha,  
  
//generamos los colores de los semáforos para indicar cuanto se han tardado.  
if(fechaTemp<=2)  
color="#2D7402";  
else if (fechaTemp>2 && fechaTemp<=4)  
color="#F4DA00";
```

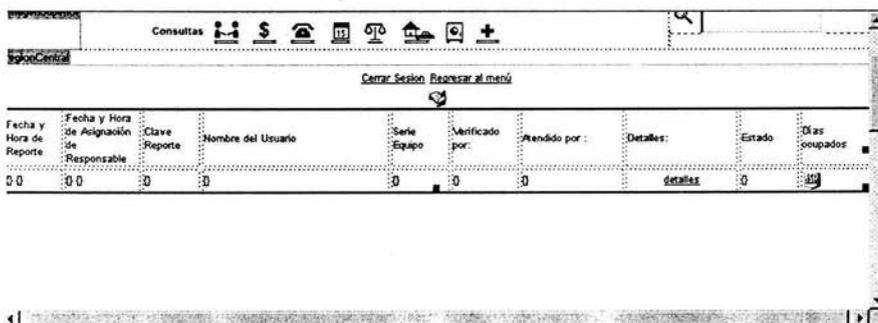
```
else if (fechaTemp>4)
color="#B60000";
else
color="#55FFCC";
%>
<tr>
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_FECHA_REPORTE"))==null || Recordset1.isNull())?"":Recordset1_data)%>
<%=(((Recordset1_data = Recordset1.getObject("HD_HORA_REPORTE"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_FECHA_RESPONSABLE"))==null || Recordset1.isNull())?"":Recordset1_data)%>
<%=(((Recordset1_data = Recordset1.getObject("HD_HORA_RESPONSABLE"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_CLAVE_REPORTE"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_USUARIO_NOMBRECORTO"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_EQUIPO_SERIE"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_VERIFICADO_POR"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_CLAVE_TECNICO"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td><div align="center"><font size="1" face="Arial, Helvetica, sans-serif"><a
href=javascript:abreDetalles(<%=Recordset1.getString("HD_CLAVE_REPORTE")%>);>detalles</a></font></div></td>
</tr>
```

```

<td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_ESTADO_REPORTE"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
<td bgcolor="<%=color%"><font color="#FFFFFF"><%=fechaTemp%"></font></td>
</tr>
<%
Repeat1__index++;
Recordset1_hasData = Recordset1.next();
}
%>
</table>

```

Quedando en vista diseño algo similar a lo que se observa a continuación:



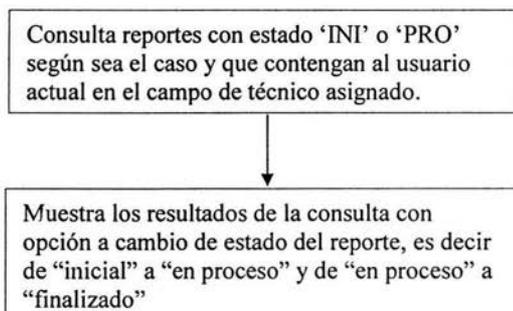
Las dos opciones restantes del menú, son aquellas que aparecen a cualquier usuario, y son: "Mis reportes asignados" y "Mis reportes asignados en proceso".

En "mis reportes asignados" todos los usuarios de este sistema administrativo pueden ver que reportes les han sido asignados y que no han empezado a trabajar sobre ellos. Cada reporte

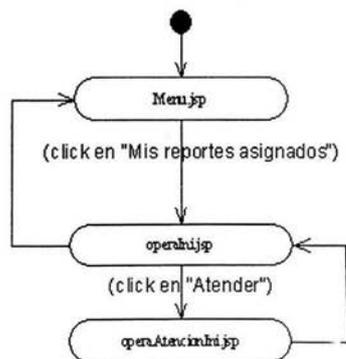
que aparece, les da la oportunidad de cambiar su estado de "inicio" a un estado de "en proceso", dando clic sobre el botón que está a un lado de ellos.

En "mis reportes asignados en proceso", como su nombre lo indica, los usuarios pueden ver que reportes ya han empezado a trabajar y ahí tienen la opción de finalizarlos dando clic en el botón que aparece al lado de cada uno de ellos.

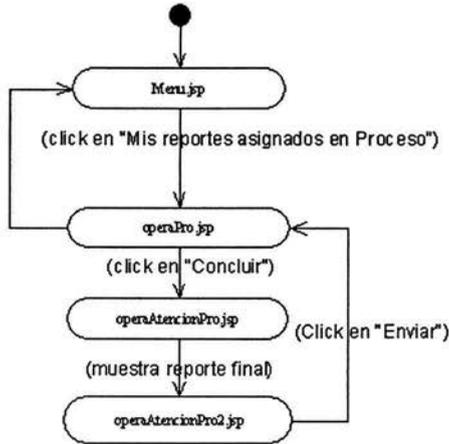
Los diagramas para estas dos opciones, son los siguientes:



#### MIS REPORTES ASIGNADOS



MIS REPORTES ASIGNADOS EN PROCESO



El código para obtener la funcionalidad buscada en estos módulos, según los diagramas, es el siguiente:

```
<%
Driver DriverRecordset1 = (Driver)Class.forName(MM_avii_DRIVER).newInstance();
Connection ConnRecordset1 =
DriverManager.getConnection(MM_avii_STRING,MM_avii_USERNAME,MM_avii_PASSWORD);
PreparedStatement StatementRecordset1 = ConnRecordset1.prepareStatement("SELECT distinct * FROM
DAP_SPG.HD_REPORTES, DAP_SPG.HD_CAT_USUARIOS WHERE HD_ESTADO_REPORTES='INI' AND
HD_EQUIPO_SERIE=HD_SERIE_REPORTES AND HD_VERIFICADO_POR is not Null AND
HD_CLAVE_TECNICO = " + Recordset1__rfctecnico + " ORDER BY HD_CLAVE_REPORTES,
HD_FECHA_REPORTES, HD_HORA_REPORTES");
ResultSet Recordset1 = StatementRecordset1.executeQuery();
boolean Recordset1_isEmpty = !Recordset1.next();
boolean Recordset1_hasData = !Recordset1_isEmpty;
Object Recordset1_data;
int Recordset1_numRows = 0;%>
```

Para mostrar los reportes en una tabla con la opción de cambio de estado, se utilizó el siguiente código:

```
<% while ((Recordset1_hasData)&&(Repeat1__numRows-- != 0)) { %>
    <tr>
        <form action="operaAtencionIni.jsp" method="post">
            <td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_FECHA_REPORTE"))==null || Recordset1.isNull())?"":Recordset1_data)%>
<%=(((Recordset1_data = Recordset1.getObject("HD_HORA_REPORTE"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
            <td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_CLAVE_REPORTE"))==null || Recordset1.isNull())?"":Recordset1_data)%></font></td>
            <td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("HD_USUARIO_NOMBRECORTO"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
            <td><font size="1" face="Arial, Helvetica, sans-serif"><%=(((Recordset1_data =
Recordset1.getObject("UBICACION_USUARIOS"))==null ||
Recordset1.isNull())?"":Recordset1_data)%></font></td>
            <td><font size="1" face="Arial, Helvetica, sans-serif"><a
href="javascript:abreDetalles(<%=Recordset1.getString("HD_CLAVE_REPORTE")%>);">detalles</a></font></td>
            <td><font size="1" face="Arial, Helvetica, sans-serif">

<input name="imageField2" type="image" src=".../Imagenes/Navegacion/men_navegacion.jpg" width="36"
height="20" border="0">

                Iniciar

                <input name="reporte" type="hidden" id="reporte" value="<%=(((Recordset1_data =
Recordset1.getObject("HD_CLAVE_REPORTE"))==null || Recordset1.isNull())?"":Recordset1_data)%>">

                <input name="estado" type="hidden" id="estado" value="PRO">

            </font></td>
        </form>
    </tr>
```

```
<%  
Repeat1__index++;  
Recordset1_hasData = Recordset1.next();  
}  
%>
```

Un punto pendiente es aquel que tiene que ver con mostrar el número de días que lleva un reporte. Para que en la interfaz donde aparecen todos los reportes, se indicara cuantos días hábiles a tardado un reporte desde su fecha de inicio hasta la fecha actual, se desarrolló un componente de Java para que nos diera la cuenta de días. El código utilizado, para tal efecto, es el siguiente:

```
package fechaLaboral;  
  
/**  
 * Title:  
 * Description:Cuenta de Dias Laborales  
 * Copyright: Copyright (c) 2003  
 * Company:  
 * @author Daniel Ruiz  
 * @version 1.0  
 */  
  
import java.util.*;  
import java.math.*;  
import java.util.Date;  
public class DaysCount {  
  
    public DaysCount() {  
    }  
  
    public int getDays(String fechalnicial, String fechaFinal){  
    StringTokenizer st=new StringTokenizer(fechalnicial,"/");
```

```
int dia1=Integer.parseInt(st.nextToken());
int mes1=Integer.parseInt(st.nextToken());
int an1=Integer.parseInt(st.nextToken()+2000;
StringTokenizer st2=new StringTokenizer(fechaFinal,"/");
int dia2=Integer.parseInt(st2.nextToken());
int mes2=Integer.parseInt(st2.nextToken());
int an2=Integer.parseInt(st2.nextToken()+2000; //para hacerla de cuatro digs.
return this.getDays(an1,mes1,dia1,an2,mes2,dia2);
```

}/Este método fue creado por Daniel Ruiz

```
public int getDays(int an1,int mes1,int dia1,int an2,int mes2,int dia2){

int an1900=an1-1900;//año 1 menos 1900
int an21900=an2-1900;//año 2 menos 1900
int mes11=mes1-1; //mes 1 menos 1
int mes21=mes2-1; //mes 2 menos 1
//convertimos los enteros en objetos Date
Date fecha1=new Date(an1900,mes11,dia1);
Date fecha2=new Date(an21900,mes21,dia2);
Date fechaaux=fecha1;
//System.out.println("hola"+fecha1.getDay());
// (0 = Sunday, 1 = Monday, 2 = Tuesday, 3 = Wednesday, 4 = Thursday,
// 5 = Friday, 6 = Saturday)
int count=0;
int dia1aux=dia1;
int mesauxiliar=fechaaux.getMonth();
int anauxiliar=fechaaux.getYear();
int diaanterior=fechaaux.getDay()-1;//ojo
//System.out.println(fechaaux.toString());
//if(fechaaux.getDay()!=0 && fechaaux.getDay()!=6){
```

```
//count=-1;
//}
while(fechaaux.compareTo(fecha2)!=1){ //hasta que las fechas no sean iguales
    if(diaanterior!=fechaaux.getDay()){
        if(fechaaux.getDay()!=0 && fechaaux.getDay()!=6){
            count=count+1;
//      System.out.println("sumo, count vale:" + count);

        }//end if
//else{
//      System.out.println("Es domingo o Sabado no sumo" + count);
//  }
}
diaanterior=fechaaux.getDay();
fechaaux.setDate(dia1aux++);//aumento un día a la fecha
// System.out.println("Despues de aumentar la fecha:"+fechaaux.toString());
if(fechaaux.getMonth()!=mesauxiliar||fechaaux.getYear()!=anauxiliar){//si cambia de mes, cambiamos el mes
//a la fecha auxiliar y que siga contando
// fechaaux.setMonth(fechaaux.getMonth());
// System.out.println("cambio de mes o cambio de año");
dia1aux=1;
mesauxiliar=fechaaux.getMonth();
anauxiliar=fechaaux.getYear();
fechaaux.setYear(anauxiliar);
fechaaux.setMonth(mesauxiliar);
fechaaux.setDate(dia1aux);
// System.out.println("Comparando da:"+fechaaux.compareTo(fecha2));
//fechaaux=fechaaux2;
//System.out.println("Esta es la nueva fecha"+fechaaux.toString());

}
```

```
//System.out.println(fechaaux.getMonth()+"vs"+mesauxiliar);  
//System.out.println(fechaaux.toString()+"el día"+fechaaux.getDay());  
}end while
```

```
//System.out.println("Cuenta Final" + count);  
//System.out.println("Cuenta Final2" + fechaaux.toString());  
return count;  
}  
}
```

Para que esta clase pudiera ser llamada desde cualquier página JSP del sistema, se copió el paquete completo "package" dentro del directorio de librerías del servidor de aplicaciones "Tomcat". Con esto, cualquier página que importase este paquete, podría crear una instancia de la clase DaysCount.

El modo de utilizar este componente es:

- a) Genero un objeto de la clase DaysCount: DaysCount dc=new DaysCount();
- b) Guardo el valor que me regresa el método de la clase DaysCout, getDays(), en una variable de tipo int: int diaslaborales=dc.getDays(2004,03,15,2004,04,01).
- c) Imprimir el número de días laborales: System.out.println("Numero de días laborales= " +diaslaborales);

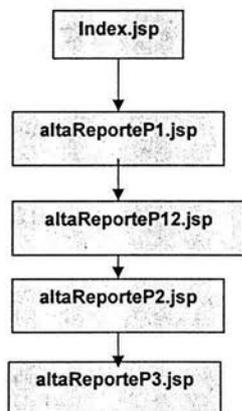
Así con la ayuda de este componente podemos determinar el número de días que lleva un reporte desde su fecha de asignación a la actual, dándole como parámetros de entrada, la fecha inicial y la fecha actual para que nos de el resultado esperado.

Con esto se concluye con las opciones del menú administrativo y con el desarrollo del código del sistema.

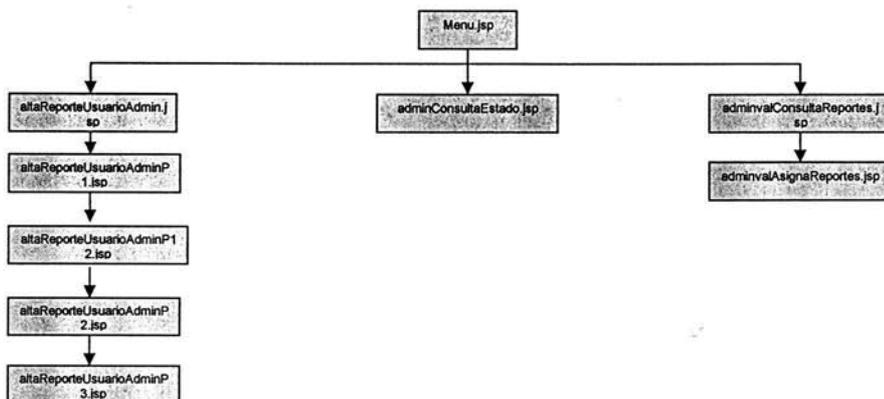
En este momento ya se tienen las interfases y el código que va a funcionar como motor del sistema, al unirlos obtenemos la funcionalidad buscada y podemos decir, que el sistema ha quedado hasta este momento en un 90% concluido; el 10% restante corresponde al visto bueno del usuario final. Cabe señalar que la unión del código Java con las interfases html, consistió en pegar el código JSP dentro de la interfaz, en las partes designadas para las partes dinámicas.

Por último, se muestra el mapa de navegación del sistema. Este mapa intenta mostrar la distribución final de cada una de las páginas JSP creadas para este sistema y su relación con las demás, en el entorno de programación.

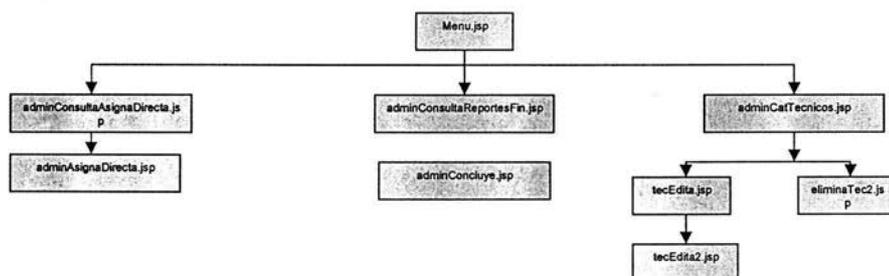
#### ALTA DE REPORTE



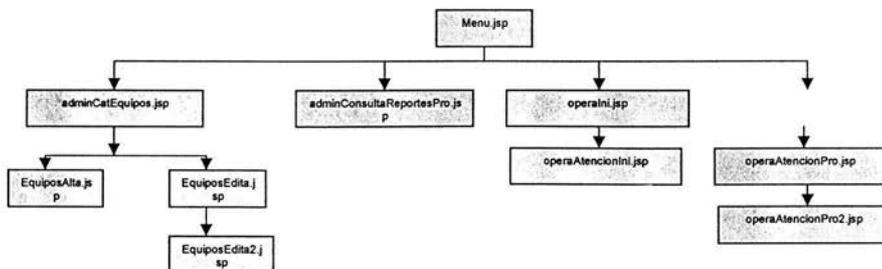
### PARTE ADMINISTRATIVA 1



### PARTE ADMINISTRATIVA 2



### PARTE ADMINISTRATIVA 3



## **CAPÍTULO 6**

### **PRUEBAS CON EL USUARIO FINAL**

Antes de presentar el sistema con el usuario final, se instaló este en un servidor de desarrollo y se le hicieron las siguientes pruebas:

- a) Acceso a la aplicación a través de la Intranet.
- b) Pruebas de seguridad
- c) Manejo de la memoria.

En las pruebas de acceso no se tuvo problema alguno, dado que el servidor de aplicación en producción se encuentra ejecutándose correctamente dentro de la red local. Al ser este servidor, el mismo utilizado por la intranet, siempre está bajo monitoreo de rendimiento y en mantenimiento preventivo. El tiempo de carga que se obtuvo de cada página JSP del sistema, fue menos de 2 segundos, lo que se consideró un tiempo aceptable.

Para las pruebas de seguridad, se dieron de alta a varios usuarios con distinta función en el sistema. Todos ellos quisieron acceder a alguna página no permitida para ellos, a lo que el sistema les mandaba un mensaje de error. Con ello comprobamos que la autorización de usuarios a través de variables de sesión fue buena alternativa, sólo restaba comprobar que no se consumiera mucha memoria con la utilización de éstas.

En el manejo de memoria, al principio se detectó que se consumía una parte bastante considerable, porque no se estaba liberando la parte de memoria que era utilizada por los usuarios al iniciar la sesión. Esto pudo corregirse configurando el servidor para que sólo mantuviera las sesiones inactivas por 2 minutos. Si en 2 minutos un usuario registrado no generaba ninguna operación sobre el sistema, este sería expulsado automáticamente y con ello quedaría liberado el espacio de memoria ocupado por esa sesión.

Una vez concluidas las pruebas, se presentó el sistema, al usuario final para realizar las pruebas de uso. En estas pruebas, se verificó que el usuario no tuviera problemas para acceder a la aplicación, que la interfaz de usuario lo guiara correctamente en cualquier parte del sistema y que el sistema cumpliera con los requerimientos planteados inicialmente.

Una vez que el usuario verificó que el desarrollo cumple con todos sus requerimientos, nos firmó el documento correspondiente donde se declara al sistema como un producto conforme.

## CONCLUSIONES

### **Sobre la tecnología empleada:**

Recapitulando, la tecnología empleada en este sistema fue, como servidor de aplicaciones Tomcat, como plataforma de desarrollo Java, como sistema operativo Linux Suse 7.3 y como sistema de base de datos, Oracle 9i.

Al utilizar Oracle 9i para el manejo de la información del sistema, se tuvo un gran soporte en el acceso y modificación de la misma. Es preciso señalar que, el mantenimiento y optimización de esta base de datos, se hace continuamente en la SHCP, lo que hace posible reducir el tiempo de respuesta para la atención de una consulta, hecha por el servidor de aplicaciones.

Con el desarrollo de este sistema puedo afirmar, que la combinación de las plataformas de software utilizadas, en este caso en particular fue:

- a) Confiable: No se registraron fallas en los sistemas de software utilizados, ni negaciones en el servicio (Denial of Service).
  
- b) Robusto: Soporta gran cantidad de usuarios simultáneos sin sacrificar tiempo en la generación de las páginas dinámicas.
  
- c) De bajo costo: El único software que tiene un costo por licenciamiento, en este sistema, es la base de datos Oracle 9i, pero la SHCP ya contaba con este recurso. Tomcat, Java y Linux son recursos gratuitos, que no requieren de licenciamiento. Tomcat puede ser adquirido desde el sitio de Apache (<http://www.apache.org>), y Java desde el sitio oficial de esta tecnología (<http://java.sun.com>).

Hay que tomar en cuenta que todo fue instalado en servidores debidamente configurados con capacidad de memoria de más de 1 GB, por ello el desempeño del sistema fue excelente. Cabe señalar que los resultados aquí mostrados pueden variar dependiendo de la configuración del ambiente donde vaya a residir este sistema.

**Sobre el desarrollo:**

Al momento de realizar este sistema, es preciso señalar, que el tiempo de producción autorizado fue bastante corto (dos semanas), por lo que se tuvo que implementar un procedimiento rápido para en el análisis y diseño de un sistema.

Debido a las características obtenidas mediante el análisis de requerimientos, fue posible de alguna manera, llevar al cabo el análisis, diseño y desarrollo casi a la par. Por ello es que no se ven diagramas tan elaborados o con alguna metodología específica, a excepción de los diagramas de actividades que provienen del Unified Modeling Language (UML). Esto se debió a que el desarrollo de todo el sistema, estuvo sólo a mi cargo, por lo que se buscó una forma de documentación que permitiera saber en que punto se encontraba el desarrollo y así poder llegar a la culminación del mismo, o en su caso, hacerle modificaciones; ambas actividades en el menor tiempo posible.

Lo recomendable, a mi punto de vista, es dar el tiempo necesario para el análisis y a la documentación del mismo, ya que esto reduce de manera considerable el tiempo de desarrollo y, en el caso de que alguna modificación o actualización del sistema deba hacerse, la persona encargada de esto no tendrá problemas, pues podrá conocer la lógica del sistema.

En lo que respecta al desarrollo con JSP, es necesario tener ciertos conocimientos de programación orientada a objetos para poder explotar al máximo las capacidades de éste lenguaje. La parte más interesante es que puedes tener el poder que brinda Java, para poder crear páginas web dinámicas.

Como se pudo notar, el paquete del lenguaje Java más utilizado en este sistema fue el "java.sql.\*", que es donde se encuentra todo lo relacionado con consultas a sistemas de bases de datos mediante JDBC. También se observó que, el hacer una consulta mediante Java no es de gran dificultad, de hecho al hacerlo una vez, las demás ocasiones son muy parecidas, generalmente lo que cambia es el texto de la consulta SQL y algunos parámetros adicionales al momento de generar el objeto ResultSet.

#### **Sobre el sistema:**

Al poner en marcha el sistema, dentro de la Intranet de la SHCP, tuvimos un claro ejemplo de cómo un sistema puede hacer más eficiente algún proceso, siempre y cuando el análisis del mismo sea lo mas completo posible. Existen varios indicadores para evaluar la efectividad de un sistema, en este caso, nuestro principal indicador, fue el tiempo, pues se redujeron considerablemente los tiempos de respuesta de los técnicos a los reportes, así como los tiempos de resolución de los mismos. Todo esto hizo que el área de soporte técnico fuera percibida de diferente manera por todos sus usuarios.

En cuanto al desempeño ("performance") del sistema, se hicieron pruebas con 100 usuarios conectados simultáneamente en el sistema, simulando con ello una situación crítica. Todos ellos se encontraban realizando alguna operación dentro del sistema. Se observó que la aplicación no tuvo ninguna baja en el tiempo promedio de respuesta y que en ningún momento se negó el servicio. Cabe señalar que, al no utilizar demasiadas variables de

sesión y cerrando cada objeto Resultset, Statement y Connection cuando ya no eran útiles, se previno de alguna manera esta situación.

Con el esquema generado de base de datos, es posible obtener una gran gama de reportes con base en las necesidades del negocio, por ejemplo, se puede saber cuantos reportes en promedio atiende un técnico, cuántos días promedio tarda en resolverse un problema, cuáles son los problemas que comúnmente se presentan, etc. Cabe señalar que esto no se llevaba a cabo antes de la creación del sistema, con lo cual se abre la posibilidad de explotar estos datos almacenados para la toma de decisiones gerenciales basadas en hechos.

**En general:**

Como conclusión general puedo decir que fue desarrollado un sistema que cumplió con los requerimientos del cliente, en el tiempo planeado y con los recursos asignados. Aunque no se utilizó una metodología de desarrollo específica (además de que en la SHCP no se tiene) o un patrón de Java en particular, si se llegó al resultado deseado con un desempeño óptimo, con lo cual el usuario quedo satisfecho.

Todo lo expuesto en este trabajo, representa una de las primeras experiencias en el campo profesional que tuve, que me dejó muchos conocimientos en lo que respecta al desarrollo de aplicaciones Web, y que ha servido como base para el desarrollo de sistemas cada vez más complejos y con mejor calidad.

## BIBLIOGRAFIA

- Brown, Burdick, Falkner, Galbraith. "Professional JSP". Wrox Press 2a Edición. 2001. pp.7-29, 35-43,923.
- Hall,Marty. "Core Servlets and Java Server Pages". Prentice Hall. 1a. Edición. 2000. pp. 228-244,286-304
- Roger S. Pressman; Ingeniería del Software;4ª Edición; Mc Graw Hill. pp 10-50.

### HIPERVÍNCULOS

- Página de COMUNET, Education Solutions <http://www.comunet-education.com>
- Página de Java SUN. <http://java.sun.com>
- Apache org <http://apache.org>
- Macromedia Dreamweaver <http://www.macromedia.com>
- Glosario de términos en Internet  
<http://www.lanyon.com/support/Glossary/Glossarya-d.htm>

**ANEXO A.**  
**MANUAL TÉCNICO Y MANUAL DE USUARIO.**

---

## ÍNDICE

	<u>Pág.</u>
<b>Manual Técnico</b>	
I. Especificaciones técnicas	IV
II. Instalación	V
<b>Manual Operativo</b>	
I. Levantamiento de reportes	VIII
II. Gestión de los reportes	XV
II.I Menú de la parte administrativa del sistema de soporte técnico	XVII
II.I.1 Alta de reporte de usuario	XVIII
II.I.2 Ver todos los reportes	XXII
II.I.3 Reportes generados no atendidos para la asignación de verificadores	XXIV
II.I.4 Asignación directa de personal para reportes	XXV
II.I.5 Verificación de finalización de reporte y cerrado del mismo	XXVI
II.I.6 Mantenimiento de la información del personal	XXVII

II.1.7	Mantenimiento de la información de los equipos	XXX
II.1.8	Reportes que asigné al personal	XXXIII
II.1.9	Mis reportes asignados (Inicio de atención de un reporte)	XXXV
II.1.10	Mis reportes asignados en proceso (Finalización de un reporte)	XXXVII



**Sistema de Soporte Técnico de la INTRANET  
de la Dirección General de Recursos Humanos de la  
Secretaría de Hacienda y Crédito Público**

# MANUAL TÉCNICO

## I. Especificaciones técnicas.

El sistema de soporte técnico de la Dirección General de Recursos Humanos de la Secretaría de Hacienda y Crédito Público, fue desarrollado como una extensión para el sistema Intranet de la misma Dirección, por lo que es necesario que se encuentre activo y en perfectas condiciones, ya que es el motor principal de nuestro sistema.

A manera informativa señalaremos que la plataforma donde se está publicando la Intranet en este momento es la siguiente:

- Lenguaje de Programación: JSP
- Procesador: Pentium 4 a 700 Mhz.
- Disco Duro: 40 GB
- Memoria: 1GB
- Sistema Operativo: Linux Suse 7.3
- Servidor de Aplicaciones: Apache Tomcat 4.0

## II. Instalación.

El disco de instalación contiene:

- a) Una carpeta de nombre **dor**
- b) Una carpeta de nombre **help desk**
- c) Un archivo de nombre **dbscript.sql**





La carpeta de nombre **dor** es el paquete que contiene la clase DORUtil.class y es la encargada de contar el número de días. Esta carpeta deberá agregarse al directorio de librerías de clases del servidor de aplicaciones Tomcat. La dirección de esta carpeta puede ser similar a:

*/raíz del servidor de aplicaciones/Server/lib/classes*

Consulte el manual de usuario del servidor de aplicaciones para su rápida localización.

La carpeta de nombre **helpdesk** debe agregarse tal cual a la carpeta designada en la aplicación de la Intranet para las aplicaciones seguras. La dirección debe ser similar a:

*/raíz del servidor de aplicaciones/webapps/ROOT/Intranet/links/seguras/*

Para mayor información consulte con el administrador de la Intranet de la Dirección General de Recursos Humanos.

El archivo **dbscript.sql** es aquél que contiene el script de bases de datos que creará las tablas en la base de datos ORACLE 8i donde se almacenará la información generada por el sistema de soporte técnico de la Dirección General de Recursos Humanos. Este archivo deberá ser ejecutado por el DBA (administrador de la base de datos) dentro de una instancia de nombre dap\_spg, que es la misma instancia donde se encuentran las tablas de bases de datos de la Intranet.

Realizado lo anterior, la aplicación estará lista para ser utilizada en la Intranet.



**Sistema de Soporte Técnico de la INTRANET  
de la Dirección General de Recursos Humanos de la  
Secretaría de Hacienda y Crédito Público**

# MANUAL DE OPERACIÓN

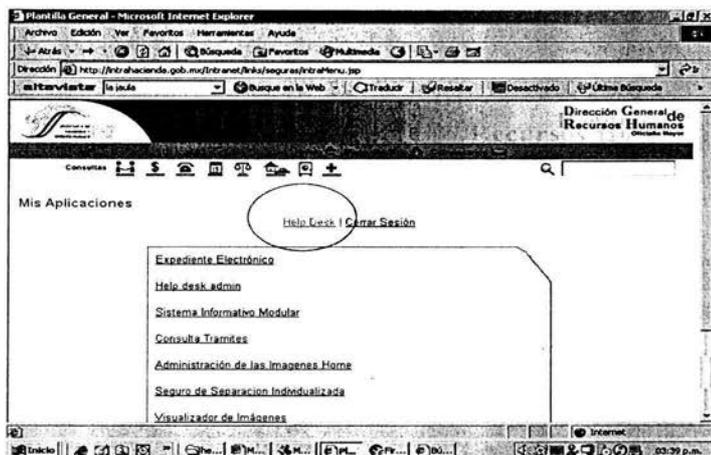
## Levantamiento de Reportes

Para poder levantar un reporte de soporte técnico hay que seguir los siguientes pasos:

- Autenticación en la Intranet:** el usuario deberá proporcionar el nombre de usuario y contraseña para su validación.



- b) Si la autenticación es correcta, aparece una imagen como la siguiente:



En la parte superior de esta página aparece un hipervínculo con la leyenda "Help Desk".

Dé clic sobre el hipervínculo.

- c) Al aparecer la siguiente pantalla, deberá dar clic en el botón CONTINUAR.





- e) En la siguiente pantalla, seleccione la o las fallas dentro del formulario, dando clic sobre el cuadro que aparece del lado izquierdo de cada descripción de falla. En caso de que no se muestre la descripción de las fallas del equipo que reporta, escriba en el campo de OTROS una breve descripción de la falla.

Plantilla General - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Inicio

Dirección http://intra.hacienda.gob.mx/intranet/links/seguras/feb06/sk/akareportaP2.jsp

Inicio

Comentarios

Cerrar Sesión

Verifique que sus datos sean correctos:

Rfc: RURD76123101  
Nombre Completo: Daniel Ruiz Rodriguez  
Marca del Equipo: Lanix  
Serie Equipo: 110003652  
Numero Inventario: 3724245  
Ubicación: EDIF. D P.B.

Seleccione la(s) fallas o servicios aplicables al equipo:

Capacitación y asesoría:  
Requiero capacitación y/o asesoría

Correo Electrónico:  
 Alta de Correo Electrónico Interno/Externo

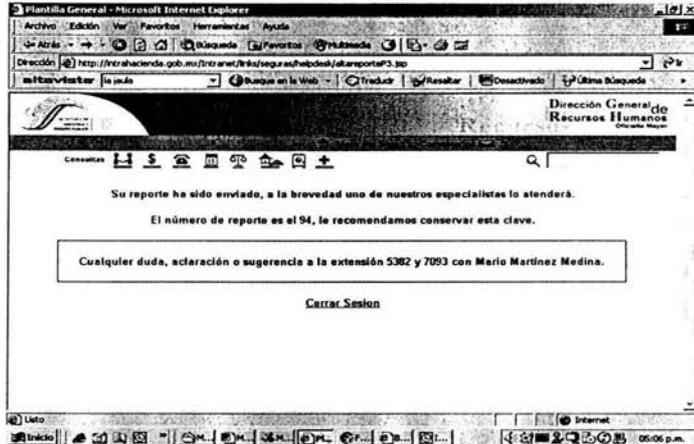
Impresoras:  
 Cambio de Tóner/Cartucho de Impresión  Configuración de Impresora (No puedo imprimir)  
 Mantenimiento de Impresora

Inicio

03:49 p.m.

Una vez señaladas las fallas, dé clic en ENVIAR para concluir con el levantamiento del reporte.

- f) Concluido el levantamiento del reporte, aparecerá un mensaje de confirmación como lo representa la siguiente imagen:



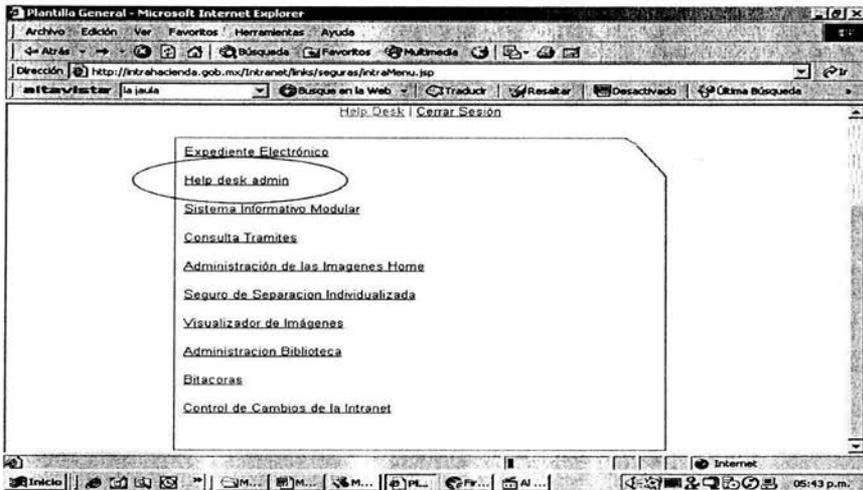
Para salir del sistema, basta con dar clic en el hipervínculo que dice CERRAR SESIÓN, o bien, cerrando la ventana del explorador.

## II. Gestión de los reportes. Área Administrativa (parte privada).

Para poder acceder a la parte administrativa del sistema de soporte técnico de la Intranet de la Dirección General de Recursos Humanos, lo primero que se debe hacer es autenticarse.



Al igual que para la parte de levantamiento de reportes, si la autenticación fue exitosa, se mostrará una pantalla que contiene un menú el cual muestra las aplicaciones que se podrán utilizar.

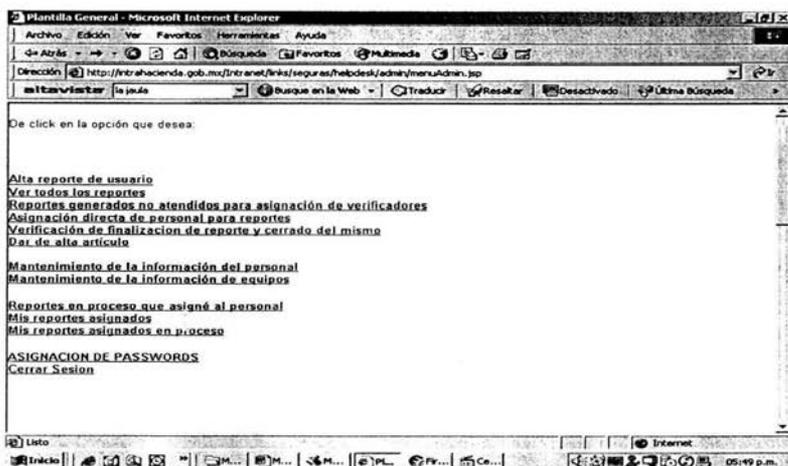


Verá que aparece un hipervínculo que dice HELP DESK ADMIN (si se tiene permitido). Al dar clic sobre éste accederá a la parte administrativa del sistema de soporte técnico.

## II.1 Menú de la parte administrativa del sistema de soporte técnico

El menú de la parte administrativa depende de los privilegios que se tengan en la aplicación. Esto quiere decir que pueden aparecer todos los incisos o sólo algunos dependiendo de la función que se le asigne: administrador, verificador u operador; pero la funcionalidad es la misma en todos los privilegios.

La pantalla del menú es la siguiente:

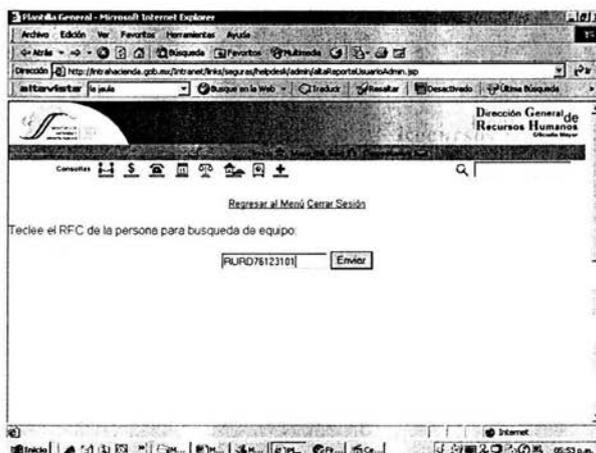


Para poder acceder a cualquiera de estas opciones del menú, basta con dar clic en el hipervínculo correspondiente.

### II.1.1 Alta de reporte de usuario.

Esta opción se utiliza cuando haya que levantar un reporte que se ha hecho por otro medio que no ha sido la Intranet (telefónico, e-mail, etc). Los pasos para levantar un reporte por un usuario, son los siguientes:

- a) Escribir el RFC (filiación) del usuario que desea levantar un reporte.



- b) En la siguiente pantalla, elija el equipo sobre el cual se quiere hacer el reporte y dé clic en el botón con la leyenda ELEGIR que aparece al lado derecho de la descripción del equipo elegido.



- c) En la siguiente pantalla, seleccione la o las fallas dentro del formulario, dando clic sobre el cuadro que aparece a la izquierda de cada descripción de falla. En caso de que ningún caso describa las fallas del equipo que reporta, escriba en el campo de OTROS una breve descripción de la misma.

Carra Sesión

Verifique que sus datos sean correctos:

RUC: RURD76123101  
 Nombre Completo: Daniel Ruiz Rodriguez  
 Marca del Equipo: Lave  
 Serie Equipo: 110003552  
 Numero Inventario: 3724245  
 Ubicación: EDF D P B.

Seleccione la(s) fallas o servicios aplicables al equipo:

Capacitación y asesoría:  
 Requero capacitación y/o asesoría

Correo Electrónico:  
 Alta de Correo Electrónico Interno/Externo

Impresoras:  
 Cambio de Tóner/Cartucho de Impresión  Configuración de Impresora (No puedo imprimir)  
 Mantenimiento de Impresora

Instalación de software | Restauración de software | Configuración de software

Que software es:

Hardware y Equipo de Computo:  
 Mantenimiento Preventivo/Correctivo

Fallas en:  Teclado  Mouse  Floppy Drive 3 1/2  
 CD-ROM  Bocinas  Audio  Monitor  No enciende

OTROS:  
 Indique aquí el problema:

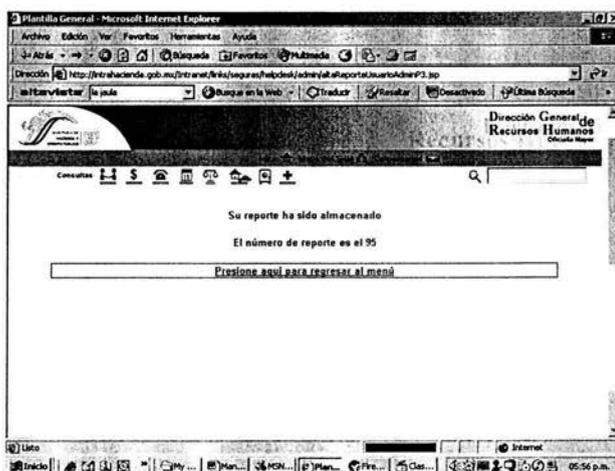
Si este reporte fue consecuencia de un reporte anterior, por favor teclee aquí en número de reporte:

Datos Adicionales(extensión telefónica, nueva ubicación, datos no correctos, etc)

CANCELAR | ENVIAR

Una vez señaladas las fallas, dé clic en ENVIAR para concluir con el levantamiento del reporte.

- d) Una vez terminado el levantamiento del reporte, aparecerá un mensaje de confirmación como lo representa la siguiente imagen:



Para salir del sistema, dé clic en el hipervínculo que lo regresa al menú, o bien cerrando la ventana del explorador.

## II.1.2 Ver todos los reportes

Esta opción del menú permite ver todos los reportes que se han levantado, el estado en que se encuentran, el número de días que ha tardado y los detalles del mismo.

Al dar clic sobre el hipervínculo con la leyenda VER TODOS LOS REPORTES, automáticamente aparecerá la pantalla siguiente:

Fecha y Hora de Reporte	Clave Reporte	Nombre del Usuario	Serie Equipo	Verificado por	Atendido por	Detalles	Estado	Días ocupados
01/04/03 17:58	2403304	Rosio Moreno Ramirez	80021431	RURD76123101	RURD76123101	detalles	FIN	279
01/04/03 17:59	1204004	Rosio Moreno Ramirez	80021431	RURD76123101	RURD76123101	detalles	FIN	279
24/03/03 16:56	2403303	Mario Antonio Martinez Medina	110003515	MAMM71012601	MAMM71012601	detalles	FIN	285
24/03/03 21:25	2403303	Mario Antonio Martinez Medina	110003515	MAMM71012601	MAMM71012601	detalles	FIN	285
01/04/03 18:04	0104003	MARTINEZ MEDINA MARIO ANTONIO	77351	VECL09070701		detalles	INI	279
01/04/03 18:23	0104003	Bianca Monica Montiel Ortiz	40000540			detalles	INI	279
01/04/03 18:25	0104003	Bianca Monica Montiel Ortiz	40000540			detalles	INI	279
01/04/03 18:26	0104003	Carlos Alberto Jimenez Granados	110003518			detalles	INI	279
01/04/03 18:29	0104003	Bianca Monica Montiel Ortiz	40000540			detalles	INI	279

Como podrá notar, se trata de un reporte que contiene la información distribuida en las siguientes columnas:

Clave Reporte	Nombre del Usuario	Serie Equipo por:	Verificado por :	Atendido por :	Detalles: Estado	Días ocupados
------------------	--------------------------	----------------------	---------------------	-------------------	------------------	------------------

Para ver el detalle de algún reporte en particular, dé clic en el hipervínculo **DETALLES** que corresponda al reporte que le interesa ver en ese momento. Aparecerá una pantalla del siguiente tipo:

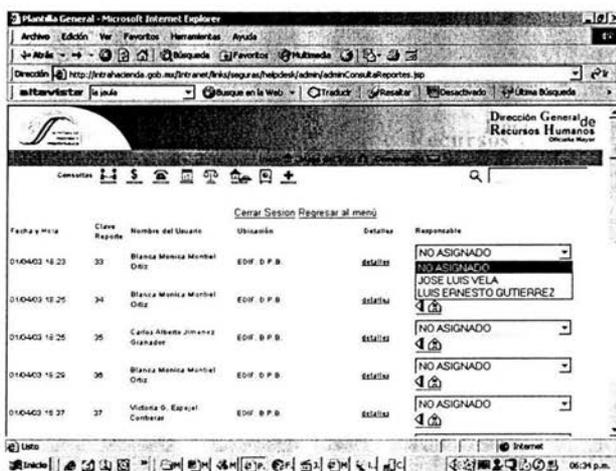


En ella podemos ver cuál es el problema que reportó el usuario. Para salir de esta pantalla basta con cerrarla.

### II.1.3 Reportes generados no atendidos para la asignación de verificadores

Esta opción del menú permite asignar un verificador para que sea el responsable de asignar este reporte a un técnico y monitorearlo. Recordemos que los verificadores fungen como intermediarios entre el administrador y los operadores (técnicos).

Al dar clic en esta opción del menú aparecerá la pantalla siguiente:



Para poder asignar un verificador a un reporte, debemos de dar clic en el menú que se despliega al lado derecho de cada reporte que se muestra en pantalla. Aparecerán los verificadores registrados, de los cuales deberá elegir uno colocando el cursor sobre su nombre, y dé clic en la imagen que aparece debajo de este menú. La pantalla se actualizará y verá que en la casilla de responsable aparece ahora el que eligió.

En la parte superior puede visualizar las opciones para regresar al menú principal y para salir de la sesión.

## II.1.4 Asignación directa de personal para reportes

En esta opción del menú principal, el administrador podrá asignar a cualquiera de los técnicos un reporte sin pasar por un verificador, pues en este caso, el verificador será el mismo administrador.

Para poder asignar un técnico a un reporte, deberá dar clic en la opción del menú correspondiente, y aparecerá una pantalla como la que se muestra a continuación:

The screenshot shows a web browser window displaying the INTRANET system. The page title is 'Plantilla General - Microsoft Internet Explorer'. The browser address bar shows 'http://intra.hacienda.gob.mx/intranet/links/seguras/helpdesk/admin/ConsultaAsignaDirecta.jsp'. The page content includes a header for 'Dirección General de Recursos Humanos' and a table of reports. The table has columns for 'Fecha y Hora', 'Clave Reporte', 'Nombre del Usuario', 'A verificar por', 'Ubicación', and 'Detalles'. To the right of the table, there is a section titled 'Asignar para operar a:' with a dropdown menu. The dropdown menu is open, showing a list of technicians: 'NO ASIGNADO', 'DANIEL OCTAVIO RUIZ Especial Especial', 'JOSE LUIS VELA', 'LUIS ERNESTO GUTIERREZ', 'MARIO ANTONIO MARTINEZ', 'OMAR URIEL SOSA', and 'RODRIGO LOPEZ'. Below the dropdown menu, there are two buttons labeled 'operadores extra'.

Fecha y Hora	Clave Reporte	Nombre del Usuario	A verificar por	Ubicación	Detalles
01/04/03 18:04	32	MARTINEZ MEDINA MARIO ANTONIO	VECL62070701	EDIF. D P.B.	<a href="#">detalles</a>
01/04/03 18:23	33	Blanca Monica Montiel Ortiz		EDIF. D P.B.	<a href="#">detalles</a>
01/04/03 18:25	34	Blanca Monica Montiel Ortiz		EDIF. D P.B.	<a href="#">detalles</a>
01/04/03 18:25	35	Carlos Alberto Jimenez Granados		EDIF. B P.B.	<a href="#">detalles</a>
01/04/03 18:29	36	Blanca Monica Montiel Ortiz		EDIF. D P.B.	<a href="#">detalles</a>

En esta pantalla aparece en cada línea un reporte, y al final de cada uno se muestra un menú con los nombres de los técnicos disponibles. Basta con sombrear el nombre mediante el teclado con la flecha hacia abajo o con el ratón (mouse), y dé clic en la imagen que aparece debajo de cada reporte para asignarle un técnico. La página se actualizará y reflejará el cambio solicitado.

Se podrá cambiar de técnico asignado mientras el reporte no haya sido atendido.

## II.1.5 Verificación de finalización de reporte y cerrado del mismo

En esta opción del menú, el administrador podrá visualizar los reportes que han sido finalizados y concluirlos, para lo cual deberá dar clic en la opción respectiva del menú. Aparecerá una pantalla como la siguiente:

Fecha y Hora de Inicio	Fecha y Hora de Finalización	Clave de Reporte	Nombre del Usuario	Ubicación	Atendió	Calificación Servicio	Comentarios del Usuario	Detalles	Concluir
24/03/03 17:01	24/03/03 17:03	24	Mario Antonio Martínez Medina	EDIF. D P.B.	MARIO ANTONIO MARTINEZ	E		detalles	Concluir
24/03/03 21:27	24/03/03 21:30	25	Mario Antonio Martínez Medina	EDIF. D P.B.	MARIO ANTONIO MARTINEZ	A		detalles	Concluir
24/03/04 22:07	24/03/04 22:12	30	Rocio Moreno Ramirez	EDIF. D P.B.	DANIEL OCTAVIO RUIZ	E	Comentarios	detalles	Concluir
12/04/04 22:00	12/04/04 22:01	31	Rocio Moreno Ramirez	EDIF. D P.B.	DANIEL OCTAVIO RUIZ	E	drgdrg	detalles	Concluir

Para concluir un reporte, dé clic en el botón CONCLUIR. La pantalla se actualizará y verá que el reporte que concluyó desaparece de la pantalla. Lo anterior denotará que la conclusión del reporte ha sido exitosa.



Observará que en la parte superior derecha aparece un hipervínculo con la leyenda AGREGAR TÉCNICO, que mostrará la siguiente pantalla:

Plantilla General - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://intrahacienda.gob.mx/Intranet/links/seguras/helpdesk/admin/tecnicos/intro\_tecnicos.jsp

aitavista Busque en la Web Traducir Resaltar Desactivado Última Búsqueda

Dirección General de Recursos Humanos  
Oficina Meyer

Consultas

Apellido Paterno

Apellido Materno

Nombre

Función OPERA

RFC OPERA  
VALIDA  
ADMIN

Inicio

Internet 05:26 p.m.

Esta pantalla contiene un formulario donde se capturarán los datos del técnico-usuario del sistema de soporte técnico de la Intranet de la Dirección General de Recursos Humanos (apellido paterno del usuario, apellido materno del usuario, nombre del usuario, la función que desempeñará [administrador, verificador u operador] y por último la filiación del usuario).

Una vez completo el formulario con los datos, dé clic en el botón AGREGAR y con ello realizará el movimiento de alta de un técnico-usuario.

Para poder editar la información de un técnico-usuario, dé clic en la imagen que aparece en la columna de EDITAR; aparecerá la siguiente pantalla:



Haga las modificaciones necesarias en los campos mostrados y dé clic en el botón de EDITAR; regresará a la pantalla principal de esta opción donde verá reflejados los cambios que efectuó.

Para borrar el registro de un técnico-usuario dé clic en la imagen que aparece en la columna ELIMINAR. Se actualizará la pantalla reflejando los cambios.

### II.1.7 Mantenimiento de la información de los equipos.

En esta opción del menú podrá actualizar la información de los equipos registrados y de los usuarios que los poseen. Al acceder a esta opción, se presentará la siguiente pantalla:

HELP DESK

Consultas

Introduzca el número de serie del equipo que desea editar.

[Alta de Equipo](#) [Regresar al Menu](#)

FILIACION	NOMBRE	EQUIPO MARCA	EQUIPO SERIE	EQUIPO INVENTARIO	EQUIPO MODELO	EDITAR/ELIMINAR
MEGE42111601	Salon	Ibm	82AT9AN	360065	350	

Para poder editar los datos de algún equipo registrado, escriba el número de serie del equipo a modificar dentro del campo de texto respectivo y dé clic en el botón de ENVIAR para que el sistema lo busque en la base de datos de equipos.

Si los datos que proporcionó para la búsqueda del equipo son correctos, aparecerá la descripción del mismo. Para poder editarlo, dé clic en la imagen en la columna EDITAR respectiva. Se mostrará la siguiente pantalla:

Plantilla General - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

← Atrás → Búsqueda Favoritos Multimedia

Dirección http://intrahacienda.gob.mx/Intranet/links/seguras/helpdesk/admin/equipos/adminEquipoEdita.jsp?serie=82AT9AN

altavista Busque en la Web Traducir Resolver Desactivado Última Búsqueda

SECRETARÍA DE HACIENDA Y CRÉDITO PÚBLICO SHCP

Dirección General de Recursos Humanos Oficina Mayor

Consultas

Regresar [Menu Principal](#)

FILLACION: MEGE42111601

NOMBRECORTO: Salon

MARCA: lbn

MODELO: 350

SERIE: 82AT9AN

INVENTARIO: 360065

AREA: PRESTACIONES AL PERSONAL

Actualizar

Listo Internet

Inicio 06:31 p.m.

En esta pantalla editará los datos del equipo y del usuario responsable, y dará clic en ACTUALIZAR, se ejecutará la acción y se regresará automáticamente a la pantalla principal de esta opción.

Para eliminar un equipo deberá buscarlo con el número de serie de la misma forma que en la opción anterior. Una vez localizado, dé clic en la imagen que aparece en la columna ELIMINAR respectiva.

La información del equipo se eliminará y regresará a la pantalla principal de esta opción.

Para poder dar de alta un equipo nuevo, dé clic en el hipervínculo ALTA DE EQUIPO. Aparecerá la siguiente pantalla:

Plantilla General - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia

Dirección http://intrahacienda.gob.mx/Intranet/links/seguras/helpdesk/admin/equipos/adminEquiposAlta0.jsp

altavista Busque en la Web Traducir Resaltar Desactivado Última Búsqueda

Consultas

[Editar Equipos](#) [Regresar al Menú principal](#)

Filiación

Nombre Completo

Equipo Marca

Equipo Modelo

Equipo Serie

Equipo Inventario

Ubicación Usuario

Area

Enviar

Listo Internet

Inicio 06:35 p.m.

En esta pantalla aparece un formulario en el que deberá escribir la filiación y nombre del usuario, la marca, modelo número de serie, número de inventario y ubicación del equipo, y por último el área administrativa a la que pertenece.

Dé clic en el botón ENVIAR para que se agregue este equipo a la base de datos. Automáticamente regresará a la pantalla principal de esta opción.

## II.1.8 Reportes que asigné al personal

Esta opción en el menú principal despliega qué reportes fueron asignados por el usuario a otros técnicos, además de poder visualizar el estado en que se encuentran. Para acceder a esta opción es necesario seleccionarla en el menú.

Aparecerá la siguiente pantalla:

The screenshot shows a Microsoft Internet Explorer browser window displaying the Intranet system. The browser's address bar shows the URL: `http://intra.hacienda.gob.mx/Intranet/links/seguras/helpdesk/admin/adminConsultaReportesPro.jsp`. The page header includes the logo of the Secretaría de Hacienda y Crédito Público and the text "Dirección General de Recursos Humanos Oficina Mayor".

Below the header, there is a navigation menu with icons for "Consultas", "Inicio", "Inicio", "Inicio", "Inicio", "Inicio", "Inicio", and "Inicio". A search bar is also present.

The main content area displays a table with the following data:

Fecha y Hora de Asignación de Responsable	Clave Reporte	Nombre del Usuario	Atendido por:	Estado del reporte
10/03/03 15:40	7	Mario Antonio Martínez Medina	DANIEL OCTAVIO RUIZ	CON
10/03/03 15:16	18	Mario Antonio Martínez Medina	MARIO ANTONIO MARTINEZ	CON
24/03/04 22:00	30	Rocio Moreno Ramirez	DANIEL OCTAVIO RUIZ	FIN
12/04/04 21:54	31	Rocio Moreno Ramirez	DANIEL OCTAVIO RUIZ	FIN

At the bottom of the browser window, the system tray shows the time as 06:41 p.m. and the status as "Listo".



**Sistema de Soporte Técnico de la INTRANET  
de la Dirección General de Recursos Humanos de la  
Secretaría de Hacienda y Crédito Público**

Mostrará un reporte con las siguientes columnas:

Fecha y Hora de Asignación de Responsable	Clave Reporte	Nombre del Usuario	Atendido por :	Estado del reporte
---	------------------	--------------------	----------------	--------------------

Esta opción es sólo de carácter informativo.

## II.1.9 Mis reportes asignados (Inicio de atención de un reporte)

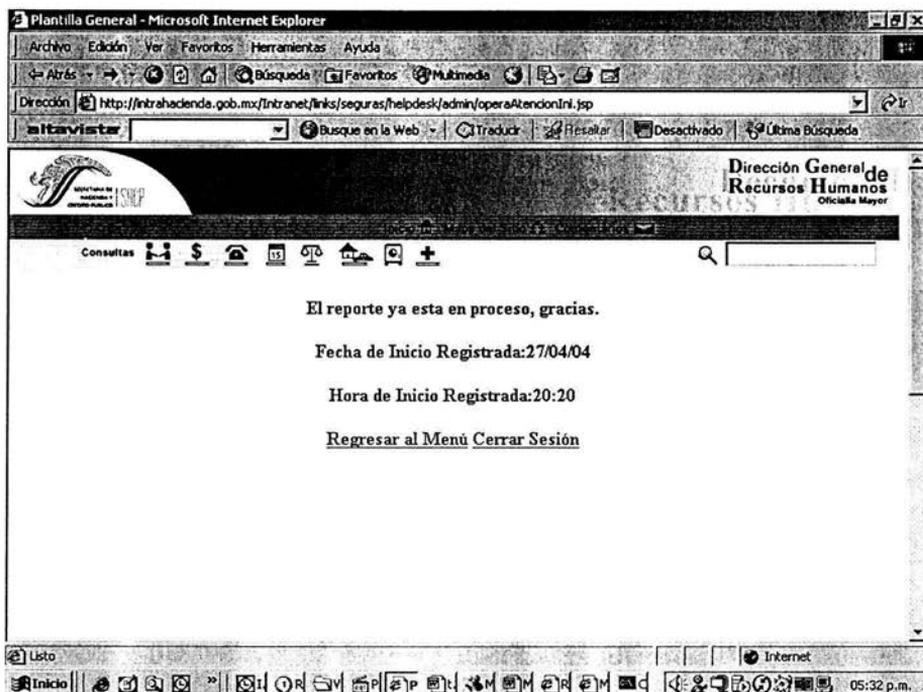
Esta opción mostrará qué reportes han sido asignados para atenderlos. Para acceder a esta opción basta con dar clic en la liga respectiva en el menú principal. Aparecerá la siguiente pantalla:

Se aprecia que del lado derecho de cada descripción de reporte aparece la opción de detalles y la opción de iniciar. Si da clic en la pantalla de DETALLES nos aparecerá una ventana con las características del reporte. Para salir de esta pantalla, cierre la ventana ya que es independiente de la pantalla principal.



**Sistema de Soporte Técnico de la INTRANET  
de la Dirección General de Recursos Humanos de la  
Secretaría de Hacienda y Crédito Público**

Al dar clic en el botón INICIAR, estará dando inicio a la atención de ese reporte, por lo que aparecerá un mensaje de confirmación como el siguiente:



Con ello concluimos la iniciación de un reporte.



Sistema de Soporte Técnico de la INTRANET  
de la Dirección General de Recursos Humanos de la  
Secretaría de Hacienda y Crédito Público

## II.1.10 Mis reportes asignados en Proceso (Finalización de un reporte)

En esta opción del menú principal podrá ver los reportes que ya se han iniciado con la opción de finalizar los mismos. Para acceder dé clic en el menú principal en la opción respectiva. Aparecerá la siguiente pantalla:

The screenshot shows a web browser window displaying the INTRANET interface. The browser's address bar shows the URL: `http://intrahacienda.gob.mx/intranet/links/seguras/helpdesk/admin/operapro.jsp`. The page header includes the SHCP logo and the text "Dirección General de Recursos Humanos Oficial Mayor". Below the header, there are navigation icons and a search bar. The main content area displays a table of assigned reports with the following data:

Fecha y Hora de Inicio	Fecha y Hora de Inicio	Clave Reporte	Nombre del Usuario	Ubicación	Detalles	Asignar para operar a:
01/04/03 14:54	28/07/03 17:12	27	Juana Castillo Cabrera	EDIF. D P.B.	<a href="#">detalles</a>	<a href="#">Concluir</a>
01/04/03 15:27	04/08/03 20:38	28	MARTINEZ MEDINA MARIO ANTONIO	EDIF. D P.B.	<a href="#">detalles</a>	<a href="#">Concluir</a>
01/04/03 17:43	28/07/03 17:14	29	Gloria Chimal Hernandez	EDIF. D 1er. Nivel.	<a href="#">detalles</a>	<a href="#">Concluir</a>

At the bottom of the browser window, the system tray shows the time as 04:52 p.m. and the status as "Listo".



Sistema de Soporte Técnico de la INTRANET  
de la Dirección General de Recursos Humanos de la  
Secretaría de Hacienda y Crédito Público

Aparecerá en cada línea los detalles de un reporte.

Si se desea obtener un detalle profundo de las características del reporte, dé clic en DETALLES. Para finalizar dé clic en la palabra CONCLUIR situada a la derecha de la pantalla. Aparecerá la siguiente pantalla:

Número de reporte: 27

Registrado por: Juana Castillo Cabrera

Atendido por: LUIS ERNESTO GUTIERREZ DELGADO

El reporte fue iniciado el día: 26/07/03 14:54

No. de Inventario Registrado: 3724098

**Esta parte debe ser llenada por el Ingeniero o Técnico:**

Breve descripción de la **solución** al problema:

**Esta parte debe ser llenada por el usuario:**

Calificación que le da al servicio:  Excelente  Bueno  Aceptable  Deficiente  Inaceptable

Comentarios acerca del servicio:

La persona que está finalizando el reporte describirá de manera concisa cómo solucionó el problema en el campo etiquetado BREVE DESCRIPCIÓN DE LA SOLUCIÓN AL PROBLEMA. Lo anterior con la finalidad de alimentar una base de datos de conocimientos para la solución de problemas en soporte técnico.



**Sistema de Soporte Técnico de la INTRANET  
de la Dirección General de Recursos Humanos de la  
Secretaría de Hacienda y Crédito Público**

Por otra parte, la persona que generó el reporte debe seleccionar la calificación que le da al servicio obtenido y, en caso de fuese necesario, escribir algún comentario.

Una vez completo el formulario, dé clic en **CONCLUIR** para finalizar el reporte. Por último aparecerá una pantalla con el mensaje de confirmación como la siguiente:

The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "Plantilla General - Microsoft Internet Explorer". The address bar contains the URL "http://intrahacienda.gob.mx/Intranet/links/seguras/helpdesk/admin/operAAtencionPro2.jsp". The browser interface includes a menu bar (Archivo, Edición, Ver, Favoritos, Herramientas, Ayuda), a toolbar with navigation buttons, and a search bar. The main content area displays the following text:

El reporte ha sido concluido satisfactoriamente con fecha 27/04/04 y a las 19:48 horas.  
[Imprima aquí el documento para firma](#)

Gracias.

[Regresar al Menu](#) [Cerrar Sesion](#)

The browser's status bar at the bottom shows the system tray with the time "05:00 p.m." and the word "Internet".