



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

AUTOMATIZACIÓN DE AUDITORIAS ADMINISTRATIVAS

DISEÑO DE UN SISTEMA PARA UNA ORGANIZACIÓN

CYNTHIA GABRIELA PARDO OROZCO

**ASESOR:
DR. RICARDO RIVERA SOLER**

MEXICO,D.F.

2003





Universidad Nacional
Autónoma de México



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.



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

AUTOMATIZACIÓN DE AUDITORIAS ADMINISTRATIVAS

**DISEÑO DE UN SISTEMA PARA UNA ORGANIZACIÓN
QUE PARA OBTENER EL TÍTULO DE:**

LICENCIADO EN ADMINISTRACIÓN

**P R E S E N T A:
CYNTHIA GABRIELA PARDO OROZCO**

**ASESOR:
DR. RICARDO RIVERA SOLER**

MEXICO,D.F.

2003



Dedicatorias

A mi padre:

Raúl Pardo Casas

*Por su apoyo en cada día de mi vida, su paciencia,
su interés, por brindarme una vida llena de felicidad,
por su confianza en cada momento y sobre todo
por su interminable fé en mí.*

A Elizabeth López:

*Por haber estado siempre presente y apoyando
a mi y a mi familia en cada momento difícil.
Por su amor incondicional.*

A mi hermano:

Cesar Pardo Orozco

Por apoyarme y cuidarme siempre.

Agradecimientos

A la UNAM:

Por permitirme ser parte de su comunidad y formarme como profesional.

*A la DCAA y las personas que la integran:
Por enriquecer mi formación, apoyarme
y hacerme parte de su equipo de trabajo,
y en especial al Maestro Juan Voutssas
por su apoyo durante mi estancia.*

A mi padre:

*Por apoyarme en cada momento e interesarse en
cada paso de mi vida profesional.*

A mis amigos y compañeros del grupo de usuarios LINUX:

*Por abrirme el panorama y enseñarme a
exigir mas de mi y del mundo que
me rodea en especial a Arturo Espinosa,
Miguel Ibarra, Miguel de Icaza y Mancha.*

A mi Asesor:

*Dr. Ricardo Rivera Soler,
Por su apoyo y ayuda.*

*Al L.A y C.P. Jorge Álvarez Anguiano,
Por inspirar este trabajo.*

A mis amigos:

*Enrique, Jesús, Gerardo, Carlos, Claudia, Verena,
Karla, Carmen, Marisol, Susana, Rodrigo, Joaquín
Por apoyarme en todo momento compartir
conmigo gratos e inolvidables momentos.*

Cynthia Gabriela Pardo Orozco.

ADVERTENCIAS

A lo largo del presente diseño de sistema para una organización se presentan temas con un nivel técnico, por tal motivo para una mejor comprensión se recomienda que los lectores se encuentren involucrados con conocimientos generales de Administración, particularmente Auditoría Administrativa, así como conceptos de Informática y desarrollo de sistemas para la Internet, debido a que el contenido de éste documento expone la transportación de la Auditoría administrativa a un sistema de cómputo.



INTRODUCCIÓN

El presente trabajo de diseño se compone de cinco capítulos comenzando por un análisis de la problemática que genera la causa de éste desarrollo en el Marco Problemático, a continuación, se muestra en el Marco Teórico un conjunto de documentos teóricos consultados para la realización de este trabajo con el fin de exponer las bases sobre las que se elaboró el desarrollo. La introducción de elementos conceptuales sobre los orígenes de este procedimiento (Auditoría Administrativa) y su metodología son mencionados en el Marco Conceptual, con el fin de exponer los lineamientos sobre los que se fijaron los eventos que el sistema llevaría a cabo.

Posteriormente se presenta el Marco Metodológico, en donde se exponen los elementos que formaron parte de éste desarrollo, así como las especificaciones que el sistema requiere y las funciones que será capaz de desarrollar, para terminar con el plan de difusión que se pretende dar al proyecto en el Marco Instrumental.

Con éste sistema se intenta aportar un elemento que facilite la realización de prácticas a nivel escolar, así como una herramienta funcional para aquellos practicantes y estudiosos del tema.

INDICE

1. MARCO PROBLEMÁTICO.....	7
1.1 Antecedentes y considerandos	7
1.2 Identificación del problema	7
1.3 Demarcación del Fenómeno	8
1.4 Conocimiento Empírico en el medio	9
1.5 Formato del cuestionario	10
1.6 Aplicación del cuestionario	11
1.6.1 Recopilación	12
1.7 Opiniones profesionales	13
1.7.1 Definición de las preguntas	13
1.7.2 Aplicación del cuestionario	14
1.8 Hipótesis preliminar	17
1.9 Objetivos	18
1.9.1 Personales	18
1.9.2 Particulares – Generales	18
2. MARCO TEORICO.....	19
2.1 Acopio de Libros	19
2.1.1 Libros de Estudio	19
2.2 Tesis	21
2.3 URL's	22
2.4 Otros	23
3. MARCO CONCEPTUAL.....	24
3.1 Antecedentes	24
3.2 Definiciones y aportaciones	26
3.2.1 Enfoque de William P. Leonard	26
3.2.1.1 Concepto	26
3.2.1.2 Aportaciones	26
3.2.1.3 Metodología	27
3.3 Enfoque de José A. Fernández Arena	29
3.3.1 Concepto	29
3.3.2 Metodología	29
3.4 Enfoque de Víctor Rubio Ragazzoni	30
3.4.1 Concepto y aportaciones	30
3.4.2 Metodología	30
3.5 Enfoque de Enrique Benjamín Franklin	33
3.5.1 Concepto	33
3.5.2 Objetivos	33
3.6 L.A. y C.P Jorge Alvarez Anguiano	37
3.6.1 Concepto	37

3.6.2 Metodología	37
4. MARCO METODOLÓGICO.....	38
4.1 Variables	38
4.2 Variables de control	39
4.3 Hipótesis definitiva	39
4.4 Definición del Universo	39
4.5 Determinación de la muestra	39
4.6 Definición del método de investigación	40
4.7 Costo de la investigación	40
4.8 Colaboradores y apoyos	40
4.9 Construcción del cuestionario	41
4.10 Cuestionario Piloto	43
4.11 Cuestionario Definitivo	45
4.12 Realización de investigación	47
4.12.1 Análisis del desarrollo	51
4.12.1.1 Requerimientos de información	51
4.12.1.1.1 Los actores	51
4.12.1.1.2 La información	51
4.12.1.2 Los objetivos del sistema Audit-Admin	52
4.12.1.3 Lista de los eventos del sistema Audit-Admin	52
4.12.1.3.1 Registrar un Usuario	52
4.12.1.3.2 Modificar un Usuario	52
4.12.1.3.3 Crear una Empresa	52
4.12.1.3.4 Editar una Empresa	52
4.12.1.3.5 Eliminar una Empresa	52
4.12.1.3.5.0 Crear una auditoría	53
4.12.1.3.5.1 Definir subfunciones a auditar	53
4.12.1.3.5.2 Definir peso de cada subfunción	53
4.12.1.3.5.3 Definir factores a evaluar	53
4.12.1.3.5.4 Definir la distribución de los puntos	53
4.12.1.3.5.5 Definir preguntas	53
4.12.1.3.5.6 Evaluar preguntas	54
4.12.1.3.6. Editar los datos generales de una Auditoría	54
4.12.1.3.7 Borrar una auditoría	54
4.12.1.3.8 Generar Reportes	54
4.12.1.3.9 Operaciones con catálogos	54
4.12.1.4 Diagrama de contexto	54
4.12.1.4.1 Breve explicación de los procesos	54
4.12.1.4.1.1 Usuario	54
4.12.1.4.1.2 Auditor	56
4.12.1.4.1.3 Administrador	57
4.12.1.4.1.4 Supervisor	57
4.13 Explicación del proceso a desarrollar para realizar una auditoría	57

4.14 Tratamiento sistematizado de la información	60
4.15 Análisis de los resultados	70
4.16 Conclusión sobre los resultados	84
4.17 Aprobación o desaprobación de la hipótesis.	84
5. MARCO INSTRUMENTAL.....	85
5.1 Propuestas de acción	85
5.2 Plan y programa de trabajo	86
6. GLOSARIO.....	87
7. BIBLIOGRAFÍA.....	88
8. ANEXOS (código de scripts en CD).	

1. MARCO PROBLEMÁTICO

1.1 Antecedentes y considerandos.

Una parte importante del proceso administrativo referente a la comprobación de la realización de objetivos, así como la supervisión y medición de como se llevaron a cabo; nos lleva a analizar el control como parte final de un ciclo que es el proceso administrativo, y, que sirve a la organización para otorgar un panorama de lo hecho, y así formular nuevas expectativas o bien planes de acción ya sea para corregir resultados o alcanzar nuevas metas.

Entre los métodos de control podemos hablar de aquellos que se aplican a cuestiones financieras de la organización y los que se basan en la medición de operaciones no financieras.

Los fenómenos económicos a que se expone la empresa hoy en día hacen necesaria la automatización de operaciones así como la adopción de nuevos métodos para registrar y obtener datos reales en la organización.

Por ello representa un problema el atraso en las herramientas para realizar Auditorías Administrativas, debido a que implican largas jornadas de papeleo para el comienzo de la investigación, la toma de datos, así como, la emisión de resultados. Esto sin contar que el proceso debe realizarse de nueva cuenta pasado un periodo con el fin de observar los cambios realizados, formarse nuevas expectativas, o bien adoptar distintos métodos para mejorar el funcionamiento de la organización.

La falta de herramientas que faciliten los pasos en una Auditoría Administrativa, provocan un incremento en los gastos de la organización que requiera de estas evaluaciones, y correrá el riesgo de que algunos de los datos tomados para ella, caduquen.

1.2 Identificación del problema

El registro de las auditorías actualmente se lleva a cabo con herramientas manuales que pueden provocar equivocaciones o pérdida de información, así como lentitud, debido a que dependen de procesos manuales que implican contar solamente con respaldos en papel de la toma de datos realizada.

Por otro lado requieren, en el peor de los casos, ser nuevamente trasladados a una aplicación no desarrollada para ello, como una hoja de cálculo o un procesador de palabras, con el fin de contabilizarlos para generar estadísticas y resultados útiles, si se manejan altos volúmenes de información (tomando en cuenta que se esté auditando una empresa de grandes dimensiones), lo que puede volverse un proceso lento y engorroso para el personal encargado de migrar estos datos.

Actualmente se utiliza en la Facultad de Contaduría y Administración (a nivel de prácticas) un Sistema desarrollado por el C.P. y L.A.E. Jorge Álvarez Anguiano y el C.P. y L.A.E. Mario Compean Toledo. El Sistema realiza la mayoría de las operaciones necesarias para registrar Auditorías Administrativas. Sin embargo, el sistema tiene los siguientes problemas:

- Es necesario cargar el sistema mediante un disquete (disco removible) donde todo lo que se elabore será guardado. Como consecuencia, es imposible guardar altos volúmenes de información, pues el usuario está limitado a la capacidad que tiene el disco flexible.
- El sistema fue desarrollado en clipper, y solo puede ser usado en modo texto desde el sistema operativo DOS. Aunque su funcionamiento es adecuado, los usuarios lo consideran poco amigable por no contar con el uso de ventanas, facilidad para imprimir, migrar los datos a otras aplicaciones o bien aprovechar las facilidades que brindan los ambientes gráficos como Windows (XP, 2000, 98, 95 etc.), Linux (Gnome, KDE etc.), Solaris (CDE), por mencionar algunos.
- Por otro lado sólo puede utilizarse en la plataforma DOS, lo que imposibilita el uso a aquellos usuarios que no cuentan con ésta plataforma.
- El uso de discos flexibles implica el riesgo de pérdida de información debido a que están hechos de materiales muy delicados al medio ambiente.
- La capacidad de compartir datos está restringida a que se comparta el disco de un usuario a otro, lo que impide que varias personas trabajen en la misma auditoría desde lugares distintos.
- No es posible acceder a datos históricos y debido a que el lenguaje en que fue programado está en desuso no es factible agregar nuevas funcionalidades al sistema.

Si bien existen actualmente equipos de cómputo capaces de resguardar datos y facilitar la captura de las Auditorías Administrativas, se carece de herramientas de uso específico para Auditorías Administrativas que:

- Permitan el seguimiento de las Auditorías de manera dedicada a lo largo del tiempo.
- Aprovechen los beneficios de independencia de plataforma que brindan las aplicaciones WEB, sin que sea necesario cargar con la información en un disco y permitiendo que pueda ser accedida desde distintos lugares.
- Aprovechen los beneficios que brindan los nuevos sistemas operativos con ambientes gráficos.
- Permitan migrar fácilmente los datos a otras aplicaciones.
- Aprovechen los beneficios de los sistemas multiusuario en donde existen restricciones a la información y posibilidad de compartir datos dentro de un equipo de trabajo.

1.3 Demarcación del Fenómeno

Este desarrollo se realiza con el fin de que todas aquellas personas interesadas en realizar Auditorías Administrativas para fines de práctica encuentren una plataforma que les permita registrar y evaluar datos pertenecientes a una organización, así como todas aquellas empresas interesadas en adoptar costumbres de evaluación periódicas a bajo costo que les permitan observar las fluctuaciones de sus niveles de productividad.

1.4 Conocimiento empírico en el medio

Para obtener información acerca de las necesidades existentes para llevar a cabo el desarrollo de un sistema se realizó un cuestionario, el cual se aplicó a diversas personas involucradas y con conocimientos del tema.

Pregunta	Justificación	Respuesta esperada
1. ¿ Que orden de importancia del 1 al 3 tienen los procesos de medición de la productividad?	Conocer la prioridad que se le da a la evaluación de la organización.	1
2. ¿Considera a la Auditoría Administrativa un elemento importante para la medición de la productividad y los objetivos organizacionales?	Ubicar la importancia que tiene la Auditoría Administrativa para el entrevistado.	SI
3. ¿Cree que los procesos para llevar a cabo auditorías son actualmente demasiado largos y poco automatizados?	Remarcar la importancia en la rapidez de la información requerida.	SI
4. ¿Cree usted que sería fácil detectar problemas de productividad y cumplimiento de objetivos sin realizar evaluaciones por medio de auditorías?	Ver que tan importante es realizar Auditorías Administrativas como medio de evaluación.	NO
5. ¿Considera que es posible evaluar cada parte de la organización mediante la Auditoría Administrativa?	Conocer los alcances de la Auditoría Administrativa en la organización.	SI

1.5 Formato del cuestionario

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CONTADURÍA Y
ADMINISTRACIÓN

CUESTIONARIO

Nombre: _____

OBJETIVO: Obtener una opinión empírica para el tema de tesis "Automatización de Auditorías Administrativas" aplicado por Cynthia Gabriela Pardo Orozco.

INSTRUCCIONES: Responda las siguientes preguntas.

1. ¿ Que orden de importancia del 1 al 3 tienen los procesos de medición de la productividad o la evaluación general del comportamiento de la organización?

1 (mayor) ()
2 ()
3 (menor) ()

2. ¿Considera a la Auditoría Administrativa un elemento importante para la medición de la productividad y los objetivos organizacionales?

() Sí () No

3. ¿Cree que los procesos para llevar a cabo auditorías son actualmente demasiado largos y poco automatizados?

() Sí () No

4. ¿Cree usted que sería fácil detectar problemas de productividad y cumplimiento de objetivos sin realizar evaluaciones por medio de auditorías?

() Sí () No

5. ¿Considera que es posible evaluar cada parte de la organización mediante la Auditoría Administrativa?

() Sí () No

1.6 Aplicación del cuestionario

Las personas a las que se les aplicó el cuestionario fueron las siguientes:

L.A Enrique Suarez Zuñiga
Desarrollador Web
Camara de Diputados.

C.P Claudia Arizmendi Jiménez
Jefe de Nomina y Contabilidad.
Prestamos.

Pasante de Contaduría Verena Gama Ugalde
Técnico Académico .
DCAA.

Miguel Ibarra
Analista y Programador.
Ximian.

Karla López Satow.
Becaria de Personal.
Camara de la industria de la construcción.

1.6.1 Recopilación

1. ¿ Que orden de importancia del 1 al 3 tienen los procesos de medición de la productividad o la evaluación general del comportamiento de la organización?

	<i>Enrique Suarez Zuñiga</i>	<i>Claudia Arizmendi</i>	<i>Verena Gama Ugalde</i>	<i>Miguel Ibarra</i>	<i>Karla López Satow</i>
1	X		X	X	X
2		X			
3					

Conclusión: Las personas entrevistadas coincidieron en que se debe dar gran importancia a la evaluación de la organización y los procesos de medición de la productividad.

2. ¿ Considera a la Auditoría Administrativa un elemento importante para la medición de la productividad y los objetivos organizacionales?

<i>Enrique Suarez Zuñiga</i>	<i>Claudia Arizmendi</i>	<i>Verena Gama Ugalde</i>	<i>Miguel Ibarra</i>	<i>Karla López Satow</i>
SI	SI	SI	SI	SI

Conclusión: Todos coincidieron en que la Auditoría Administrativa es un elemento importante para la medición de la productividad y objetivos organizacionales.

3. ¿ Cree que los procesos para llevar a cabo auditorías son actualmente demasiado largos y poco automatizados ?

<i>Enrique Suarez Zuñiga</i>	<i>Claudia Arizmendi</i>	<i>Verena Gama Ugalde</i>	<i>Miguel Ibarra</i>	<i>Karla López Satow</i>
Sí	Sí	Sí	Sí	Sí

Conclusión: Es necesario contar con herramientas que mejoren la realización de Auditorías Administrativas dentro de la organización.

4. ¿ Cree usted que sería fácil detectar problemas de productividad y cumplimiento de objetivos sin realizar evaluaciones por medio de auditorías ?

<i>Enrique Suarez Zuñiga</i>	<i>Claudia Arizmendi</i>	<i>Verena Gama Ugalde</i>	<i>Miguel Ibarra</i>	<i>Karla López Satow</i>
No	No	No	No	No

Conclusión: Las auditorías son necesarias para llevar a cabo evaluaciones que permitan conocer el estado en que se encuentra la organización .

5. ¿ Considera que es posible evaluar cada parte de la organización mediante la Auditoría Administrativa?

<i>Enrique Suarez Zuñiga</i>	<i>Claudia Arizmendi</i>	<i>Verena Gama Ugalde</i>	<i>Miguel Ibarra</i>	<i>Karla López Satow</i>
SI	SI	SI	SI	SI

Conclusión: Coincidieron en que el alcance de la auditoría puede elevarse a todos los niveles y áreas.

1.7 Opiniones profesionales

Para obtener más información acerca del trabajo de investigación, se aplicó un cuestionario a personas profesionales en el tema, personas que dirigen áreas de personal o bien trabajan como catedráticos de la materia. De acuerdo a la información que se esperaba obtener.

1.7.1 Definición de las preguntas

Pregunta	Justificación	Respuesta esperada
1. ¿Considera a la Auditoría Administrativa un elemento importante dentro de la medición de la productividad?	Ubicar la importancia que tiene la Auditoría Administrativa para el entrevistado.	<i>SI</i>
2. ¿Considera que es posible proponer mejoras de automatización en la elaboración de las auditorías?	Conocer si los entrevistados consideran que existen aún mejoras por realizar en la auditoría.	<i>SI</i>
3. ¿Considera que existen constantes errores cuando se realiza papeleo o bien registros manuales en la Auditoría Administrativa?	Remarcar la importancia del registro automático.	<i>SI</i>
4. ¿Cree usted que sería fácil detectar problemas de productividad y cumplimiento de objetivos sin realizar evaluaciones por medio de auditorías?	Ver que tan importante es realizar Auditorías Administrativas como medio de evaluación.	<i>NO</i>

<p>5. ¿Considera que es necesario guardar la información de auditorías anteriores para analizarla contra la de nuevas auditorías?</p>	<p>Verificar que los entrevistados estén interesados en el resguardo de archivos históricos.</p>	<p><i>SI</i></p>
--	--	------------------

1.7.2 Aplicación del cuestionario

Las personas a las que se les aplicó el cuestionario fueron las siguientes:

Rodrigo Tablas Hernández

Director de Ventas ITC.

Carmen Zavaleta

Director de Relaciones Públicas.

Café Juquila SA de CV.

L.A. Marisol Oropeza

Becaria en Dirección general de empleo y capacitación.

Gobierno del DF.

Guillermo González Garrido

Jefe depto de Mensajería.

PNUMA(United Nations Enviroment Program) – ONU.

1¿ Considera a la Auditoría Administrativa un elemento importante dentro de la medición de la productividad?				
	Rodrigo Tablas Hernández	Carmen Zavaleta	L.A. Marisol Oropeza	Guillermo González Garrido
	SI	SI	SI	SI
Conclusión: Las personas entrevistadas coincidieron en que se debe dar gran importancia a la evaluación de la organización y los procesos de medición de la productividad .				
2¿ Considera que es posible proponer mejoras de automatización en la elaboración de las auditorías?				
	Rodrigo Tablas Hernández	Carmen Zavaleta	L.A. Marisol Oropeza	Guillermo González Garrido
	SI	SI	SI	SI
Conclusión: Todos coincidieron en que la Auditoría Administrativa es posible realizar propuestas de mejoras.				
3¿ Considera que existen constantes errores cuando se realiza papeleo o bien registros manuales en la AA?				
	Rodrigo Tablas Hernández	Carmen Zavaleta	L.A. Marisol Oropeza	Guillermo González Garrido
	Sí	Sí	Sí	Sí
Conclusión: Es necesario contar con herramientas que mejoren la realización de Auditorías Administrativas dentro de la organización, pues los procesos manuales aumentan las posibilidades de generar errores.				
4¿ Cree usted que sería fácil detectar problemas de productividad y cumplimiento de objetivos sin realizar evaluaciones por medio de auditorías ?				
	Rodrigo Tablas Hernández	Carmen Zavaleta	L.A. Marisol Oropeza	Guillermo González Garrido
	No	No	No	No
Conclusión: Las auditorías son necesarias para llevar a cabo evaluaciones que permitan conocer el estado en que se encuentra la organización .				
5¿ Considera que es necesario guardar la información de auditorías anteriores para analizarla contra la de nuevas auditorías?				

<i>Rodrigo Tablas Hernández</i>	<i>Carmen Zavaleta</i>	<i>L.A. Marisol Oropeza</i>	<i>Guillermo González Garrido</i>
SI	SI	SI	SI
Conclusión: Coincidieron en que el alcance de la auditoría puede elevarse a todos los niveles ya áreas.			

1.8 Hipótesis preliminar

Variable independiente

- El sistema generará información que servirá para tomar decisiones, dichas decisiones son tomadas con base en la votación de un comité dependiendo de las políticas e intereses pertenecientes a la organización.
- La veracidad de los datos proporcionados en una Auditoría dependerá del comité o aquellas personas encargadas de proveer de información al auditor.
- La rapidez con que la auditoría sea realizada dependerá también de la disposición que preste la organización y las personas facultadas para proveer de información a la auditoría.

Variable dependiente

- La falta de herramientas que faciliten los pasos en una Auditoría Administrativa, provocan un incremento en los gastos de la organización que requiera de estas evaluaciones.
- La lentitud con que se realicen las auditorías a raíz de esto, provocan lentitud en la toma de decisiones.
- La falta de una herramienta que aproveche los beneficios de migración de datos entre aplicaciones, así como envío para facilitar compartir la información

Proposición positiva

- *En el desarrollo de una Auditoría Administrativa una vez que los datos han sido recopilados, facilitará la agilización en el registro y consulta de los mismos.*
- *En el desarrollo de una Auditoría Administrativa una vez que los datos han sido recopilados, facilitarán la generación de información oportuna para la formación de nuevos objetivos o correcciones en los procesos para mejorar la productividad.*
- *En el desarrollo de una Auditoría Administrativa una vez que los datos han sido recopilados, facilitarán el diagnóstico de funcionamiento de las distintas áreas evaluadas.*
- *En el desarrollo de una Auditoría Administrativa una vez que los datos han sido recopilados, facilitarán la detección oportuna de problemas y puntos fuera de la observación diaria.*

Proposición negativa

- *En el desarrollo de una Auditoría Administrativa una vez que los datos han sido recopilados, no facilitará la cooperación del personal con quién se realiza la auditoría.*

- *En el desarrollo de una Auditoría Administrativa una vez que los datos han sido recopilados; no facilitará la coordinación de los diferentes elementos del comité para definir las áreas críticas.*
- *En el desarrollo de una Auditoría Administrativa una vez que los datos han sido recopilados; no facilitará las operaciones de investigación y observación que el auditor deba realizar.*
- *En el desarrollo de una Auditoría Administrativa una vez que los datos han sido recopilados; no facilitará la generación de recomendaciones que son producto del análisis de un experto sobre los resultados de una auditoría.*

1.9 Objetivos

1.9.1 Personales

1. Obtener el título profesional como Licenciado en Administración por medio del desarrollo de un sistema de Auditoría Administrativa, de acuerdo al Reglamento General de Exámenes, Capítulo IV Exámenes Profesionales y de Grado, Artículos del 44 al 56.
2. Ampliar mis conocimientos sobre el proceso administrativo aplicado al desarrollo de sistemas, así como el uso de sistemas de evaluación dentro de las organizaciones.
3. Aplicar los conocimientos obtenidos en la Licenciatura de administración realizando una aportación práctica para la agilización de las Auditorías Administrativas.
4. Aprender y mejorar mis capacidades de abstracción y análisis al transportar el proceso de Auditoría Administrativa a un sistema de cómputo mediante el uso de bases de datos, servicios de red y un lenguaje de programación.

1.9.2 Particulares - Generales

1. Aportar un sistema para la realización de prácticas de Auditoría Administrativa en la Facultad de Contaduría y Administración.
2. Realizar una aportación intelectual a la sociedad y a la comunidad de Software Libre en México.
3. Probar que la Auditoría Administrativa es una herramienta de gran importancia que debe ser explotada en el desarrollo y mejora de las organizaciones.

2. MARCO TEORICO

2.1 Acopio de Libros.

2.1.1 Libros de Estudio

Nombre: **La auditoria Administrativa**

Autor: José Antonio Fernández Arena

Editorial: Diana

Edición: 2da. Edición

ISBN: 968-13-2165-0

Capítulo 1

Definición de la Auditoria Administrativa.

La auditoria administrativa es la revisión objetiva, metódica y completa (I); de la satisfacción de los objetivos institucionales (II); con base en los niveles jerárquicos de la empresa (III); en cuanto a su estructura y (IV); con la participación individual de los integrantes de la institución.

Capítulo 2

Escala de apreciación

En un principio las calificaciones serán totalmente empíricas pero conforme se vaya desarrollando esta disciplina irá recabando experiencias muy valiosas para establecer evaluaciones cada día más objetivas. Aun cuando la determinación de una calificación sea empírica no se puede decir que los factores a considerar sean improvisados a última hora. Estos serán bastante más sencillos de implantar y determinar mediante el establecimiento de convenciones.

Capítulo 3

Método de William P. Leonard

El esquema general esta compuesto de Un Examen, a)Áreas de Estudio (funciones específicas, departamentos, divisiones, la empresa); b)Detalles a estudiar (planes y objetivos, la estructura de la empresas, políticas y sistemas, métodos de control, recursos humanos y físicos, estándares, medición de resultados). Una Evaluación a) Proceso (Influencia económica, estructura adecuada, adecuación de los controles, métodos de protección, causas de variación, utilización de los recursos, métodos para trabajar) b)Análisis e interpretación (Estudios de los elementos, diagnosis detallada, determinar propósitos, deficiencias, balance analítico, prueba de eficiencia, búsqueda de problemas, soluciones, alternativas, métodos simplificados)

Capítulo 4

Método del American Institute of Management

El esquema general separa las funciones a investigar y el tratamiento que se dará a la organización dependiendo del tipo de empresa, ya sea de tipo religioso, de tipo educativo, un hospital, una institución sin fines de lucro, o bien una institución confines de lucro.

Capítulo 5

Método de análisis de Excelencia de la revista Fortune

El estudio considera 306 empresas repartidas en 32 grupos industriales; todas ellas forman parte de la clasificación de la revista en su lista de las quinientas compañías más importantes en productos y servicios.

Las respuestas recibidas y tabuladas en las ocho categorías analizadas hacen posible una calificación de las empresas y su calificación ponderada. Las respuestas dan una seguridad estadística que permite validar las opiniones de los expertos.

Capítulo 6

Método de Auditoria del control de calidad total

La auditoria de control de calidad, sirve para hacer el seguimiento del proceso de control. Realiza el diagnostico del caso, y muestra como corregir las fallas que pueda tener. Revisar la calidad es estudiar la de un producto determinado tomando muestras de tiempo ya sea dentro de la empresa o en el mercado

Capítulo 7

Método de análisis factorial desarrollado por investigadores del Banco de México.

El esquema general estudia el medio ambiente, política y dirección, productos y procesos, financiamiento, medios de producción, fuerza de trabajo, suministros, actividad productora, mercadeo, contabilidad y estadísticas.

Capítulo 8

Método de la revisión del proceso administrativo desarrollado por Manuel D' Azaola.

El esquema general es 1)Fijar los objetivos, proporcionar los medios humanos y materiales para alcanzarlos y organizar ambos. 2) Dotar a la empresa de los recursos económicos suficientes para que desarrolle sus actividades con normalidad.3)Contratar al personal idóneo para cada función y adiestrarlo para que cumpla de manera eficiente. 4) Diseñar artículos y servicios que satisfagan necesidades humanas y que tengan una demanda real, para que la empresa obtenga un beneficio razonable al realizarlos.

Capítulo 9

Método de José Antonio Fernández Arena.

Los cuestionarios que se utilizan para realizar la auditoria administrativa están formados con base en los principios de esa disciplina del saber humano cumpliendo con los sig. objetivos: de servicio, social y económico.

Nombre: **Análisis y valuación de puestos**

Autor: Rafael J. Hernández

Editorial: UNAM

Edición: 5ta. Edición

Capítulo 1

Definición de Análisis de puestos

Es un método de investigación que sirve para conocer los elementos que integran un puesto determinado de trabajo.

Capítulo 2

Modelo de un análisis de labores

Es realizado con el propósito de jerarquizar los puestos de una empresa y, así mismo, comparar sus salarios con los de la competencia, para actuar en consecuencia.

Capítulo 3

Valuación de puestos

El objetivo de este proceso es determinar la importancia y utilidad relativa de los distintos puestos que se dan en una empresa, para establecer pagos diferenciales y justos, en función de la jerarquización, y, con ello, corregir las posibles deficiencias y evitar la injusticia en los salarios.

Capítulo 4

Sistemas de valuación

Método de alineamiento, método de escalas o grados predeterminados.

Capítulo 5

Método de Comparación de factores

Ordenación de los puestos tipo en función de diversas características generales, asignación de un valor monetario a cada uno de dichos factores, combinación moderada de ambos resultados, para establecer un rango entre los puestos valuados.

Capítulo 6

Método de valuación por puntos

Sistema técnico mediante el cual se asignan ciertas cantidades de valor denominadas puntos a cada una de las características fundamentales de los puestos, mismas que se designan con el nombre de factores.

2.2 Tesis

Título: **Desarrollo de la Internet de la Coordinación General de Servicios Financieros de la H. Cámara de Diputados LVII Legislatura**

Autor: Enrique Suárez Zúñiga

Fecha: 2001

Carrera: Licenciatura en Administración

Facultad: Facultad de Contaduría y Administración

Ubicación: 001-00622-s1-2001

Título: **Generación de código Java a partir de diseños UML**
Autor: Jesús García Flores
Fecha: 2002
Carrera: Licenciatura en Informática
Facultad: Facultad de Contaduría y Administración
Ubicación: 001-00623-G5-2002

2.3 URL's

Dirección:

<http://www.informador.com.mx/Lastest/feb97/07feb97/Univ2.htm#Pres2>

Contenido:

La función de la auditoria Administrativa en el control interno de las empresas Lic. Nicolás Hernández Rosales. Centro Universitario de Los Altos. U. de G.

Dirección:

<http://www.benjaminfranklin.8k.com/artic09.htm>

Contenido:

Auditoría Administrativa: Conceptos Fundamentales De la Revista Emprendedores Num.66
Noviembre - Diciembre 2000 por Enrique Benjamín Franklin .

Dirección:

<http://www.geocities.com/gehg48/APUNTES.html>

Contenido:

Apuntes de Auditoría Administrativa por Gilberto Eduardo Hazas

Dirección:

<http://www.programacion.com/>

Contenido:

Cursos en línea de programación

Dirección:

<http://www.geocities.com/SiliconValley/Station/8266/perl/>

Contenido:

Cursos en línea de programación con perl

2.4 Otros

Título: Apuntes de Auditoría Administrativa

Autor: L.A. y C.P. Jorge Álvarez Anguiano

Comentario:

Los apuntes analizan el proceso de auditoría administrativa desde su planeación hasta su ejecución y seguimiento, explicando y llevando a cabo el seguimiento de un ejemplo.

3. MARCO CONCEPTUAL

3.1 Antecedentes

Origen e Historia

Los primeros antecedentes escritos en materia de Auditoría Administrativa parecen provenir del Instituto de Auditores Internos Norteamericanos: una discusión de 1954 sobre el alcance de la Auditoría Interna de Operaciones Técnicas, y un artículo de Arthur H. Kems, de la Standar Oil Company of California, aparecida en 1948 sobre Auditoría de Operaciones, citados en el libro de Bradford: Operational Auditing Handbook de 1964. En 1959 se publicó un libro por Víctor Lazzano, el cual se titula Systems and Procedures, a Handbook of Bussines se publicó en 1962, bajo el título de The Management Audit. Las obras anteriores han sido traducidas al español. En México se tiene un antecedente en el libro: Guía para Estudios de Economía Industrial, de Alfred W. Klein y Nathan Grabinsky, publicado por el departamento de Investigaciones del Banco de México, S.A.

En el año de 1935, James O. McKinsey, en el seno de la American Economic Association sentó las bases para lo que él llamó "Auditoría Administrativa", la cual, en sus palabras, consistía en "una evaluación de una empresa en todos sus aspectos, a la luz de su ambiente presente y futuro probable."

Más adelante, en 1953, George R. Terry, en Principios de Administración, señala que "La confrontación periódica de la planeación, organización, ejecución y control administrativos de una compañía, con lo que podría llamar el prototipo de una operación de éxito, es el significado esencial de la Auditoría Administrativa."

Dos años después, en 1955, Harold Koontz y Ciryll O´Donnell, también en sus Principios de Administración, proponen a la auto-auditoría, como una técnica de control del desempeño total, la cual estaría destinada a "evaluar la posición de la empresa para determinar dónde se encuentra, hacia dónde va con los programas presentes, cuáles deberían ser sus objetivos y si se necesitan planes revisados para alcanzar estos objetivos."

El interés por esta técnica llevan en 1958 a Alfred Klein y Nathan Grabinsky a preparar El Análisis Factorial, obra en cual abordan el estudio de "las causas de una baja productividad para establecer las bases para mejorarla" a través de un método que identifica y cuantifica los factores y funciones que intervienen en la operación de una organización.

Transcurrido un año, en 1959, ocurren dos hechos relevantes que contribuyen a la evolución de la Auditoría Administrativa: 1) Víctor Lazzaro publica su libro de Sistemas y Procedimientos, en el cual presenta la contribución de William P. Leonard con el nombre de Auditoría Administrativa y, 2) The American Institute of Management, en el Manual of Excellence Managements integra un método para auditar empresas con y sin fines de lucro, tomando en cuenta su función, estructura, crecimiento, políticas financieras, eficiencia operativa y evaluación administrativa.

El atractivo por el tema se extiende al ámbito académico y, en 1960, Alfonso Mejía Fernández, de la Escuela Nacional de Comercio y Administración de la Universidad

Nacional Autónoma de México, en su tesis profesional *La Auditoría de las Funciones de la Gerencia de las Empresas*, realiza un recuento de los aspectos estructurales y funcionales que el nivel gerencial de las empresas debe contemplar para aplicar una Auditoría Administrativa.

Para 1962, Roberto Macías Pineda, de la Escuela Superior de Comercio y Administración del Instituto Politécnico Nacional, dentro del programa de doctorado en ciencias administrativas, en la asignatura *Teoría de la Administración*, destina un espacio para presentar un trabajo de Auditoría Administrativa.

Por otra parte, en 1964, Manuel D´Azaola S., de la Escuela Nacional de Comercio y Administración de la Universidad Nacional Autónoma de México, en su tesis profesional *La Revisión del Proceso Administrativo*, considera la necesidad de que las empresas analicen su comportamiento a partir de la revisión de las funciones de dirección, financiamiento, personal, producción, ventas y distribución, así como registro contable y estadístico.

A finales de 1965, Edward F. Norbeck da a conocer su libro *Auditoría Administrativa*, en donde define el concepto, contenido e instrumentos para aplicar la auditoría. Asimismo, precisa las diferencias entre la Auditoría Administrativa y la auditoría financiera, y desarrolla los criterios para la integración del equipo de auditores en sus diferentes modalidades.

En 1966, José Antonio Fernández Arena, presenta la primera versión de su texto *La Auditoría Administrativa*, en la cual desarrolla un marco comparativo entre diferentes enfoques de la Auditoría Administrativa, presentando una propuesta a partir de su propia visión de la técnica.

Más adelante, en 1971, se generan dos nuevas contribuciones: Agustín Reyes Ponce, en *Administración de Personal*, dedica un apartado para tratar el tema, ofreciendo una visión general de la Auditoría Administrativa, en tanto que William P. Leonard publica *Auditoría Administrativa: Evaluación de los Métodos y Eficiencia Administrativos*, en donde incorpora los conceptos fundamentales y programas para la ejecución de la Auditoría Administrativa.

Para 1977, se suman las aportaciones de dos autores en la materia. Patricia Diez de Bonilla en su *Manual de Casos Prácticos sobre Auditoría Administrativa*, propone aplicaciones viables de llevar a la práctica y, Jorge Álvarez Anguiano, en *Apuntes de Auditoría Administrativa* incluye un marco metodológico que permite entender la Auditoría Administrativa de manera por demás accesible.

En 1978, la Asociación Nacional de Licenciados en Administración, difunde el documento *Auditoría Administrativa*, el cual reúne las normas para su implementación en organizaciones públicas y privadas.

Poco después, en 1984, Robert J. Thierauf presenta *Auditoría Administrativa con Cuestionarios de Trabajo*, trabajo que introduce a la Auditoría Administrativa y a la forma de aplicarla sobre una base de preguntas para evaluar las áreas funcionales, ambiente de trabajo y sistemas de información.

En 1988, la oficina de la Contraloría General de los Estados Unidos de Norteamérica prepara las *Normas de Auditoría Gubernamental*, que son revisadas por la Contraloría Mayor de Hacienda (entidad de la Secretaría de Hacienda y Crédito Público), las cuales

contienen los lineamientos generales para la ejecución de auditorías en las oficinas públicas.

Al iniciarse la década de los noventa, la Secretaría de la Contraloría General de la Federación se dio a la tarea de preparar y difundir normas, lineamientos, programas y marcos de actuación para las instituciones, trabajo que, en su situación actual, como Secretaría de Contraloría y Desarrollo Administrativo, continúa ampliando y enriqueciendo.

3.2 Definiciones y aportaciones

3.2.1 Enfoque de William P. Leonard.

3.2.1.1 Concepto

Es el examen comprensivo y constructivo de la estructura de una empresa, de una institución o cualquier otra parte de una organización, en cuanto a los planes y objetivos, sus métodos y controles, su forma de operación y sus facilidades humanas y físicas.

Para Leonard la Auditoría Administrativa pretende ayudar y complementar a la administración en determinadas áreas que requieren economías prácticas mejoradas.

3.2.1.2 Aportaciones

Realización de evaluación de la actuación del personal directivo.

Hace el uso de las técnicas de Auditoría Administrativa como: gráficas de organización, gráficas de proceso, gráficos de distribución de formas, simplificación de trabajo, investigación de operación etc. Vislumbra la Auditoría Administrativa como un servicio de asesoramiento integral a la dirección de la empresa.

Método

1. Estudio preliminar
2. Programa de Auditoría
3. Investigación
4. Análisis interpretación y síntesis
5. Preparación del informe
6. Discusión con los directamente afectados
7. Informe final a la dirección
8. Revisión sistemática

3.2.1.3 Metodología

El propósito y las finalidades de la Auditoría Administrativa, son sacar a la luz irregularidades o deficiencias en cualquiera de los elementos examinados e indicar posibles mejoras en las operaciones. Para Leonard las tareas de la Auditoría Administrativa, se convierte en un servicio especializado prestado a la dirección superior, empleando el método científico, en otras palabras se trata de un servicio de consultoría de dirección perteneciente a la empresa.

Su alcance, puede comprender una determinada función o actividad, departamento o grupo de departamentos, división o grupos de divisiones, o la empresa en su totalidad.

Las áreas de investigación puede incluir una apreciación de los métodos de pronóstico, planeación de entregas, medidas de actuación, costos de ingeniería, de proyectos, estimación y fijación de precios, comunicaciones, equipos electrónicos de procesos de datos y efectividad de los directivos involucrados. Lo anterior implica cubrir en forma integral el área de control interno administrativo u operativo, o sea el campo de la función control de la dirección superior.

Factores a revisar y evaluar:

A) Planes y objetivos: en numerosas empresas la administración no da pleno cumplimiento a los principios de organización. Antes de que el auditor pueda valorar la eficiencia de una operación, es necesario entender los planes y objetivos generales de la empresa.

B) Estructura orgánica: es vital una planeación constante de la estructura orgánica, para satisfacer los objetivos y futuras necesidades de ella. Las deficiencias en la estructura son:

- Estructura mal proyectada para satisfacer necesidades.
- Funciones no coordinadas y controladas.
- Falta de delegación, autoridad y responsabilidad en forma apropiada.
- Personal no calificado en puestos.
- Falta de flexibilidad para situaciones cambiantes

Al revisar el organigrama de una empresa, el auditor debe empezar por cerciorarse de que si refleje o no la situación verdadera, actual de las funciones orgánicas. Debe de analizar cada uno de los elementos para determinar si el organigrama precisa los niveles de autoridad y responsabilidad.

C) Políticas: La política está claramente expuesta. La expresión de una política es necesaria para la toma de decisiones, de producción, finanzas, personal, compras, etc. Al cotejar políticas el auditor tomará en cuenta la totalidad de los elementos que incurren; por ejemplo, la evaluación de políticas de ventas involucrará calidad, diseño, estilo, número y variedad de productos.

D) Sistemas y procedimientos: Debe de tener en cuenta su propósito y su forma de decidir sobre sus méritos respecto a la forma en que sirve la empresa. El sistema y procedimiento debe estar proyectado de tal forma que se obtengan los resultados deseados, debe de determinar lo que se está haciendo, dónde, cuándo, cómo, y por quién; cada uno debe ser estudiado y evaluado.

E) Métodos de control: Son medios por los cuales la dirección obtiene ciertos resultados para conseguir una acción coordinada y hacer que el trabajo se realice en la forma proyectada. Comprendiendo el desempeño real contra lo esperado, trátase de objetivo, norma, etc.

El auditor está obligado a hacer un estudio, tanto de los registros como del procedimiento, y siempre que sea necesario hacer las recomendaciones oportunas para que se hagan revisiones.

F) Formas de operación: Por lo que respecta a la mano de obra hay que determinar la fuerza de trabajo básica para satisfacer calidad, producción, etc.

La operación de toda la empresa implica mantener una adecuada relación entre las diversas funciones, para conseguir resultados. Otros factores son los cambios en la economía del país, ya que afectan la operación.

G) Recursos humanos y materiales: La evaluación comprende una revisión para determinar la capacidad y los puntos débiles y fuertes de la gente. Para ello el auditor llevará el examen de las condiciones de trabajo programas de adiestramiento, evaluación del trabajo, calificación de méritos, desarrollo de actividades.

En la Auditoría Administrativa de personal habrá que atender al sistema que se sigue para llevar registros y a la preparación y trámite de los informes. El auditor buscará causas de descontentos en el empleo, falta de observación en el cumplimiento de políticas y procedimientos, coordinación deficiente, etc., su propósito será revelar cualquier debilidad, error, deficiencia.

3.3 Enfoque de José A. Fernández Arena

3.3.1 Concepto

Es la revisión objetiva, metódica y completa de la satisfacción de los, objetivos institucionales, con base en los niveles jerárquicos de la empresa, en cuanto a su estructura y a la participación individual de los integrantes de la institución.

Este autor propone la realización de la Auditoría Administrativa, combinando un análisis de objetivos, de los recursos y del proceso administrativo.

3.3.2 Metodología

Esquema de estudio

1. – Existencia de objetivos

- a) De servicio. Se enfocan a satisfacer las necesidades de los consumidores, ofreciendo buenos productos o servicios.
- b) Protección de los intereses económicos personales y sociales de los empleados y obreros de la empresa, del gobierno y de la comunidad, logrando la satisfacción mediante buenas relaciones humanas y públicas, esto con respecto al objetivo social.
- c) Económico. Protección de los intereses económicos de la empresa, de los acreedores y sus accionistas, logrando la satisfacción de éste grupo mediante la obtención de utilidades.

2. – La dirección

Función ejercida por los niveles jerárquicos más altos de la empresa, quienes se encargan de revisar los avances de la empresa enfocados en cuantos a los objetivos, basando su análisis en el desarrollo de las políticas.

3. – Los recursos

Persigue la utilización equilibrada de los recursos:

- a) Humanos.
- b) Materiales.
- c) Técnicos.

La integración dota a la estructura de las partes necesarias de acuerdo con el esquema de la organización, por lo que obtendrá buenos empleados, fondos

suficientes y su adecuada inversión y sistemas y procedimientos acordes con la magnitud de la empresa.

Como resultado de la organización, la empresa quedará fragmentada en funciones dentro del concepto general administrativo.

4. – El proceso administrativo.

Los objetivos institucionales y el mecanismo de operación requieren del proceso administrativo, que consistirá en planear, implementar y controlar.

3.4 Enfoque de Víctor Rubio Ragazzoni

3.4.1 Concepto y aportaciones

Proporciona una opinión profesional, obtenida, como resultado de estudio e investigación y evaluaciones de carácter profesional, quedando bajo la responsabilidad de la persona que tenga la autoridad de la unidad u organización social, la ejecución de las sugerencias que se proporcionen; entendiéndose que la responsabilidad no se puede delegar y es inherente a la función de la dirección. Puede implantar y ejecutar sus recomendaciones con previa autorización del responsable de la función.

La función de la Auditoría Administrativa consiste en el análisis y dictamen de las actividades que lleva a cabo una unidad administrativa, para verificar que se ajusten a los objetivos y políticas establecidas, así como comprobar la utilización racional de los recursos técnicos, materiales y financieros, y el aprovechamiento del personal operacional y evaluar las medidas de control.

3.4.2 Metodología

1ª. Etapa

Planeación. Es la función donde se define el desarrollo secuencial de las actividades encaminadas dentro de los programas, así como la determinación del tiempo requerido para el desarrollo de cada una de sus etapas.

La finalidad consiste en poder prever, anticipadamente a la acción, todos aquellos factores que se requieran, y que por ausencia de estos no limiten el curso de acción a seguir en pro de los objetivos esperados.

En la planeación el auditor debe considerar:

- Características particulares de la empresa
- Finalidad de los trabajos.

- Secuencia de su desarrollo (seguir un ordenamiento).
- Estimación de tiempos para el desarrollo de cada trabajo.
- Determinación del personal que intervenga (selección del personal apto)
- Determinación de los recursos que se utilizarán (se ajustarán las características del objetivo, a la capacidad y experiencia del personal que le auxiliará)
- Determinación de medios materiales.
- Determinación del apoyo que el organismo social proporcionará

Para la elaboración de los programas específicos de Auditoría, antes de iniciar la auditoría es necesario, recopilar y recabar información, de los manuales de organización y operación, catalogo de formas, proyectos, fundamento legal, planos, y programas, informes, etc., con el objeto de contar con elementos que sirven de base para la elaboración de programas específicos de revisión.

Concluida la investigación preliminar se procederá a un análisis de la documentación e información recabada, a fin de detectar el área o áreas que puedan representar alguna problemática y necesidad de revisión más detallada. Una vez conocido el problema en origen y magnitud a estudiar y los objetivos a cumplir, deberá realizarse un programa específico.

Los auditores deben tener capacidad en:

- Diseños de sistemas y procedimientos
- Análisis de puestos
- Organización y métodos
- Manejo y control de proyectos
- Técnicas de investigación

Antes de iniciar la Auditoría, se diseñarán los instrumentos necesarios que facilitan el desarrollo del estudio. Ejemplo: Cuestionarios, guías de entrevista, listas checables etc.

2^a. Etapa

Aquí comienza la Auditoría puesto que se inicia con la ejecución de los programas para obtener la información necesaria de las áreas sujetas al estudio.

Iniciando la presentación del examen se debe de hacer especial énfasis en los fines y aspectos constructivos de la Auditoría, indicando el resultado que arroje la investigación que será en beneficio del área auditada. Es necesario que esta presentación se pregunte al responsable o director de la unidad, si tiene interés en que alguna área en particular sea revisada con mayor profundidad o detalle y porqué.

En esta etapa es donde se aplican las técnicas de investigación, para la capacitación y análisis de la información, documentación, formas y aspectos operativos a saber.

Técnicas de investigación:

A) Encuestas:

a) Cuestionarios (respuesta cerrada, abierta o selectiva; general de Auditoría; funcional; análisis de condiciones de trabajo, de procedimientos y de formas).

b) Entrevistas (estilos y técnicos)

B) Interpretación de documentos:

a) Verificación (comprobar físicamente si se cumplen los objetivos, políticas y procedimientos).

b) Observación.

3ª. Etapa, Evaluación de Resultados y recomendaciones.

Terminada la investigación, el auditor procederá a formular conclusiones, en el siguiente orden:

- Concentración de los datos obtenidos en la investigación, que le fueron proporcionados, a fin de estudiarlos y proceder a su clasificación.
 - Clasificación de los datos, para que le sirvan de guía para formular conclusiones.

 - Evaluación de resultados.

 - Determinación de la solución, o sea, la conclusión a que ha llegado el auditor, después de evaluado los resultados de las investigaciones que efectuó. Una buena solución será obtenida observando los siguientes puntos:
 - Utilidad que reporta al cliente.
 - Sujeción a los objetivos y políticas establecidas.
 - Requerimientos del cliente.
 - Recursos disponibles.
 - Costo de ejecución
 - Riesgos probables
 - Posibilidad de Control.
-

- Tiempo de realización.
- Desarrollo de la empresa.
- Condiciones legales.
- Derechos

La evaluación deberá comprender las funciones de la administración:

- A) Planeación (objetivos y políticas; planes y programas)
- B) Organización (estructura orgánica; funciones; niveles de autoridad y responsabilidad, manual de organización).
- C) Integración (elemento humano, recursos materiales, técnicos y financieros).
- D) Dirección (supervisión, comunicación, coordinación y delegación).
- E) Control (medidas de control que aseguren los resultados, registros y controles operativos; sistemas y procedimientos administrativos y manual de operación).

4ª. Etapa: Presentación

3.5 Enfoque de Enrique Benjamín Franklin

3.5.1 Concepto

Podemos definir a la Auditoría Administrativa como el examen integral o parcial de una organización con el propósito de precisar su nivel de desempeño y oportunidades de mejora.

3.5.2 Objetivos

Entre los objetivos prioritarios para instrumentarla de manera consistente tenemos los siguientes:

De control.- Destinados a orientar los esfuerzos en su aplicación y poder evaluar el comportamiento organizacional en relación con estándares preestablecidos.

De productividad.- Encauzan las acciones para optimizar el aprovechamiento de los recursos de acuerdo con la dinámica administrativa instituida por la organización.

De organización.- Determinan que su curso apoye la definición de la estructura, competencia, funciones y procesos a través del manejo efectivo de la delegación de autoridad y el trabajo en equipo.

De servicio.- Representan la manera en que se puede constatar que la organización está inmersa en un proceso que la vincula cuantitativa y cualitativamente con las expectativas y satisfacción de sus clientes.

De calidad.- Disponen que tienda a elevar los niveles de actuación de la organización en todos sus contenidos y ámbitos, para que produzca bienes y servicios altamente competitivos.

De cambio.- La transforman en un instrumento que hace más permeable y receptiva a la organización.

De aprendizaje.- Permiten que se transforme en un mecanismo de aprendizaje institucional para que la organización pueda asimilar sus experiencias y las capitalice para convertirlas en oportunidades de mejora.

De toma de decisiones.- Traducen su puesta en práctica y resultados en un sólido instrumento de soporte al proceso de gestión de la organización.

ALCANCE

Por lo que se refiere a su área de influencia, comprende su estructura, niveles, relaciones y formas de actuación. Esta connotación incluye aspectos tales como:

- Naturaleza jurídica
- Criterios de funcionamiento
- Estilo de administración
- Proceso administrativo
- Sector de actividad
- Ámbito de operación
- Número de empleados
- Relaciones de coordinación
- Desarrollo tecnológico
- Sistemas de comunicación e información
- Nivel de desempeño
- Trato a clientes (internos y externos)
- Entorno
- Productos y/o servicios
- Sistemas de calidad

CAMPO DE APLICACIÓN

En cuanto a su campo, la Auditoría Administrativa puede instrumentarse en todo tipo de organización, sea ésta pública, privada o social.

En el Sector Público se emplea en función de la figura jurídica, atribuciones, ámbito de operación, nivel de autoridad, relación de coordinación, sistema de trabajo y líneas generales de estrategia. Con base en esos criterios, las instituciones del sector se clasifican en:

1. Dependencia del Ejecutivo Federal (Secretaría de Estado)
2. Entidad Paraestatal
3. Organismos Autónomos
4. Gobiernos de los Estados (Entidades Federativas)
5. Comisiones Intersecretariales
6. Mecanismos Especiales

En el Sector Privado se utiliza tomando en cuenta la figura jurídica, objeto, tipo de estructura, elementos de coordinación y relación comercial de las empresas, sobre la base de las siguientes características:

1. Tamaño de la empresa
2. Sector de actividad
3. Naturaleza de sus operaciones

En lo relativo al tamaño, convencionalmente se las clasifica en:

1. Microempresa
2. Empresa pequeña
3. Empresa mediana
4. Empresa grande

Lo correspondiente al sector de actividad se refiere al ramo específico de la empresa, el cual puede quedar enmarcado básicamente en:

- | | |
|-----------------------|-------------------------|
| 1.-Telecomunicaciones | 19. Agrícola |
| 2. Transportes | 20. Pesquera |
| 3. Energía | 21. Química |
| 4. Servicios | 22. Forestal |
| 5. Construcción | 23. Farmacéutica |
| 6. Petroquímica | 24. Alimentos y Bebidas |
| 7. Turismo | 25. Informática |
| 8. Cinematografía | 26. Siderurgia |
| 9. Banca | 27. Publicidad |
| 10. Seguros | 28. Comercio |
| 11. Maquiladora | |
| 12. Electrónica | |
| 13. Automotriz | |
| 14. Editorial | |
| 15. Arte gráfico | |
| 16. Manufactura | |
| 17. Auto partes | |
| 18. Textil | |

En cuanto a la naturaleza de sus operaciones, las empresas pueden agruparse en:

1. Nacionales
2. Internacionales
3. Mixtas

También tomando en cuenta las modalidades de:

1. Exportación
2. Acuerdo de Licencia
3. Contratos de Administración
4. Sociedades en Participación y Alianzas Estratégicas
5. Subsidiarias

En el Sector Social, se aplica considerando dos factores:

1. Tipo de organización
2. Naturaleza de su función

Por su tipo de organización son:

1. Fundaciones
2. Agrupaciones
3. Asociaciones
4. Sociedades
5. Fondos
6. Empresas de solidaridad
7. Programas
8. Proyectos
9. Comisiones
10. Colegios

Por la naturaleza de su función se les ubica en las siguientes áreas:

- | | |
|----------------------------|---------------------------------------|
| 1 Educación | 7 Popular |
| 2 Cultura | 8 Empleo |
| 3 Salud y seguridad social | 9 Alimentación |
| 4 Política | 10 Derechos humanos |
| 5 Obrero | 11 Apoyo a marginados y discapacitado |
| 6 Campesino | |

Podemos concluir, que la aplicación de una Auditoría Administrativa en las organizaciones puede tomar diferentes cursos de acción, dependiendo de su estructura orgánica, objeto, giro, naturaleza de sus productos y servicios, nivel de desarrollo y, en particular, con el grado y forma de delegación de autoridad.

La conjunción de estos factores, tomando en cuenta los aspectos normativos y operativos, las relaciones con el entorno y la ubicación territorial de las áreas y mecanismos de control establecidos, constituyen la base para estructurar una línea de acción capaz de provocar y promover el cambio personal e institucional necesarios para que un estudio de Auditoría se traduzca en un proyecto innovador sólido

- a. Evolución
- b. Clasificación
- c. Tendencias

3.6 L.A. y C.P Jorge Álvarez Anguiano

3.6.1 Concepto

La Auditoría Adiministrativa es un servicio específico que se proporciona a petición concreta de un cliente; es independiente de cualquier otra, realizada por un equipo de profesionales que aceptan la responsabilidad de emitir una opinión y sugerir específicamente acerca de la eficiencia de la administración ejecutando un examen completo y exhaustivo.

3.6.2 Metodología

- Investigación preliminar.
- Entrevistas previas.
- Definición del área a investigar.
- Determinación del tiempo disponible para la Auditoría.
- Selección del personal para realizar la Auditoría.
- Aspectos que el personal deberá considerar para realizar la Auditoría.
- Documentos auxiliares a solicitar.
- Obtención y estudio de la información.
- Evaluación.
 - Ponderación de funciones.
 - Determinación de factores a evaluar.
 - Asignación de puntos a los factores a evaluar.
 - Determinación de grados.
- Informe de Auditoría.
 - Alcance de la Auditoría Administrativa.
 - Apreciación crítica de los aspectos fundamentales.
 - Evaluación de la empresa.
 - Recomendaciones tendientes a la eliminación de errores y al mejoramiento de los aspectos que se consideran pertinentes.
 - Anexos.
 - Discusión de problemas con los supervisores.
 - Control de las recomendaciones.

4. MARCO METODOLÓGICO

4.1 Variables

Variable independiente

- Durante el desarrollo de una auditoría, una vez generados los reportes y recomendaciones por el sistema y el auditor respectivamente, la organización, llevará a cabo las medidas que el comité facultado para ello decida.
- La veracidad de los datos ingresados al sistema dependerá del comité o aquellas personas encargadas de proveer de información al auditor.
- La rapidez con que la auditoría sea realizada dependerá también de la disposición que preste la organización y las personas facultadas para proveer de información a la auditoría.

Variable dependiente

- Facilitará el registro y elaboración de Auditorías Administrativas, tomando en cuenta que aprovechará el conocimiento con que cuentan los usuarios hoy en día para utilizar las herramientas orientadas al uso de Internet (browser).
- Permitirá generar información oportuna y verdadera en cada momento del proceso.
- Permitirá disminuir los errores que los procesos manuales provocan, como pueden ser operaciones mal calculadas como conteos de datos, promedios o cifras que dependan de operaciones matemáticas que el sistema puede realizar en altos volúmenes con rapidez y exactitud.
- Permitirá reducir los tiempos de elaboración de las auditorías, tomando en cuenta que los resúmenes de información se generarán de manera automática una vez que los datos básicos han sido ingresados, y que es posible reutilizar información que ha sido capturada una vez, como catálogos de preguntas, datos de empresas etc.
- Permitirá compartir datos dentro de un equipo de trabajo.
- Permitirá acceder al sistema desde cualquier equipo de cómputo que cuente con un browser o navegador capaz de acceder al servicio vía http, sin necesidad de realizar instalaciones de bases de datos ni archivos ejecutables en cada equipo que se utilice para acceder a los datos de las auditorías.
- Permitirá a los desarrolladores agregar nuevos módulos, reportes y funcionalidades con facilidad, sin necesidad de recompilar un solo código, habilitándose en el momento en que el desarrollador considere que puede ser utilizado.
- Permitirá a las empresas adecuar los catálogos, módulos, reportes y funcionalidades para su tamaño y giro específico.

4.2 Variables de control

El registro de las auditorías actualmente se lleva a cabo con herramientas manuales (que pueden provocar equivocaciones en la información, lentitud, e impide facilitar la comparación de los datos con el paso del tiempo) o mediante un sistema poco amigable para los usuarios que no aprovecha los beneficios de los sistemas operativos con escritorios gráficos que permiten compartir los datos con otros usuarios mediante envío o acceso a aplicaciones remotas o bien migrar datos entre aplicaciones para darles otro tratamiento.

El desaprovechamiento de las nuevas tecnologías en el desarrollo de las Auditorías Administrativas, entorpece la agilidad con que éstas puedan llevarse a cabo para brindar un servicio óptimo y eficiente a las organizaciones.

4.3 Hipótesis definitiva

El desarrollo de un sistema de auditoría que proporcione la posibilidad de registrar distintas auditorías con el paso del tiempo a una o varias empresas, facilitará la generación de información histórica que permita observar el desarrollo productivo de la organización, del mismo modo aminorará los costos para la organización por concepto de desarrollo de la misma, y permitirá al auditor. Registrar con rapidez los resultados que sus auditorías generen, así como disminuir los errores que los procesos manuales puedan generar.

4.4 Definición del Universo

El universo considerado para la investigación se compone de personas y empresas involucradas en el desarrollo de auditorías administrativas, dichas personas u organizaciones deben de utilizar un método de Auditoría Administrativa similar al propuesto por el L.A y CP Jorge Álvarez Anguiano.

Por la limitación de recursos (tiempo, económicos e información disponible), no me es posible hacer una cuantificación próxima a la realidad, razón por la que la muestra se determinará como se explica en el siguiente punto.

4.5 Determinación de la muestra

Debido a que no pretendo hacer una investigación de prueba hipotética plena, el tipo de muestra seleccionada de acuerdo al tipo de investigación, corresponde a una muestra no probabilística por juicio.

La opinión de las personas dedicadas al desarrollo de actividades de auditoría, o bien que su actividad laboral les capacite para tener conocimiento acerca de la Auditoría Administrativa, así como los conocimientos adquiridos en su profesión a aquellos que estudiaron la carrera en la FCA y cursaron la materia de Auditoría Administrativa, será lo que me permitirá concluir con un criterio para aprobar o no la relación hipotética.

4.6 Definición del método de investigación

El método de investigación que se aplicará para la probanza de la hipótesis del trabajo de investigación, son las encuestas y el desarrollo de un sistema de Auditoría Administrativa. A través de las encuestas se busca conocer las opiniones y puntos de vistas de profesionales en el tema. El desarrollo de un Sistema es la aportación que hace este trabajo, en el se explicarán las propuestas que se hacen para mejorar y agilizar la realización de auditorías administrativas.

4.7 Costo de la investigación

Tipo	Descripción	Parcial	Total
Recursos Humanos	Desarrollador	\$ 220,000	
<i>Total Recursos Humanos</i>			\$ 220,000
Bienes Inmuebles	Renta local	\$ 20,000	
<i>Total Bienes Inmuebles</i>			\$ 20,000
Bienes Muebles	Equipo de cómputo	\$ 12,000	
	Impresora	\$ 4,500	
<i>Total Bienes Muebles</i>			\$ 16,500
Directos	Papel	\$ 600	
	Cartuchos para impresora	\$ 1,700	
<i>Total Directos</i>			\$ 2,300
Indirectos	Teléfono	\$ 2,300	
	Luz	\$ 3,000	
	Internet	\$ 3,600	
	Copias	\$ 300	
	Otros	\$ 500	
<i>Total Indirectos</i>			\$ 9,700
<i>Total</i>			\$ 268,500

4.8 Colaboradores y apoyos

Se recibió apoyo e información acerca del proceso de Auditoría Administrativa, así como un sistema realizado anteriormente por parte del L.A y C.P. Jorge Álvarez Anguiano.

4.9 Construcción del cuestionario

Pregunta	Justificación	Respuesta esperada
1. Considera usted que sería útil para el auditor un sistema en línea de Auditoría Administrativa.	Ver que tan importante es disponer de un sistema de Auditoría Administrativa desde cualquier punto de la organización	<i>Sí</i>
2. ¿Cree Ud. que un sistema de Auditoría Administrativa disminuiría los errores manuales al registrar la información?	Saber como es considerado el registro correcto de las auditorías administrativas dentro de la organización	<i>Sí, (porque permite automatizar parte del proceso y realiza la validación de los datos.)</i>
3. Por el motivo de que realizar las auditorías requiere una inversión de tiempo considerable, ¿considera que el tiempo se vería reducido al ser estas automatizadas?	Verificar la importancia que tiene el tiempo de desarrollo de una auditoría.	<i>Sí, (porque los auditores cobran dependiendo del tiempo que requiera la auditoría y una disminución de tiempos es una disminución de costos.)</i>
4. ¿Cree que un sistema de auditoría facilitaría el realizar auditorías periódicamente en una organización?	Mostrar la importancia de la periodicidad con que deben realizarse las auditorías.	<i>Sí</i>
5. ¿Cree que un sistema de auditoría ayudaría a comparar los datos de auditorías anteriores con los actuales?	Conocer la importancia de guardar datos históricos en el sistema	<i>Sí</i>

<p>6. ¿Cree que un mismo sistema de auditoría pueda ser utilizado en varios tipos de organización?</p>	<p>Verificar que se considere la posibilidad de utilización del sistema en diferentes tipos de empresa.</p>	<p><i>Si, (siempre y cuando este realizado con la flexibilidad necesaria para ello.)</i></p>
<p>7. ¿Considera importante que existan supervisores de la información que ingresa el auditor?</p>	<p>Verificar que se tomen en cuenta las posibilidades de realizar control de calidad de manera interna al auditor.</p>	<p><i>Si, (es importante verificar que el auditor realiza sus actividades apegado a la información que le provee la organización.)</i></p>
<p>8. ¿Considera importante que se registre documentación acerca de la fuente de los datos ingresados?</p>	<p>Evaluar la importancia de respaldar la fuente de datos</p>	<p><i>Si, (es de gran importancia conocer los documentos que respaldan la auditoría.)</i></p>
<p>9. ¿Qué ventajas encuentra Ud. en la utilización de un sistema de Auditoría Administrativa?</p>	<p>Conocer las ventajas que aporta la utilización del sistema</p>	<p><i>Ayuda a disminuir los errores en los procesos manuales. Agilizar las actividades de registro de datos. Comparar la productividad de las áreas en diferentes lapsos de tiempo Homogeneizar los reportes de auditoría.</i></p>
<p>10. ¿Para quién considera útil la existencia de un sistema de Auditoría Administrativa?</p>	<p>Conocer los posibles usuarios del sistema desarrollado.</p>	<p><i>Los auditores administrativos. Los dueños de negocios interesados en conocer la productividad de su organización. Los alumnos de la materia de Auditoría Administrativa Las universidades que den esta materia. Todos aquellos que se interesen en los sistemas de evaluación de la productividad.</i></p>

4.10 Cuestionario Piloto

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CONTADURÍA Y
ADMINISTRACIÓN

CUESTIONARIO

Nombre: _____

OBJETIVO: Obtener una opinión profesional para el tema de tesis "Automatización de Auditorías Administrativas" aplicado por Cynthia Gabriela Pardo Orozco.

INSTRUCCIONES: Responda las siguientes preguntas.

1. ¿ Considera usted que sería útil para el auditor un sistema en línea de Auditoría Administrativa.?

() Sí

() No

2. ¿ Cree Ud. Que un sistema de Auditoría Administrativa disminuiría los errores manuales al registrar la información?

() Sí

() No

3. ¿ Por el motivo de que realizar las auditorías requiere una inversión de tiempo considerable, ¿considera que el tiempo se vería reducido al ser estas automatizadas?

() Sí

() No

4. ¿ Cree que es útil realizar auditorías periódicamente con el paso del tiempo en una organización?

() Sí

() No

5. ¿ Cree que es importante comparar los datos de auditorías anteriores con los actuales?

() Sí

() No

6. ¿Cree que un mismo sistema de auditoría pueda ser utilizado en varios tipos de organización?

() Sí

() No

7. ¿Considera importante que existan supervisores de la información que ingresa el auditor?

() Sí

() No

8. ¿Considera importante que se registre documentación acerca de la fuente de los datos ingresados?

() Sí

() No

9. ¿Qué ventajas encuentra ud, en la utilización de un sistema de Auditoría Administrativa?

10. ¿Para quién considera útil la existencia de un sistema de Auditoría Administrativa?

4.11 Cuestionario Definitivo

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE CONTADURÍA Y
ADMINISTRACIÓN**

CUESTIONARIO

Nombre: _____

OBJETIVO: Obtener una opinión profesional para el tema de tesis "Automatización de Auditorías Administrativas" aplicado por Cynthia Gabriela Pardo Orozco.

INSTRUCCIONES: Responda las siguientes preguntas.

1 ¿ Cree Ud. Que un sistema de Auditoría Administrativa disminuiría los errores manuales al registrar la información?

() Sí () No

2 Por el motivo de que realizar las auditorías requiere una inversión de tiempo considerable, ¿considera que el tiempo se vería reducido al ser estas automatizadas?

() Sí () No

3 ¿ Cree que es útil realizar auditorías periódicamente con el paso del tiempo en una organización?

() Sí () No

4 ¿ Cree que es importante comparar los datos de auditorías anteriores con los actuales?

() Sí () No

5 ¿ Cree que un mismo sistema de auditoría pueda ser utilizado en varios tipos de organización?

() Sí () No

6 ¿Considera importante que existan supervisores de la información que ingresa el auditor?

() Sí

() No

7 Considera importante que se registre documentación acerca de la fuente de los datos ingresados

() Sí

() No

8 ¿Qué ventajas encuentra ud, en la utilización de un sistema de Auditoría Administrativa?

() Sí

() No

9 ¿Para quién considera útil la existencia de un sistema de Auditoría Administrativa?

4.12 Realización de investigación

Las personas entrevistadas que contestaron el cuestionario son:

Rodrigo Tablas Hernández
Director de Ventas ITC

Carmen Zavaleta
Director de Relaciones Públicas
Café Juquila SA de CV

L.A. Marisol Oropeza
Becaria en Dirección general de empleo y capacitación
Gobierno del DF

Verena Gama Ugalde
Técnico Académico DCAA-UNAM.

C.P. Claudia Arizmendi Jiménez
Jefe de depto. de Crédito y Cobranza

Guillermo González Garrido
Jefe depto de Mensajería
PNUMA(United Nations Enviroment Program) - ONU

1. ¿ Cree Ud. Que un sistema de Auditoría Administrativa disminuiría los errores manuales al registrar la información					
<i>Rodrigo Tablas Hernández</i>	<i>Carmen Zavaleta</i>	<i>Marisol Oropeza</i>	<i>Claudia Arizmendi Jiménez</i>	<i>Verena Gama Ugalde</i>	<i>Guillermo González Garrido</i>
Sí	Sí	Sí	Sí	Sí	Sí
Conclusión: Las personas entrevistadas aceptaron que un sistema automático de captura disminuiría no solamente los errores manuales, debido a que no existiría la necesidad de transportar papelería, que en ocasiones genera pérdida de documentos. Con esto se acepta la idea de que una auditoría realizada de manera automática tendrá beneficios palpables para aquellos que la llevan a cabo.					
2. ¿ Por el motivo de que realizar las auditorías requiere una inversión de tiempo considerable, ¿ considera que el tiempo se vería reducido al ser estas automatizadas?					
<i>Rodrigo Tablas Hernández</i>	<i>Carmen Zavaleta</i>	<i>Marisol Oropeza</i>	<i>Claudia Arizmendi Jiménez</i>	<i>Verena Gama Ugalde</i>	<i>Guillermo González Garrido</i>
Sí	Sí	Sí	Sí	Sí	Sí
Conclusión: Las personas entrevistadas aceptaron que el tiempo de realización de una auditoría sería menor, con ello se concluye que existen elementos beneficios a nivel de costos que apoyan la realización de un sistema de computo que automatice las auditorías administrativas.					
3. ¿ Cree que un sistema de auditoría facilitaría el realizar auditorías periódicamente con el paso del tiempo en una organización?					
<i>Rodrigo Tablas Hernández</i>	<i>Carmen Zavaleta</i>	<i>Marisol Oropeza</i>	<i>Claudia Arizmendi Jiménez</i>	<i>Verena Gama Ugalde</i>	<i>Guillermo González Garrido</i>
Sí	Sí	Sí	Sí	Sí	Sí
Conclusión: Se afirma por parte de los entrevistados que conviene a la organización la realización de periódicas evaluaciones apoyando la posibilidad de realizarlas con más facilidad teniendo un sistema de auditoría.					
4. ¿ Cree que un sistema de auditoría ayudaría a comparar los datos de auditorías anteriores con los actuales?					
<i>Rodrigo Tablas Hernández</i>	<i>Carmen Zavaleta</i>	<i>Marisol Oropeza</i>	<i>Claudia Arizmendi Jiménez</i>	<i>Verena Gama Ugalde</i>	<i>Guillermo González Garrido</i>
Es igual	Sí	Sí	Sí	Sí	Sí
Conclusión: La mayoría opina que la captura de los datos de una auditoría en un sistema realizado para ello ayudaría a comparar datos de distintos periodos a través del tiempo.					
5.- ¿ Cree que un mismo sistema de auditoría pueda ser utilizado en varios tipos de organización?					

<i>Rodrigo Tablas Hernández</i>	<i>Carmen Zavaleta</i>	<i>Marisol Oropeza</i>	<i>Claudia Arizmendi Jiménez</i>	<i>Verena Gama Ugalde</i>	<i>Guillermo González Garrido</i>
Sí	Sí	Sí	No	No	Sí
<p>Conclusión: En este punto, los entrevistados creen en su mayoría que los tiempos de implementación efectivamente se reducen. Al momento en que el programador recibe durante el comienzo de la programación los modelos, diagramas y documentación de un sistema, puede agilizar el tiempo de desarrollo, todo ello dependiendo del conocimiento que tenga para interpretar los elementos del UML en código Java.</p>					
<p>6.-¿ Considera importante que existan supervisores de la información que ingresa el auditor?</p>					
<i>Rodrigo Tablas Hernández</i>	<i>Carmen Zavaleta</i>	<i>Marisol Oropeza</i>	<i>Claudia Arizmendi Jiménez</i>	<i>Verena Gama Ugalde</i>	<i>Guillermo González Garrido</i>
Sí	Sí	Sí	No	No	Sí
<p>Conclusión: La mayoría de los entrevistados concuerdan en que es importante a nivel de control la existencia de supervisores que verifiquen el seguimiento de las auditorías.</p>					
<p>7.-¿ Considera importante que se registre documentación acerca de la fuente de los datos ingresados?</p>					
<i>Rodrigo Tablas Hernández</i>	<i>Carmen Zavaleta</i>	<i>Marisol Oropeza</i>	<i>Claudia Arizmendi Jiménez</i>	<i>Verena Gama Ugalde</i>	<i>Guillermo González Garrido</i>
Sí	Sí	Sí	No	No	Sí
<p>Conclusión: La mayoría de los entrevistados concuerdan en que es importante tener registros de la documentación y las entrevistas que se realizaron que respalden los resultados de las auditorías.</p>					

8.-¿ Qué ventajas encuentra ud, en la utilización de un sistema de Auditoría Administrativa?

Rodrigo Tablas Hernández	Carmen Zavaleta	Marisol Oropeza	Claudia Arizmendi Jiménez	Verena Gama Ugalde	Guillermo González Garrido
Agilidad en la realización Menos errores Generación de mejores informes	Rapidez en la obtención de resultados	Facilidad para ver los resultados de la auditoría Facilidad para hacer informes	Mayor control acerca de los datos Menor incidencia de errores manuales	La posibilidad de generar estadísticas con la información El uso de tecnologías que faciliten la realización de las auditorías.	Detección rápida de errores y mejor manejo de la información Rapidez en la realización de las auditorías.

Conclusión: Los entrevistados concuerdan en que las auditorías serían mas rapidas y generarían información con mas facilidad al existir un sistema encargado del registro de las auditorías.

9.-¿ Para quién considera útil la existencia de un sistema de Auditoría Administrativa?

Rodrigo Tablas Hernández	Carmen Zavaleta	Marisol Oropeza	Claudia Arizmendi Jiménez	Verena Gama Ugalde	Guillermo González Garrido
Para las empresas que realizan estos procedimientos	Para cualquier empresa que quiera realizar auditorías y tenga personal capacitado para ello.	Para los estudiantes de la materia en la FCA Para los auditores administrativos en las empresas	Para una empresa de consultoría	Para todos aquellos que tengan un acercamiento con la Auditoría Administrativa, conozcan el proceso y sepan usar un equipo de computo	Para estudiantes de la materia, auditores y las empresas que quieran usarlo.

Conclusión: Las respuestas de las personas fueron variadas pero concuerdan en puntos como el de empresas que se dediquen a la consultoría en Auditoría Administrativa, estudiantes de la materia, y empresas que tengan el soporte necesario y deseen llevarlas a cabo en su organización.

4.12.1 Análisis del desarrollo

A continuación se desglosan los requerimientos del sistema a nivel lógico

4.12.1.1 Requerimientos de información

4.12.1.1.1 Los actores:

- El sistema requiere la participación de un comité representante de la organización encargado de proporcionar información requerida para realizar la auditoría.
- La participación de un auditor encargado de registrar información que respalde las evaluaciones de las organizaciones.
- Por otro lado requiere la existencia de empresas y subfunciones a ser evaluadas por el auditor.
- Requiere también de un supervisor , encargado de verificar las operaciones y registros realizados por el auditor.
- Y por ultimo de un administrador de sistema que instale el sistema, verifique el buen funcionamiento del mismo y realice el mantenimiento continuo que la aplicación requiera.

4.12.1.1.2 La información:

El sistema requiere como datos para registrar a un nuevo auditor:

Nombre del auditor

El sistema requiere como datos para registrar a una nueva empresa:

Nombre de la empresa

Dirección: calle, número, delegación, código postal, colonia, estado, municipio

Nombre del contacto

Puesto del contacto

Teléfono

Fax

Mail

Giro de la empresa

Fecha de fundación

Clasificación, grupal, general y detallada

El sistema requiere como datos para registrar a una nueva auditoría:

Nombre del auditor.

Fecha de inicio de la auditoría.

Fecha de cierre de la auditoría.

Horas dedicadas a la auditoría.

Nombre de la empresa a auditar.

Nombre del contacto a quien se dirige la auditoría

Comentarios de la auditoría

4.12.1.2 Los objetivos del sistema Audit-Admin son:

- Prestar un servicio gratuito de prácticas de Auditoría Administrativa para los estudiantes de la materia en la FCA.
- Difundir el uso de la Auditoría Administrativa en la micro y pequeña empresa por medio de una aplicación gratuita migrable a otros sistemas.
- Registrar la información concerniente a una o varias auditorías administrativas realizadas en una o varias entidades distintas.
- Recibir los datos de evaluación para generar reportes y estadísticas acerca de las auditorías.
- Guardar datos históricos con el fin de evaluar el comportamiento de las organizaciones con el paso del tiempo.

4.12.1.3 Lista de los eventos del sistema Audit-Admin son:

4.12.1.3.1 Registrar un Usuario

- Proporcionar datos generales del nuevo auditor
- Elegir un login y un password

4.12.1.3.2 Modificar un Usuario

- Entrar al sistema con login y password de administrador de sistema
- Elegir opción de modificación de usuarios
- Elegir un usuario del listado
- Realizar las modificaciones a los datos del usuario

4.12.1.3.3 Crear una Empresa

- El auditor Proporciona login y password.
- El auditor Accesa al sistema.
- El auditor Abre una nueva empresa.
- El auditor Define los datos de la empresa.

4.12.1.3.4 Editar una Empresa

- El auditor Proporciona login y password.
- El auditor Accesa al sistema.
- El auditor Abre una empresa existente.
- El auditor Define los nuevos datos de la empresa.

4.12.1.3.5 Eliminar una Empresa

- El auditor Proporciona login y password.

- El auditor Accesa al sistema.
- El auditor selecciona la opción de borrado
- El auditor elige la empresa a eliminar.

4.12.1.3.5.0 Crear una auditoría

- El auditor Proporciona login y password.
- El auditor Accesa al sistema.
- El auditor elige una empresa a auditar.
- El auditor Define los datos requeridos para generar una auditoría nueva.

4.12.1.3.5.1 Definir subfunciones a auditar

- El auditor genera la lista de impresión de subfunciones del catalogo.
- El auditor ingresa los datos que el comité le presento para definir las subfunciones a evaluar.
- El auditor ingresa cuales serán las subfunciones a evaluar para la auditoría.

4.12.1.3.5.2 Definir peso de cada subfunción

- El auditor genera la lista de impresión de las subfunciones tomadas para ser evaluadas por el comité.
- El auditor ingresa los pesos que el comité designo.
- El auditor ingresa cuales serán los pesos asignados a la organización con respecto a la información que el comité proporcionó.

4.12.1.3.5.3 Definir factores a evaluar

- El auditor genera la lista de impresión de factores del catalogo.
- El auditor ingresa los datos que el comité le presento para definir los factores a evaluar.
- El auditor ingresa cuales serán los factores a evaluar para la auditoría.

4.12.1.3.5.4 Definir la distribución de los puntos

- El auditor genera la matriz de puntos para impresión.
- El auditor ingresa los datos que el comité le presento para la distribución de los puntos.

4.12.1.3.5.5 Definir preguntas

- El auditor genera la lista de impresión de preguntas a realizar para ser evaluadas por el comité.
- El auditor elige las preguntas que el comité acepto, y agrega las preguntas que el comité considere pertinentes.
- El auditor relaciona las preguntas a realizar con las subfunciones y factores seleccionados.

4.12.1.3.5.6 Evaluar preguntas

- El auditor genera las listas de impresión de preguntas a realizar para ser efectuadas
- El auditor inserta en el sistema los resultados que las preguntas ofrecieron
- El auditor inserta las observaciones a las preguntas realizadas

4.12.1.3.6 Editar los datos generales de una Auditoría

- Ingresar al sistema con su login y password
- Elige la auditoría a editar
- Editar los datos

4.12.1.3.7 Borrar una auditoría

- Ingresar al sistema con su login y password
- Elegir auditoría a borrar

4.12.1.3.8 Generar Reportes

- Ingresar al sistema con su login y password
- Elegir auditoría
- Elegir tipo de reporte a generar
- Generar reporte

4.12.3.9 Operaciones con catálogos

- Elegir la operación de alta, modificación o baja
- Elegir el catalogo a modificar
- Insertar los datos requeridos por la forma presentada

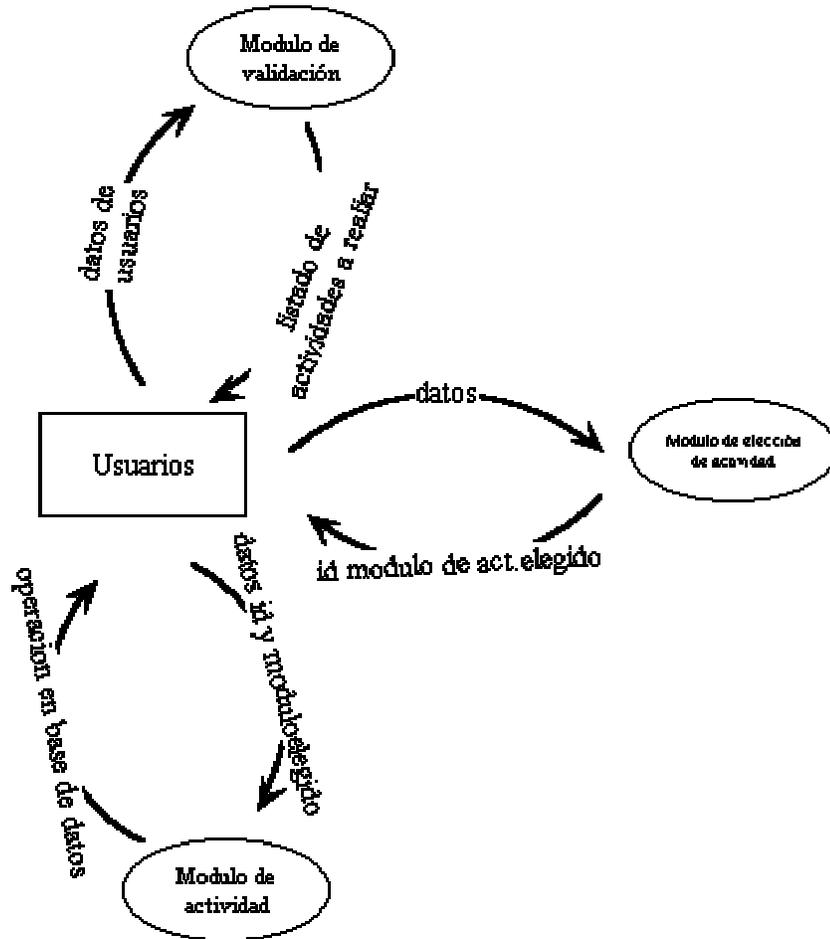
4.12.1.4 Diagrama de contexto

4.12.1.4.1 Breve explicación de los procesos

4.12.1.4.1.1 Usuario

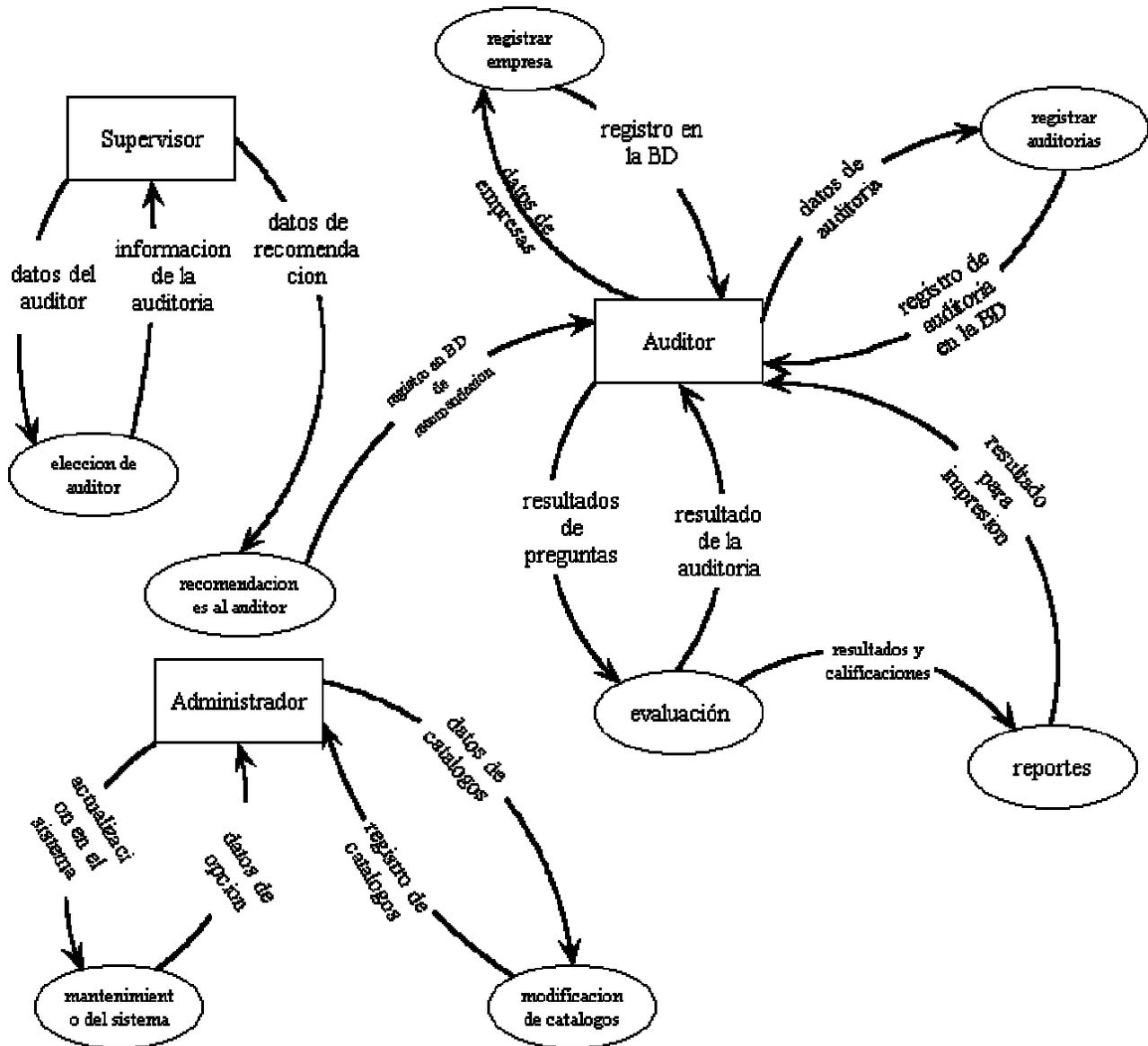
1) El usuario inserta datos de identificación (login y password), el sistema los revisa y entrega de regreso una lista de las actividades que le son permitidas realizar dependiendo del tipo de usuario.

2) El usuario envía al sistema su elección de actividad y el sistema le da como resultado el módulo que realiza la operación deseada.



2) El usuario ingresa los datos que requiera la operación, el sistema los recibe e ingresa en la base de datos.

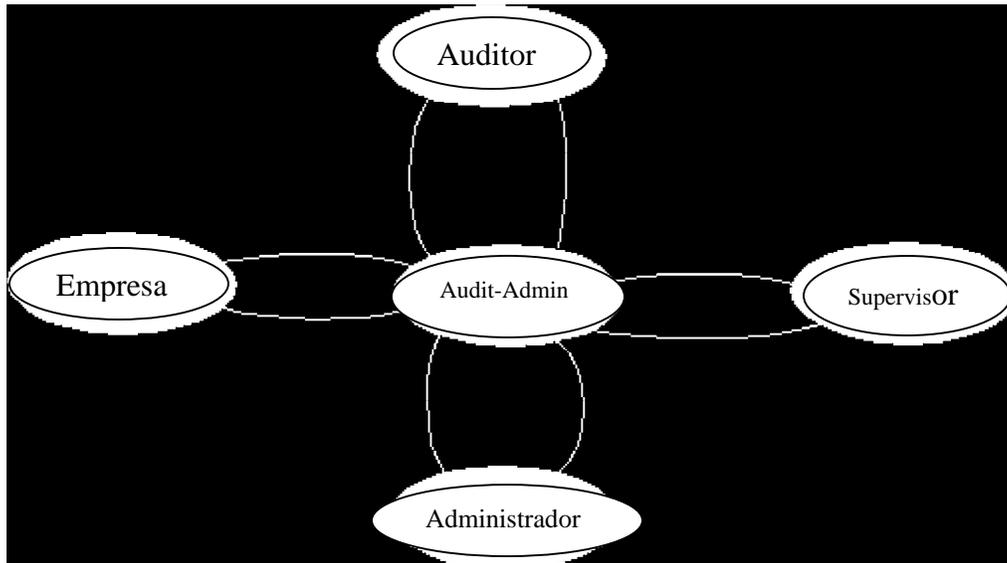
El panorama de cada tipo de usuario se explica a continuación:



4.12.1.4.1.2 Auditor

El auditor inserta los datos recopilados de una auditoría, las observaciones acerca de una empresa y la evaluación realizada, el sistema le devuelve un registro de existencia de la empresa, la auditoría, las actividades realizadas, los reportes y el calculo del resultado de dicha auditoría.

4.12.1.4.1.3 Administrador



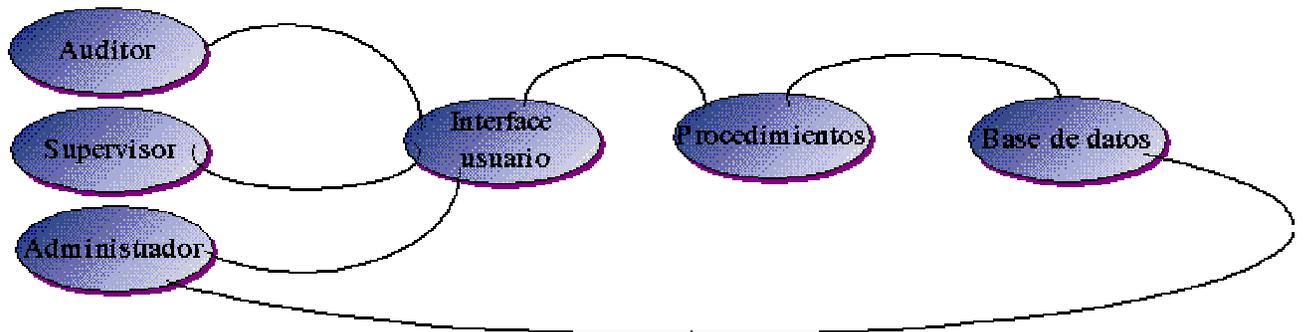
El administrador inserta datos acerca de los nuevos usuarios del sistema o bien los elimina y recibe un aviso de existencia de nuevo registro o bien de que las modificaciones realizadas sobre los usuarios han sido realizadas.

El administrador inserta datos acerca de los catálogos del sistema o bien los elimina y recibe un aviso de existencia de nuevo registro o bien de que las modificaciones realizadas han sido realizadas.

4.12.1.4.1.4 Supervisor

El supervisor accesa a la información del sistema mediante su identificación y los datos del auditor o la auditoría que desea verificar, y el sistema le devuelve la información requerida acerca del auditor o la auditoría.

4.13 Explicación del proceso a desarrollar para realizar una auditoría



Administrador

- Se encarga de instalar el sistema en el equipo destinado para ello.
- Llena los catálogos necesarios para el funcionamiento del sistema, subfunciones, factores, grupos.
- Captura los usuarios que utilizaran el sistema.
- Da mantenimiento a los datos.

Auditor

Fase 1

Registros necesarios

1 Realizar una entrevista preliminar para conocer los objetivos de la Auditoría Administrativa.

2a Debe registrar primeramente una empresa a la cual se aplicará una auditoría, esto es ingresar los datos mínimos de registro de una empresa

2b o bien seleccionar una empresa del catálogo de empresas existentes.

3a Generar una nueva auditoría seleccionando la empresa a que se va a aplicar y llenando datos específicos de la nueva auditoría como son el nombre del auditor, las horas que dedicara al día a la auditoría a realizar, la fecha de inicio de la auditoría y la fecha de fin, el contacto de la organización a quien se dirigen los reportes de lo encontrado.

3b seleccionar una auditoría ya insertada para realizar las siguientes operaciones

Fase 2

Comenzar con la toma de datos de auditoría

4 Imprimir el catálogo de subfunciones y factores para proporcionarlo al comité y que este defina que factores y subfunciones se tomaran para realizar la auditoría, así como recibir nuevos factores y subfunciones necesario que no existan en el catalogo actualmente.

5 Recibir el catálogo con las subfunciones y el orden de importancia que se aplicara a las subfunciones así como el porcentaje que se aplicara a a cada uno de ellos.

6 Registrar en el sistema las subfunciones y su peso dentro de la organización.

7 Registrar en el sistema los factores definidos por el comité.

8 Imprimir la lista de distribución de puntos al comité para definir la distribución de puntos definitiva

9 Registrar los datos del comité

10 Imprimir el catálogo de preguntas por factor subfunción para que el comité defina cuales son las que se utilizaran

11 Ingresar la información que definió el comité.

12 Elegir las preguntas que el comité definió y relacionarlas a los factores y subfunciones anteriormente seleccionados.

13 Imprimir las preguntas y aplicar los cuestionarios.

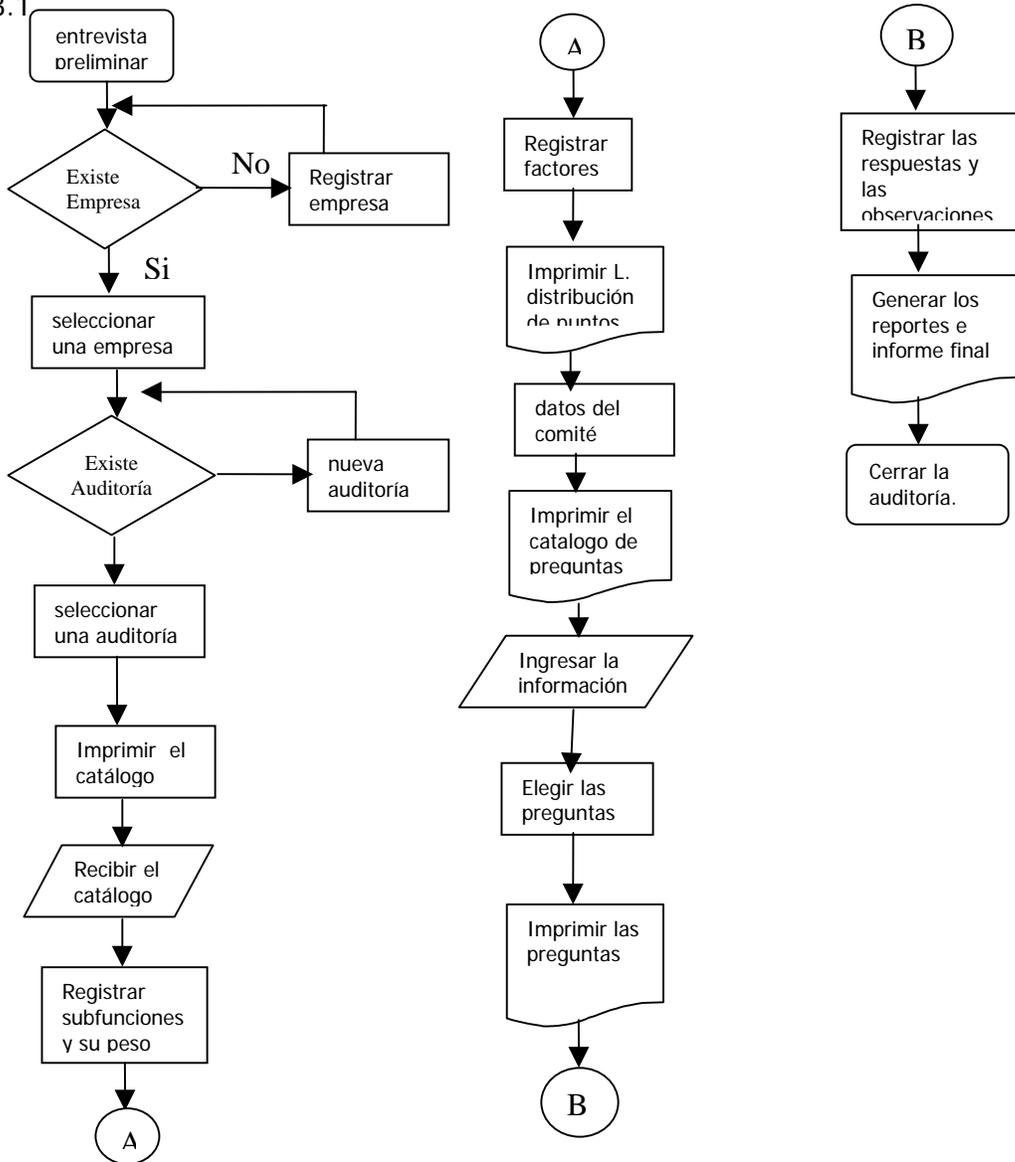
14 Registrar las respuestas y las observaciones correspondientes.

15 Generar los reportes e informe final.

16 Cerrar la auditoría.

El diagrama 14.13.1 lo explica.

14.3.1



4.14 Tratamiento sistematizado de la información

Para llevar a cabo el esquema propuesto se requiere contar con una base de datos, para lo que se propone el siguiente listado de tablas.

Auditoría	cat_factores
cat_preguntas	cat_subfunciones
clasif_detalle	clasif_general
clasif_grupal	Clasificacion
Delegacion	Distrib_puntos
Empresa	Estado
Evaluacion	fac_eva
Giro	Grados
Grupos	Ponderacion_sub
Preguntas	sub_porcentaje
Tipousuario	Usuarios

Formación de las tablas:

Tabla "auditoría"			
Column	Type	Modifiers	
id_empresa	integer	not null	Dato identificador de la empresa
id_auditoria	integer	not null	Dato identificador de la auditoría
id_usuario	integer	not null	Dato identificador del usuario
n_auditor	character(50)		Nombre del auditor
ap_auditor	character(50)		Apellido paterno del auditor
am_auditor	character(50)		Apellido materno del auditor
hr_dedicadas	double precision		Horas dedicadas a la auditoría
Fecha_inicio	date		Fecha de inicio de la auditoría
Fecha_final	date		Fecha final de la auditoría
comentarios	text		Comentario y recomendaciones del auditor.

Tabla "cat_factores"			
Column	Type	Modifiers	
id_factor	integer	not null	Dato identificador del factor
Nombre	character(100)	not null	Nombre o descripción del factor
Primary key: cat_factores_pkey			Llave primaria de la tabla
Tabla "cat_preguntas"			
Column	Type	Modifiers	
id_pregunta	integer		Dato identificador de la pregunta
Descripcion	text		Nombre o descripción de la pregunta
Tabla "cat_subfunciones"			
Column	Type	Modifiers	
id_sub	integer	not null	Dato identificador de la subfunción
nombre	character(100)	not null	Nombre o descripción de la subfunción
Primary key: cat_subfunciones_pkey			Llave primaria de la tabla
Tabla "clasif_detalle"			
Column	Type	Modifiers	
id	character(5)		Identificador de la tabla de catalogo de clasificaciones a detalle
Descripcion	Text		Descripción de el detalle de la clasificación
division	character(1)		Identificador de la tabla de clasificación general
grupo	character(2)		Identificador de la tabla de clasificación general
Tabla "clasif_general"			
Column	Type	Modifiers	
division	character(1)		Identificador de la tabla de clasificación general
Descripcion	character(255)		Descripción de la clasificación general

Tabla "clasif_grupal"			
Column	Type	Modifiers	
grupo	character(2)		Identificador de la tabla de clasificación de grupo
Descripcion	Text		Descripción de la clasificación de grupo.
division	character(1)		Identificador de la tabla de clasificación general
Tabla "clasificacion"			
Column	Type	Modifiers	
id_clas	character(2)	not null	Identificador de la clasificación
nota	character(11)	not null	
Descripcion	character(250)		
Primary key: clasificacion_pkey			
Tabla "delegacion"			
Column	Type	Modifiers	
id_del	integer	not null	Dato Identificador del estado
nombre	character(60)	not null	
Primary key: delegacion_pkey			
Tabla "distrib_puntos"			
Column	Type	Modifiers	
id_auditoria	integer	not null	Dato identificador de la auditoría
id_sub	integer	not null	Dato identificador de la subfunción
id_factor	integer	not null	Dato identificador del factor
puntos	integer		
Primary key: distrib_puntos_pkey			

Tabla "empresa"			
Column	Type	Modifiers	
id_empresa	integer	not null	Dato identificador de la empresa
id_clas	integer	not null	Identificador de la clasificación
id_giro	integer	not null	Dato identificador del giro
id_est	character(3)	not null	Dato Identificador del estado
id_del	integer		Dato Identificador de la delegación
nombre	character(50)		
nombre_emp	character(150)		
telefono	character(21)		
a_paterno	character(50)		
a_materno	character(50)		
fecha_fundacion	Date		
calle	character(50)		
numero	character(4)		
colonia	character(50)		
cp	character(5)		
municipio	character(50)		
clasgen	integer		
clasgrup	integer		
clasdet	integer		
fax	integer		
email	Text		
puesto	character(255)		
Primary key: empresa_pkey			
Tabla "estado"			
Column	Type	Modifiers	
id_est	character(3)	not null	Dato Identificador del estado
nombre	character(21)	not null	
Primary key: estado_pkey			

Tabla "evaluacion"			
Column	Type	Modifiers	
id_grado	integer		
id_auditoria	integer	not null	Dato identificador de la auditoría
id_empresa	integer	not null	Dato identificador de la empresa
id_pregunta	integer	not null	Dato identificador de la pregunta
id_sub	integer	not null	Dato identificador de la subfunción
id_factor	integer	not null	Dato identificador del factor
Observaciones	Text		
Tabla "fac_eva"			
Column	Type	Modifiers	
id_factor	Integer	not null	Dato identificador del factor
id_sub	Integer	not null	Dato identificador de la subfunción
id_empresa	Integer	not null	Dato identificador de la empresa
puntos	Integer		
Tabla "giro"			
Column	Type	Modifiers	
id_giro	character(2)	not null	Dato identificador del giro
descripcion	character(250)		Descripción del giro
Primary key: giro_pkey			
Tabla "grados"			
Column	Type	Modifiers	
id_grado	integer	not null	Dato identificador del grado
apreciacion	Character(9)	not null	Descripción del grado
cump_superior	integer		Porcentaje máximo del grado
cump_inferior	integer		Porcentaje mínimo del grado
Primary key: grados_pkey			Llave primaria de la tabla grado
Tabla "grupos"			
Column	Type	Modifiers	
id_grupo	integer	not null	Dato identificador de la tabla de grupo
descripcion	Character(30)		Descripción del grupo
Estatus	integer		Estado del grupo

Primary key: grupos_pkey			Llave primaria de la tabla
--------------------------	--	--	----------------------------

Tabla "ponderacion_sub"			
Column	Type	Modifiers	
id_sub	integer	not null	Dato identificador de la subfunción
id_auditoria	integer	not null	Dato identificador de la auditoría
porcentaje	double precision		Porcentaje de importancia de la subfunción dentro de la organización a que se aplica la auditoría.

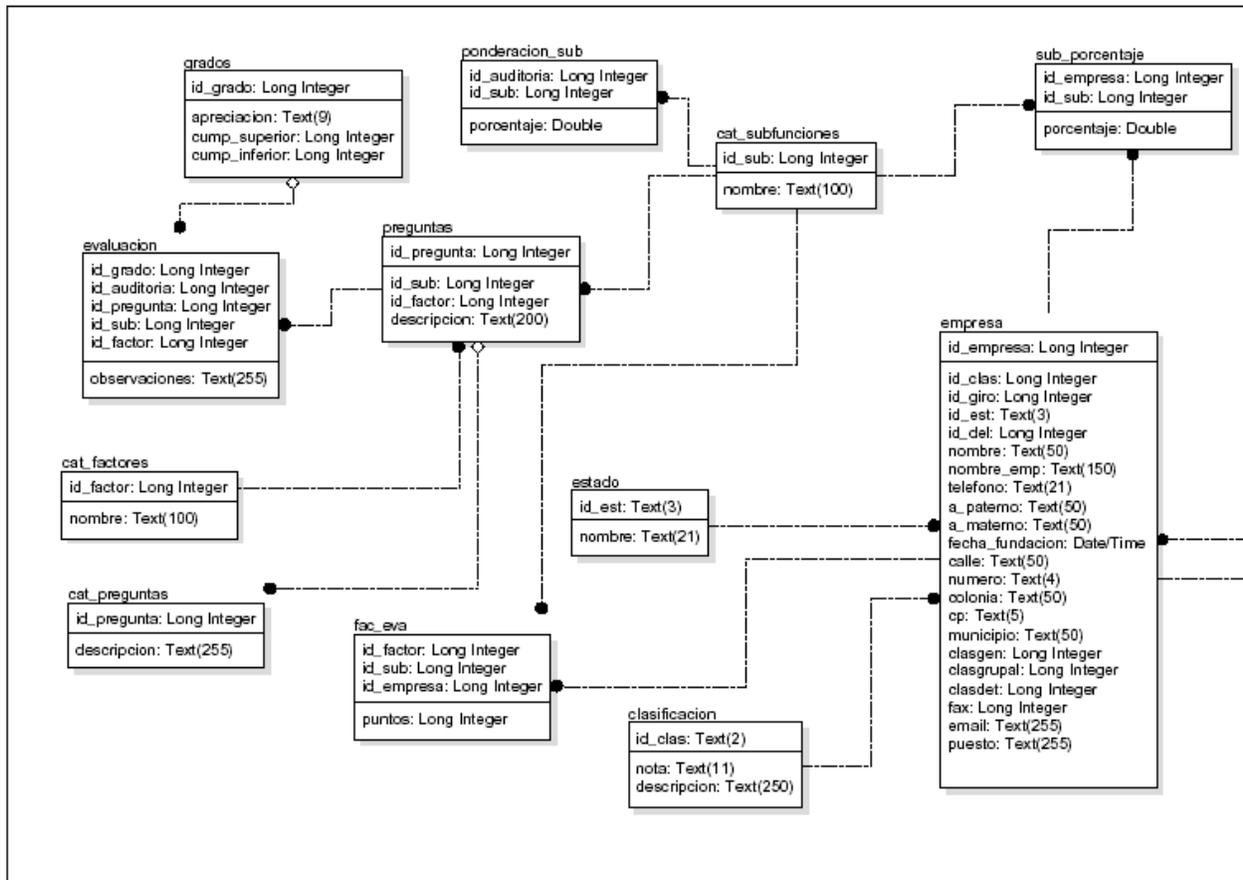
Tabla "preguntas"			
Column	Type	Modifiers	
id_pregunta	integer	not null	Dato identificador de la pregunta
id_sub	integer	not null	Dato identificador de la subfunción
id_factor	integer	not null	Dato identificador del factor
descripcion	character(200)		Descripción de la pregunta.
Primary key: preguntas_pkey			

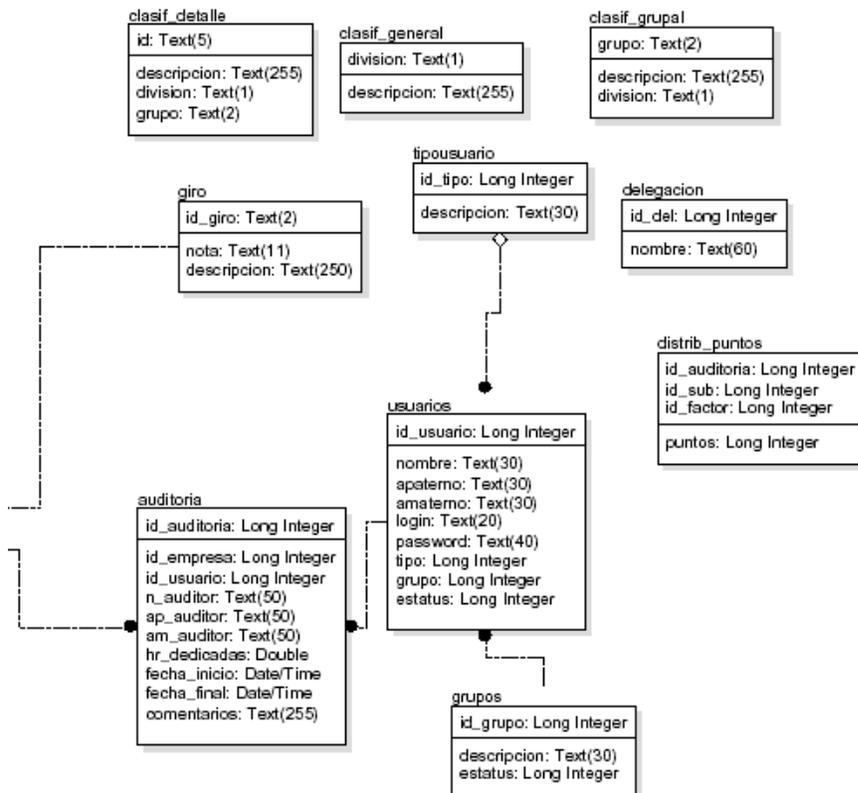
Tabla "sub_porcentaje"			
Column	Type	Modifiers	
id_sub	integer	not null	Dato identificador de la subfunción
id_empresa	integer	not null	Dato identificador de la empresa
porcentaje	double precision		Porcentaje de la subfunción dentro de la empresa a que se aplica la auditoría.

Tabla "tipousuario"			
Column	Type	Modifiers	
id_tipo	integer	not null	Dato identificador del tipo de usuario
descripcion	character(30)		Descripción del tipo de usuario
Primary key: tipousuario_pkey			Llave primaria de la tabla de tipo de usuario

Tabla "usuarios"			
Column	Type	Modifiers	
id_usuario	integer	not null	Dato identificador del usuario
Nombre	character(30)		Nombre del usuario
apaterno	character(30)		Apellido paterno del usuario
Amaterno	character(30)		Apellido materno del usuario
login	character(20)		Login del usuario
password	character(40)		Password del usuario
tipo	Integer		Tipo de usuario
grupo	Integer		Grupo al que pertenece el usuario
Estatus	Integer		Estado del usuario
Primary key: usuarios_pkey			Llave primaria de la tabla de usuarios

A continuación se muestra el diagrama de relaciones de las tablas listadas





El desarrollo del sistema se hará sobre el lenguaje de programación tomando en cuenta que es un lenguaje libre que puede bajarse libre de precio por Internet y esta diseñado con funciones para manejo de texto con facilidad, al mismo tiempo que puede ser utilizado para realizar cgis o bien aplicaciones que pueden distribuirse vía Internet mediante un servicio de http.

La base de datos elegida para el desarrollo es PostgreSQL por ser de libre uso y presentar módulos de acceso de conexión para perl desarrollados para prestar servicio vía Internet.

El código de creación de la base de datos se presenta en la primera parte de anexos.

El código de los scripts de Perl se presenta en la segunda parte de anexos.

Con base en los beneficios que presentan los servicios de Web como:

- Independencia de plataforma, pues los cgis pueden ser llamados desde cualquier máquina con diferentes plataformas que sea capaz de interpretar html y javascript
- Independencia con las actualizaciones pues las actualizaciones se realizan directamente en el servidor que presta el servicio y no es necesario realizar actualizaciones o bien instalar el sistema en cada máquina en que va a ser utilizado.
- Facilidad de acceso, pues simplemente se requiere una computadora con acceso a Internet para que el sistema pueda ser utilizado.

- Distribución de procesos tomando en cuenta que las validaciones se realizan mediante el uso de JavaScript desde la máquina en la que se realizan las peticiones.

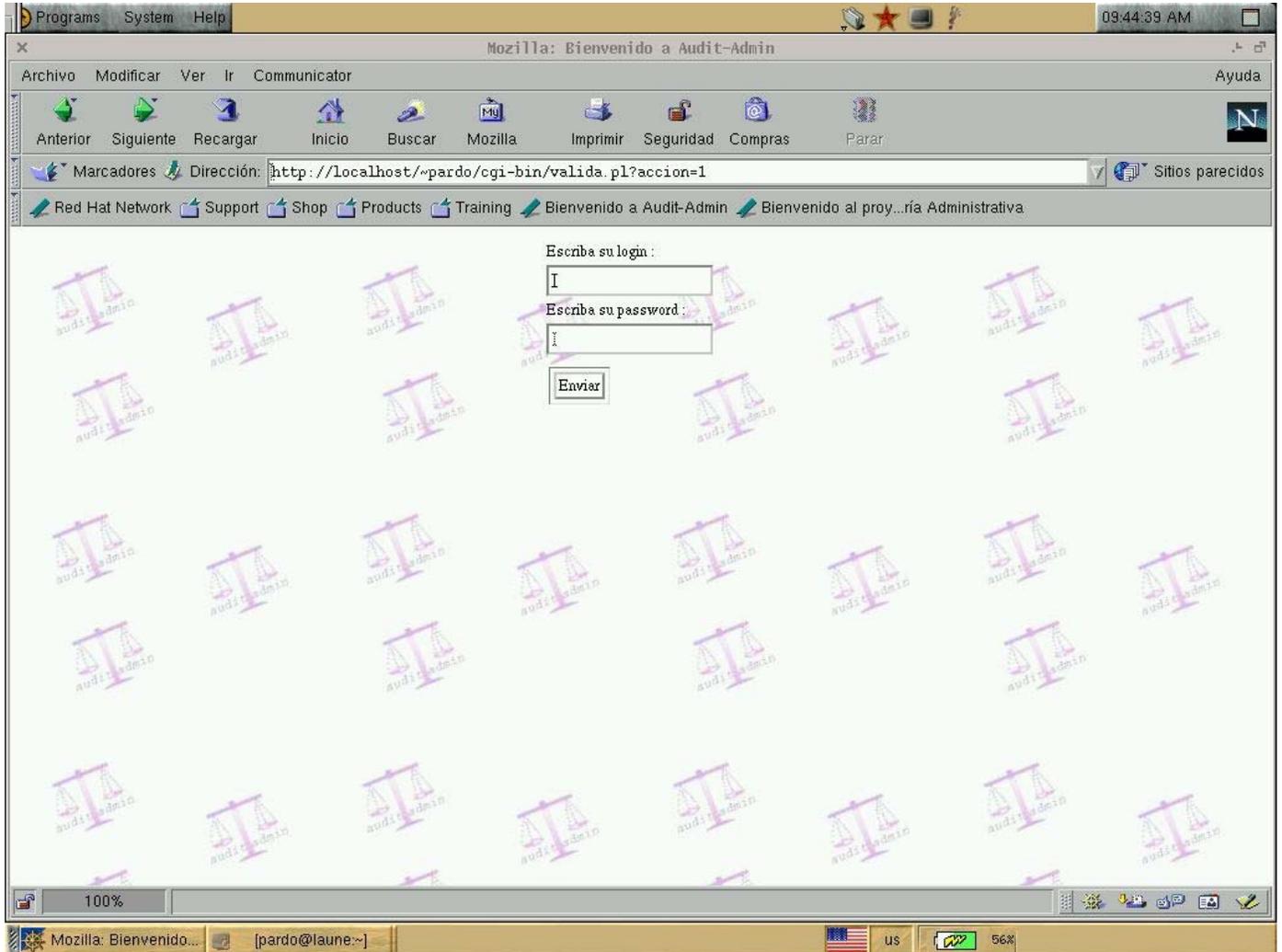
Los módulos que comprende el sistema son:

Inserción de nuevos usuarios	insertUser.pl	Administrador
Modificación de los usuarios	modifUser.pl	Administrador
Cambios a los catálogos	changeCat.pl	Administrador
Borrado de usuarios	delUser.pl	Administrador
Inserción de elementos al catalogo	insertCat.pl	Administrador
Selección de operaciones a catálogos del sistema	opCatalogos.pl	Administrador
Selección de operaciones con usuarios	opUsuarios.pl	Administrador
Trabajar con una auditoría	workAudit.pl	Administrador
Seleccionar una auditoría	choseaudit.pl	Auditor
Definir la distribución de puntos	defineDistrib.pl	Auditor
Definición de los factores	defineFactor.pl	Auditor
Selección de las preguntas	defineQuestion.pl	Auditor
Selección de preguntas para relacionarlas con el catalogo	defineQuestionA.pl	Auditor
Definición de las subfunciones	defineSub.pl	Auditor
Definición de los pesos de subfunciones	defineWeight.pl	Auditor
Evaluación de preguntas	evalQuestion.pl	Auditor
Impresión de listados de catálogos	imprime.pl	Auditor
Inserción de una nueva auditoría	insertAudit.pl	Auditor
Elección de auditorías	opAuditorías.pl	Auditor
Operaciones con empresas para realizar cambios	opEmpChange.pl	Auditor
Borrado de empresas	opEmpDel.pl	Auditor
Insertado de empresas	opEmpInsert.pl	Auditor
Selección de operaciones con empresas	opEmpresas.pl	Auditor

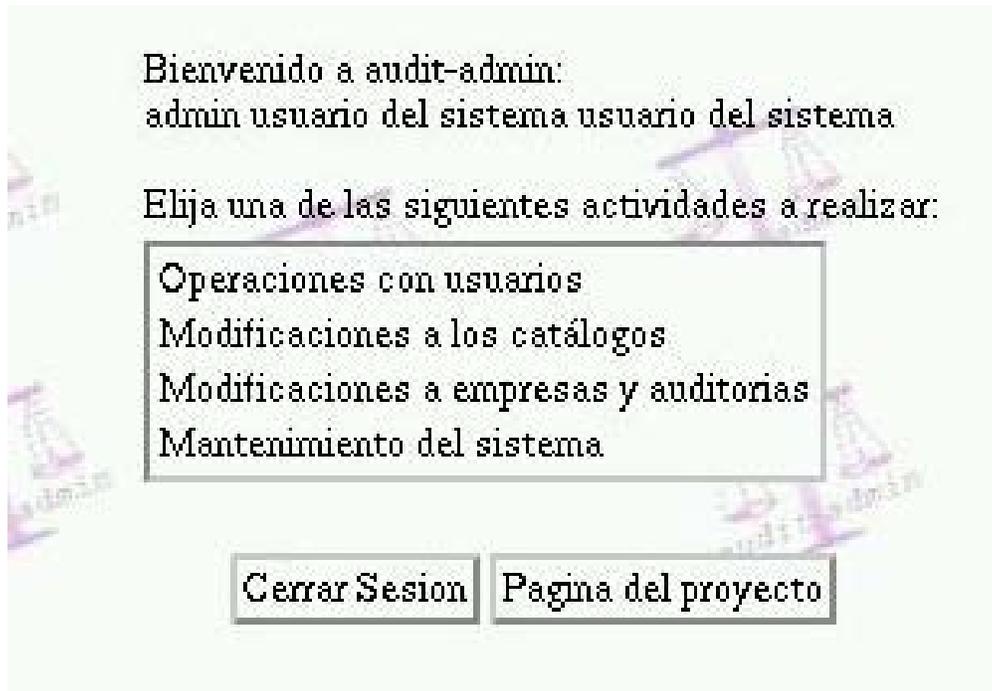
Requerimientos	
Servidor	
CPU (probado en AMD athlon 4) 400 Megahertz	Software Apache Web Server 1.3
Espacio mínimo de 475 MB	PostgreSQL 7.2 Perl 5.6.1
Requerimiento de memoria 128 MB	
Cliente	
Browser Netscape o mozilla 3.7 o superior	(el sistema ha sido probado en netscape 4.7 o mozilla aunque iexplorer 5.0 o superior en teoría puede ser usado)
* El sistema fué desarrollado bajo la versión Linux RedHat 7.3	

4.15 Análisis de los resultados

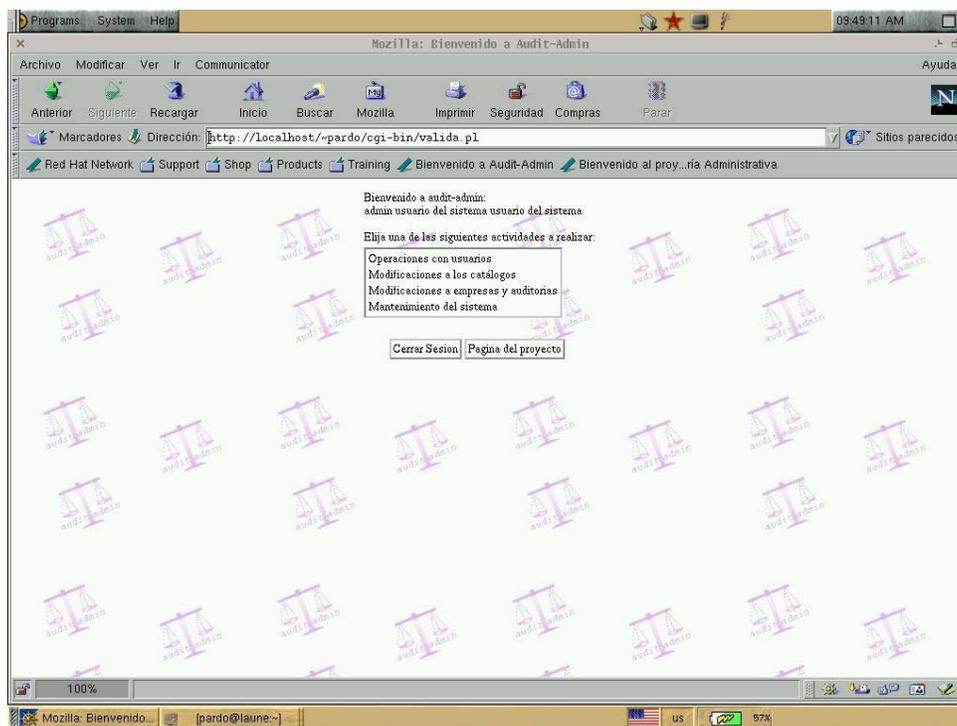
La siguiente imagen muestra la pantalla de ingreso al sistema audit-admin, donde es necesario teclear un nombre de usuario o login y un password.



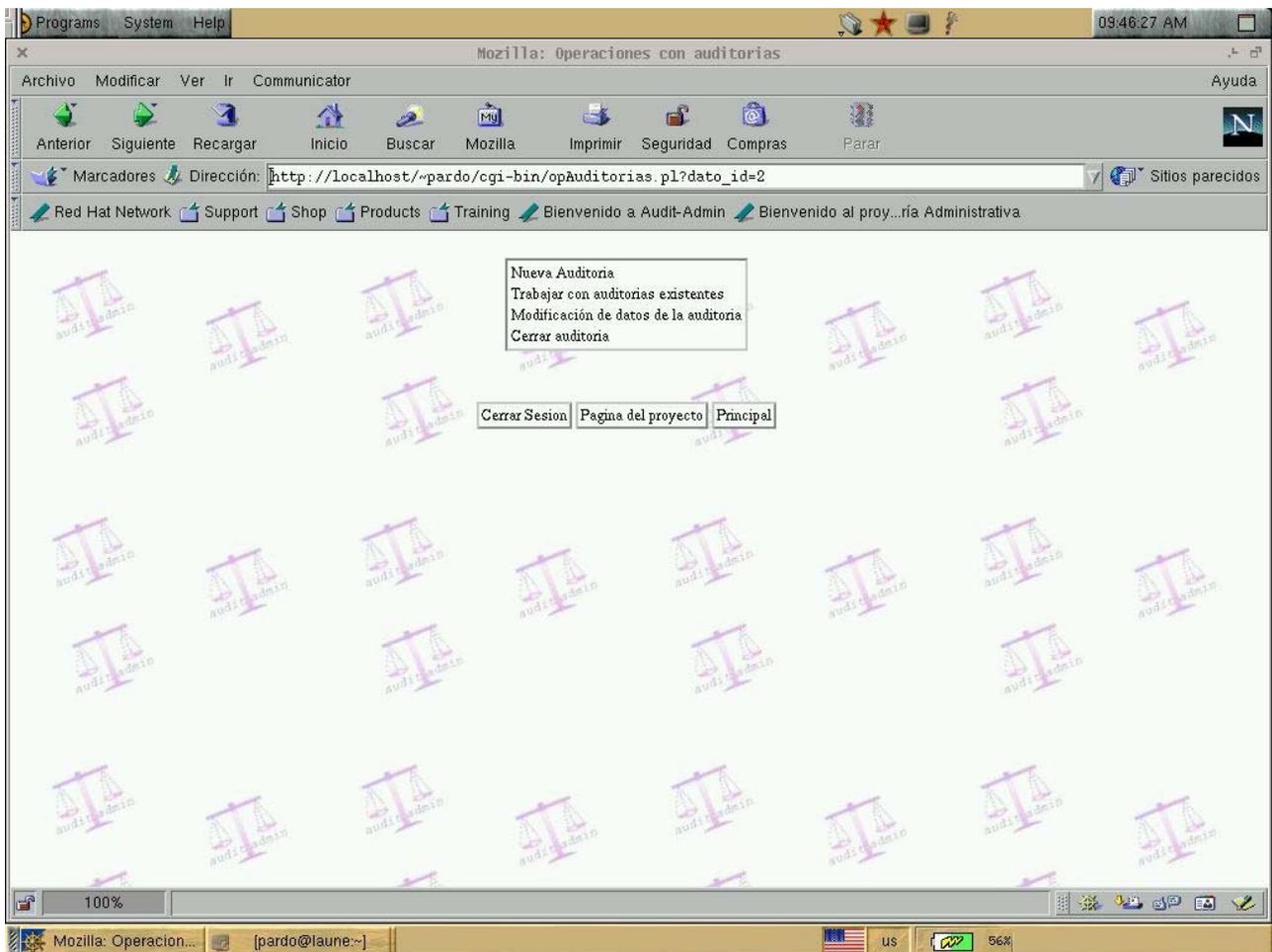
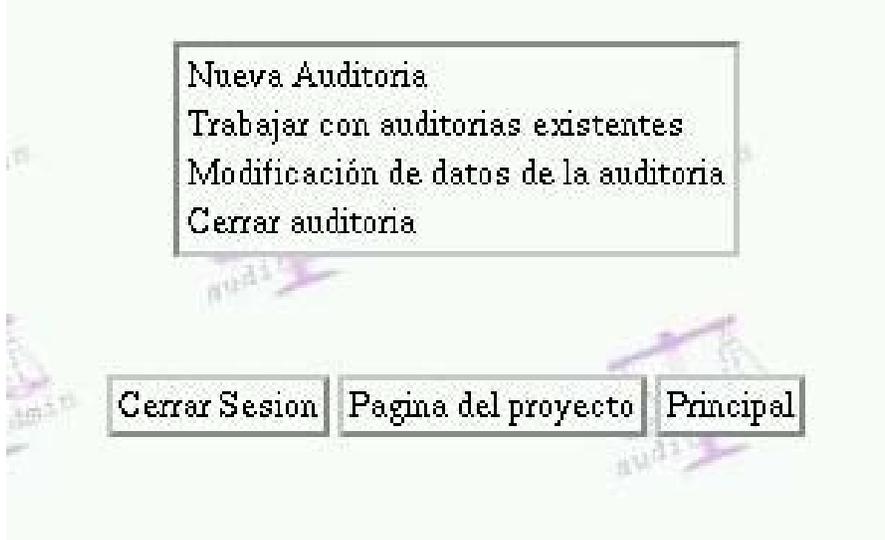
Sí el usuario es de tipo administrador se mostrará la siguiente pantalla de inicio que muestra las operaciones que se pueden realizar, el login y el nombre del usuario que ingreso al sistema.



Al dar click sobre cualquiera de las opciones del listado aparecerá la siguiente pantalla donde se llevará a cabo la actividad.

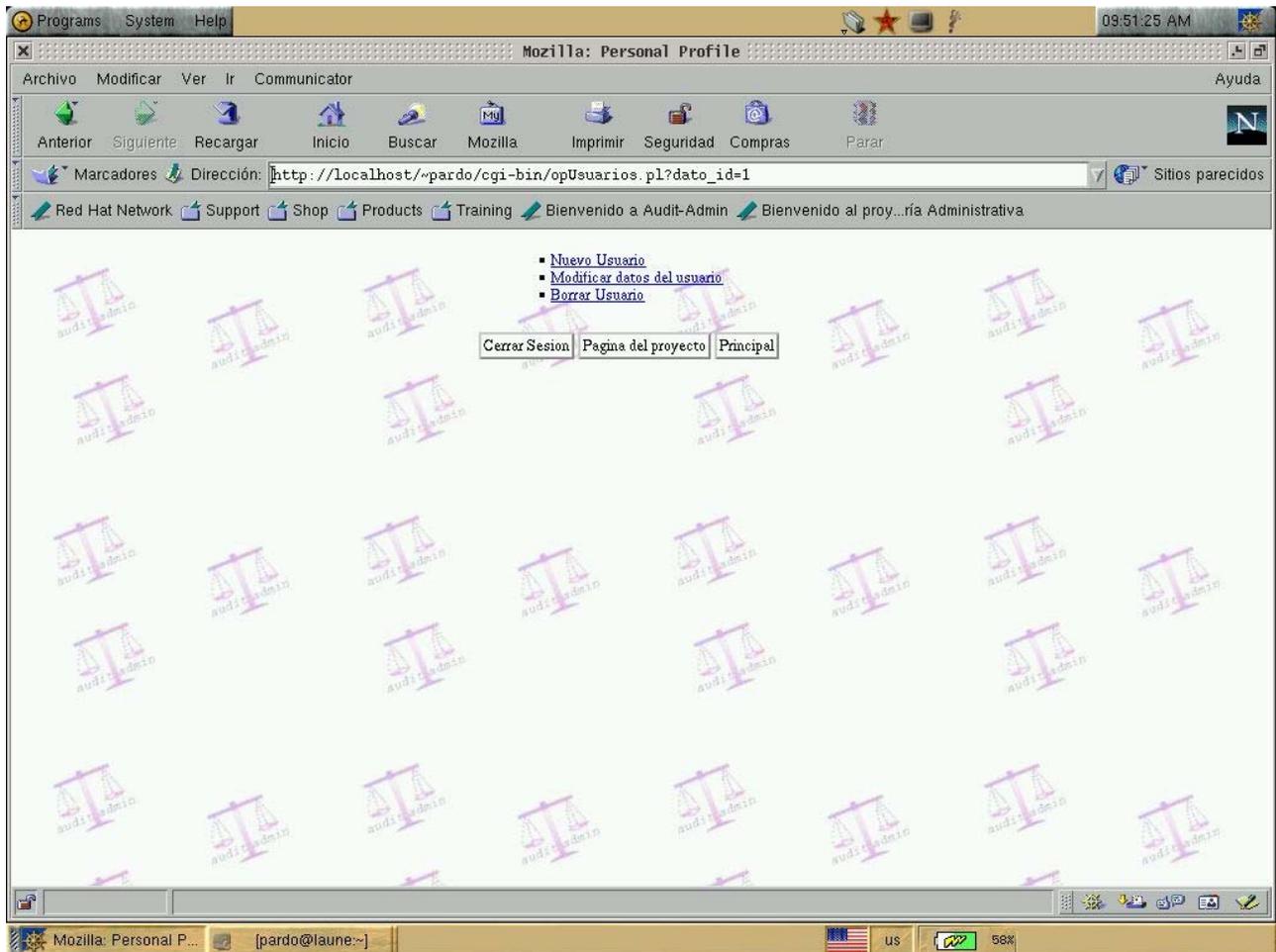
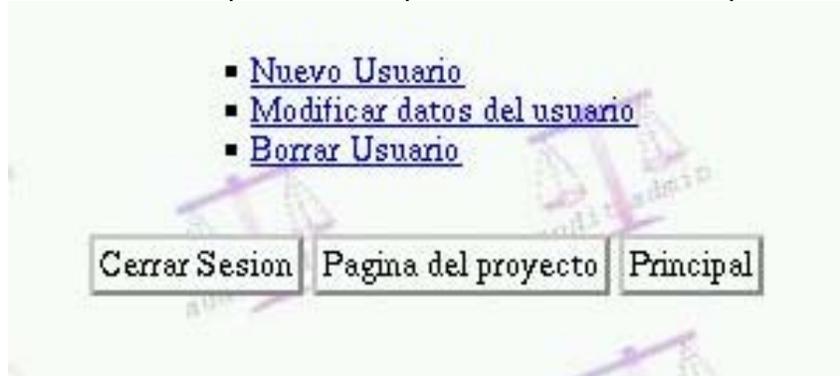


La siguiente pantalla es el listado que aparece cuando ingresa un usuario de tipo auditor.



El desarrollo de las operaciones del supervisor no se llevo a cabo por cuestiones de tiempo, solo se llevo a cabo a manera de documentación. Sin que esto afecte el funcionamiento correcto del sistema.

A continuación se muestra la pantalla de operaciones con usuarios (administrador)



Modulo de inserción de usuarios

Mozilla: Personal Profile

MODULO DE ALTA DE USUARIOS AUDIT-ADMIN

Para dar de alta nuevos usuarios es necesario insertar los siguientes datos:

Nombre del usuario: I

Apellido Paterno: I

Apellido Materno: I

Login: I

Password: I

Seleccione un grupo al que el usuario pertenecerá: 2001

Selecciona el tipo de usuario que se dará de alta: Auditor

Insertar Usuario Cancelar

Modulo de modificación de usuarios, donde es necesario elegir a un usuario del listado y realizar las modificaciones.

Mozilla: Untitled Document

Actualice los campos con los nuevos valores:

Nombre del usuario: Juanjo

Apellido Paterno: Solorzarno

Apellido Materno: Gomez

Login: Juanjo

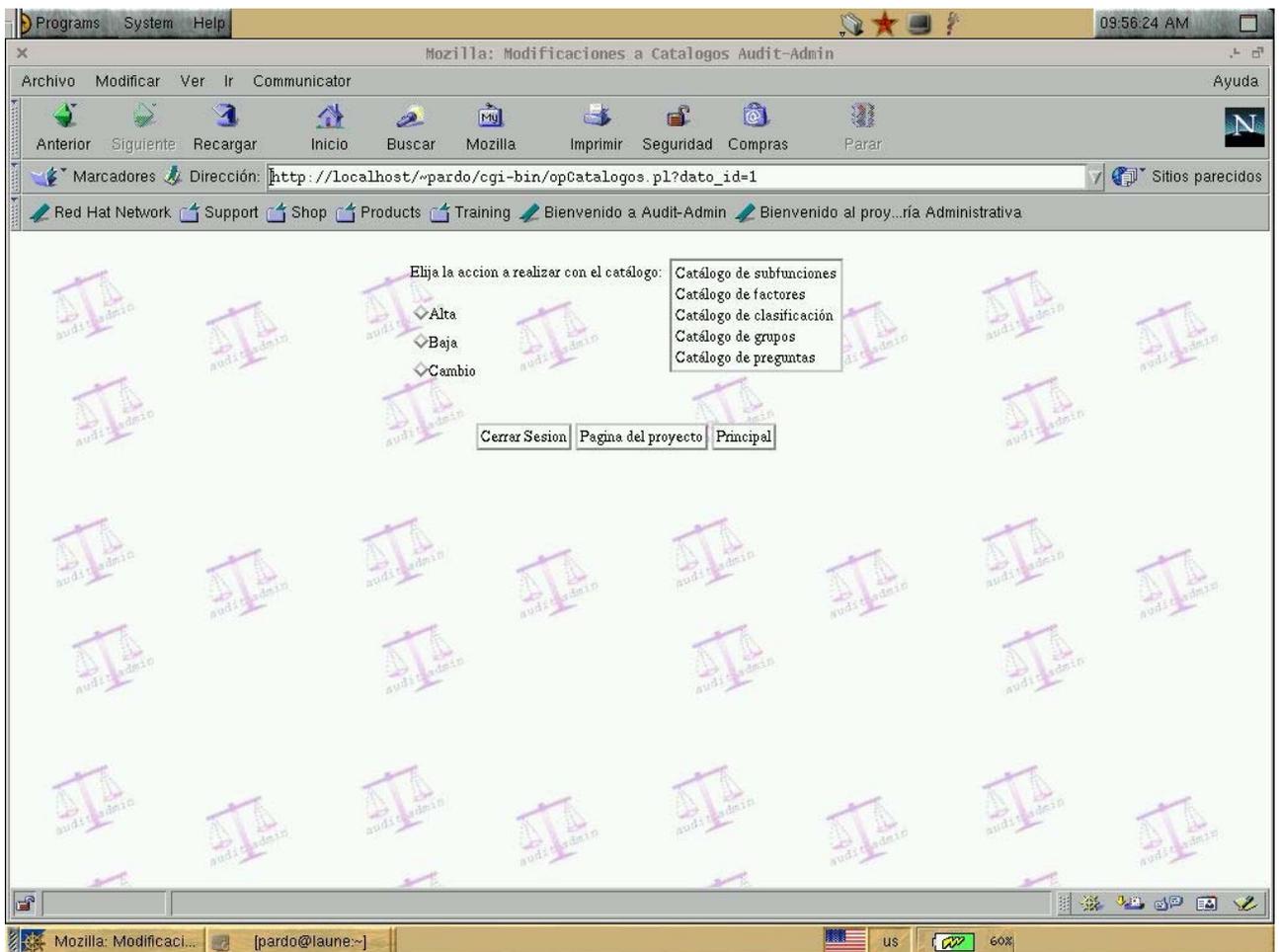
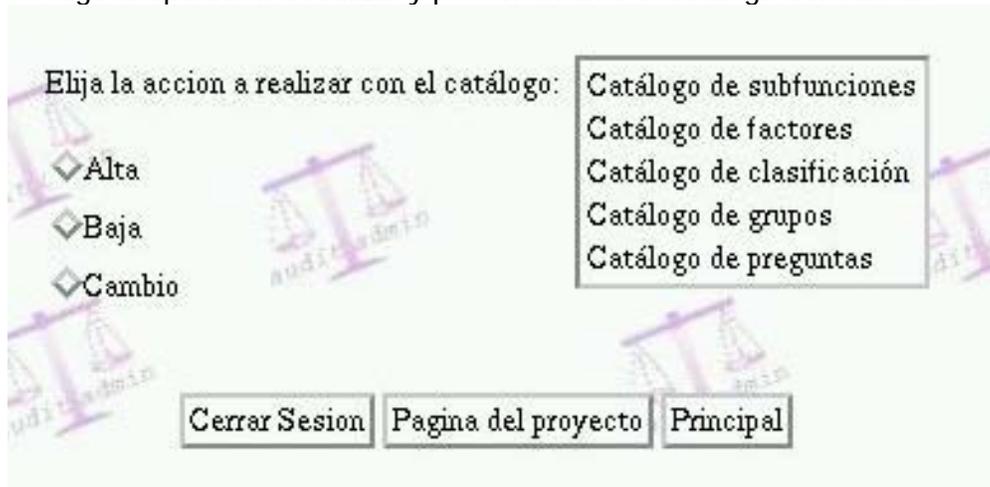
Password: *****

Seleccione un grupo al que el usuario pertenecerá: "2003"

Selecciona el tipo de usuario que se dará de alta: "Auditor"

Actualizar

A continuación se muestra la pantalla de operaciones con catálogos (administrador) donde se elige la operación a realizar y posteriormente el catálogo a modificar.



A continuación se muestran las operaciones con auditorías para realizar por el auditor, donde es necesario seleccionar una operación y posteriormente la auditoría a la que se aplicará.

- MODULO DE OPERACIONES CON AUDITORIAS - AUDIT-ADMIN

Operación a realizar

- ◇ 1. Definir subfunciones a auditar
- ◇ 4. Distribucion de puntos
- ◇ 2. Definir peso de cada subfunción
- ◇ 5. Definir preguntas
- ◇ 3. Definir factores a evaluar
- ◇ 6. Evaluar preguntas

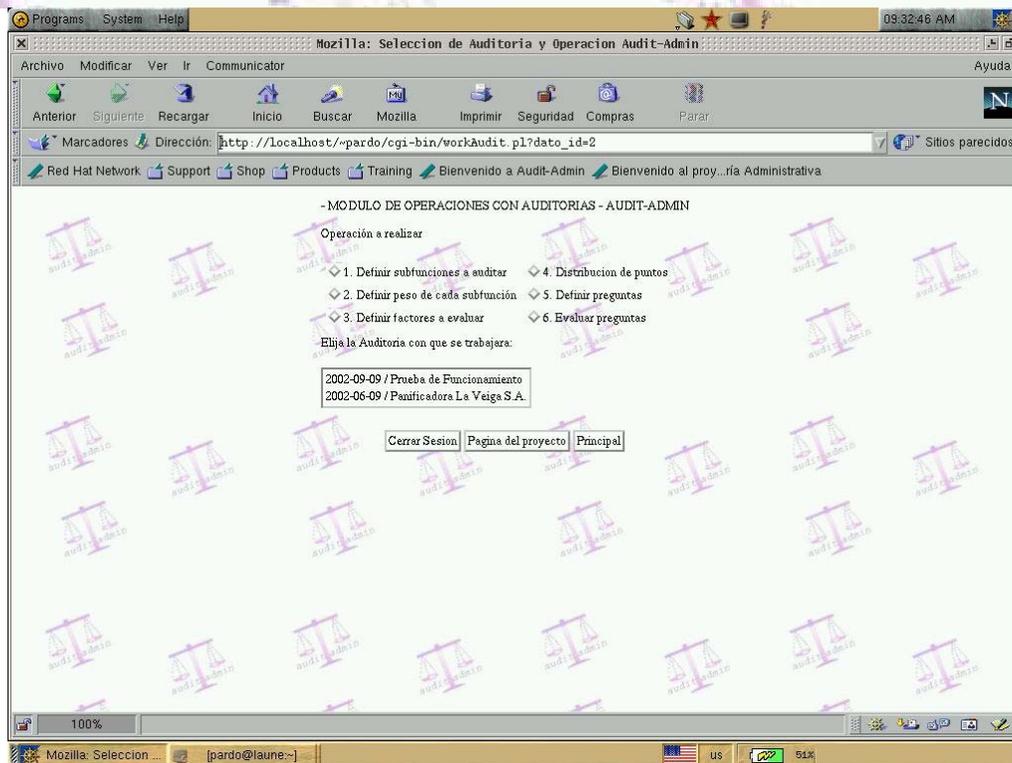
Elija la Auditoria con que se trabajara:

2002-09-09 / Prueba de Funcionamiento
2002-06-09 / Panificadora La Veiga S.A.

Cerrar Sesion

Pagina del proyecto

Principal



A continuación se muestra la pantalla de inserción de empresas (auditor).

MODULO DE INSERCIÓN DE DATOS NUEVAS EMPRESAS AUDIT-ADMIN

Para dar insertar empresas es necesario llenar los siguientes datos:

Nombre de la Empresa

Fecha de fundación "dd/mm/aaaa"

Clasificación General

Clasificación Tamaño

Nombre del Contacto

Apellido Paterno

Apellido Materno

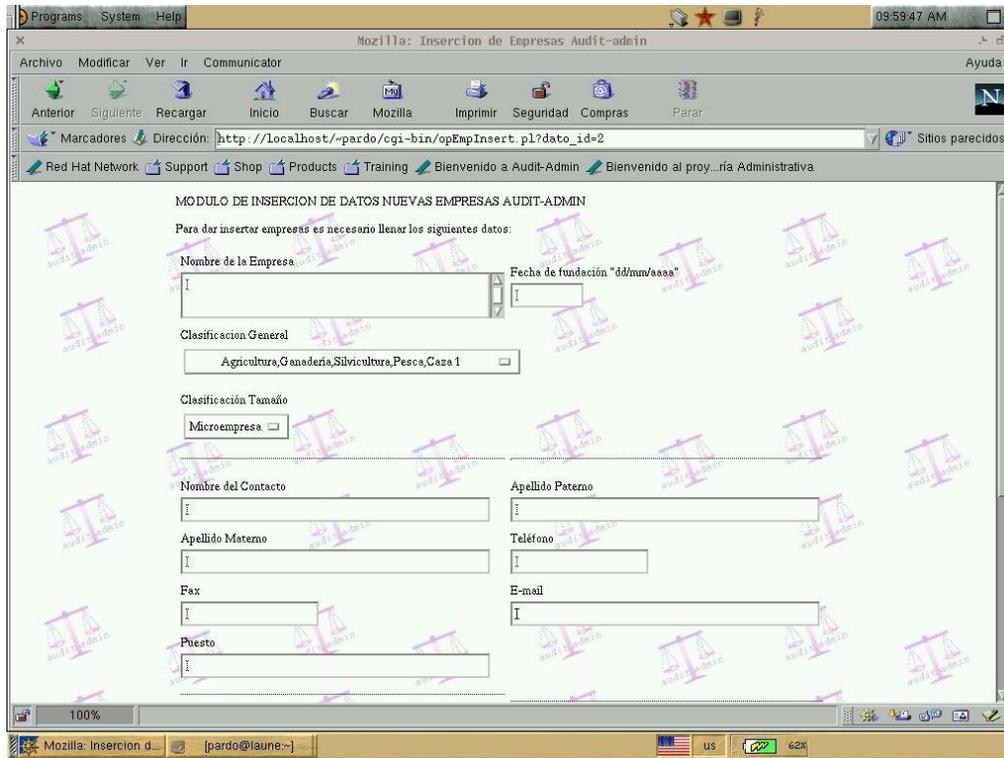
Teléfono

Fax

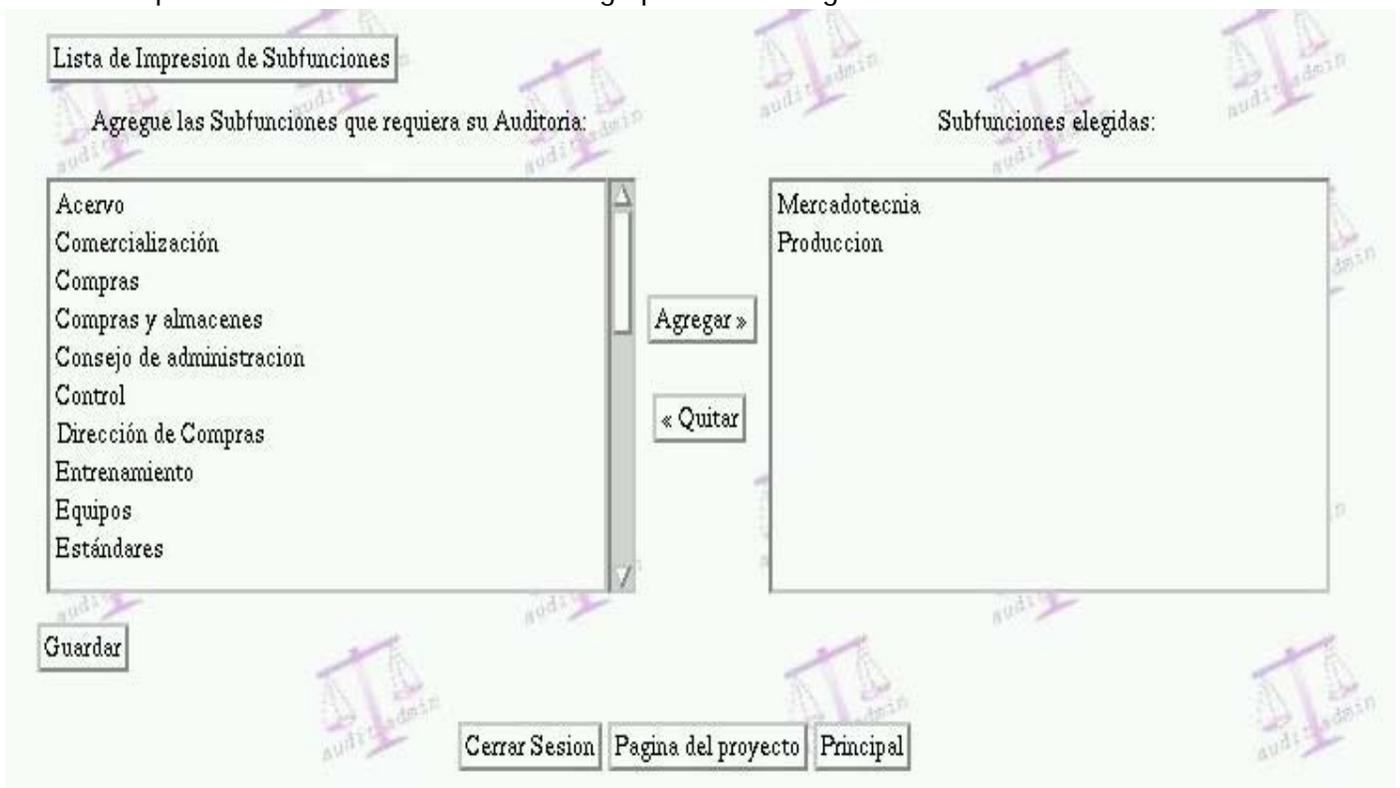
E-mail

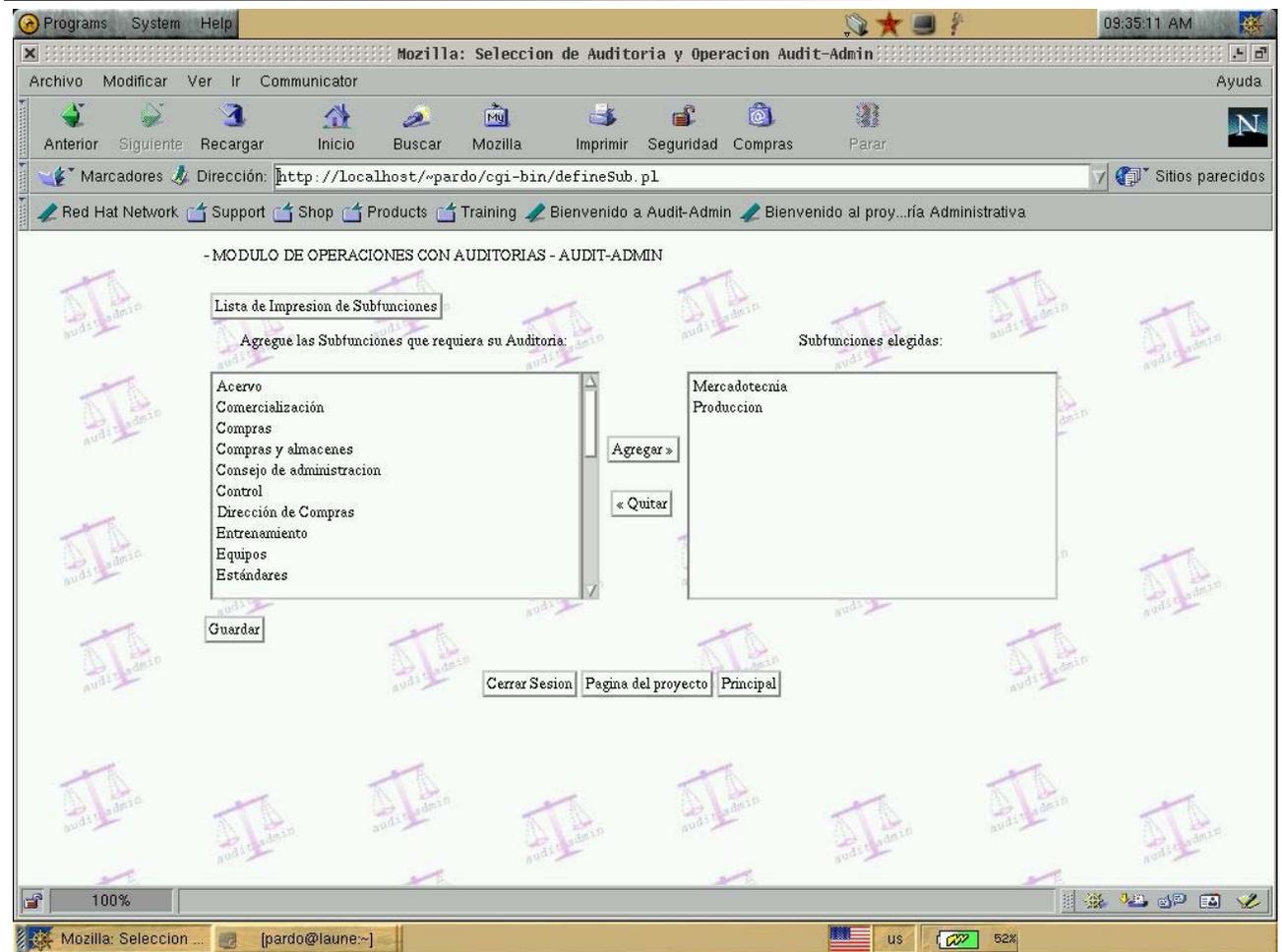
Puesto

A continuación se muestra la pantalla completa

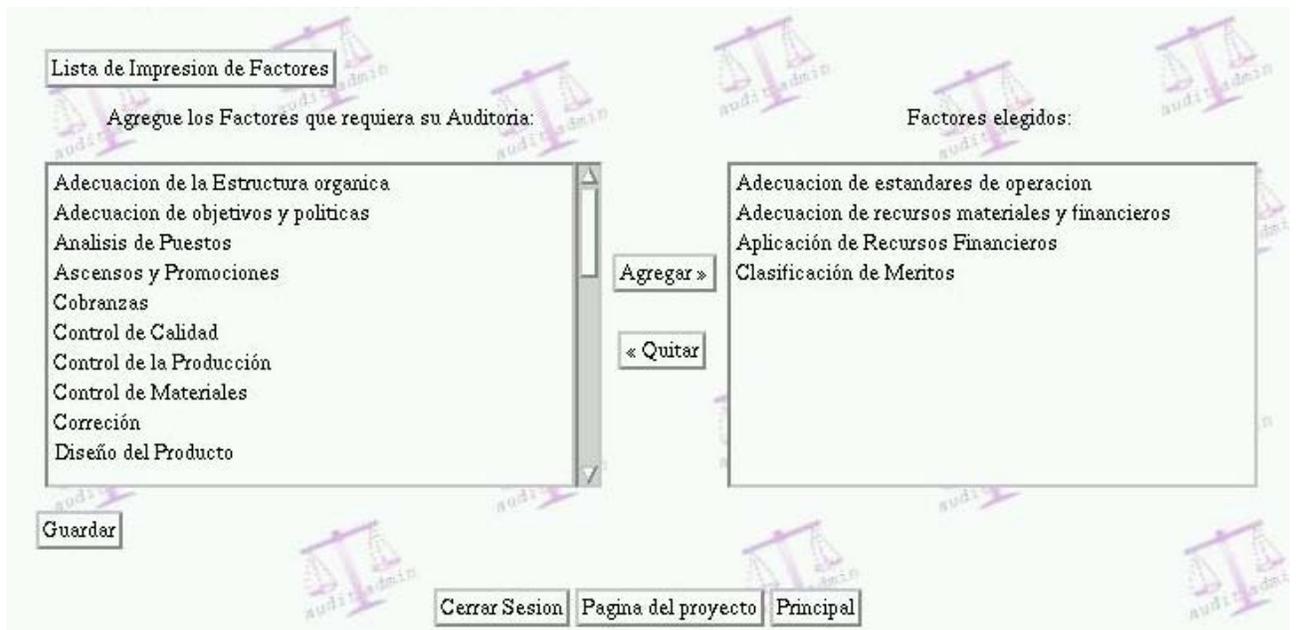


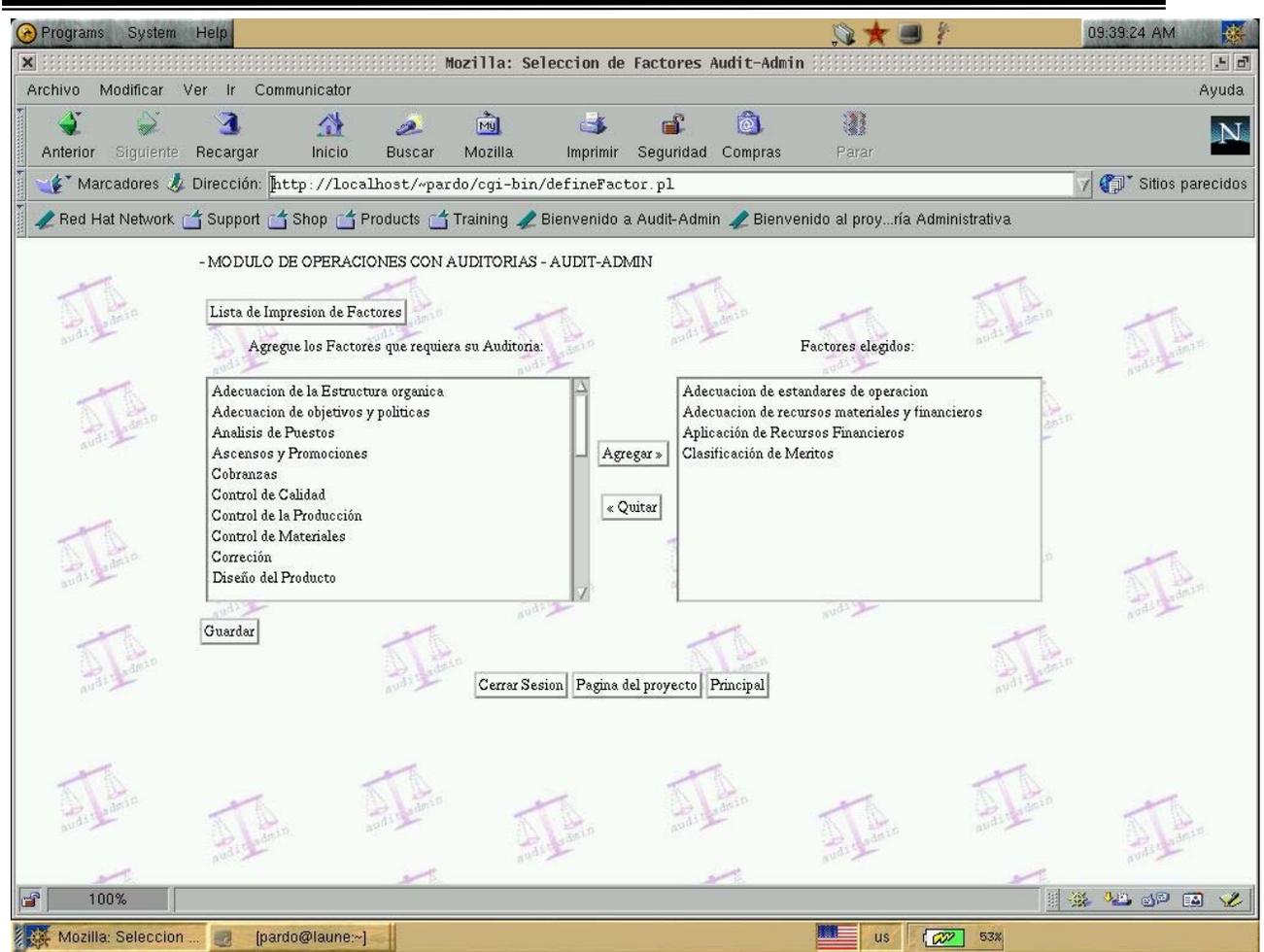
La siguiente pantalla muestra la elección de subfunciones del catálogo así como la opción de impresión de subfunciones del catálogo para ser entregadas al comité.





A continuación se muestra la pantalla de selección de subfunciones junto con la opción de la lista de impresión de factores para ser seleccionados por el comité.

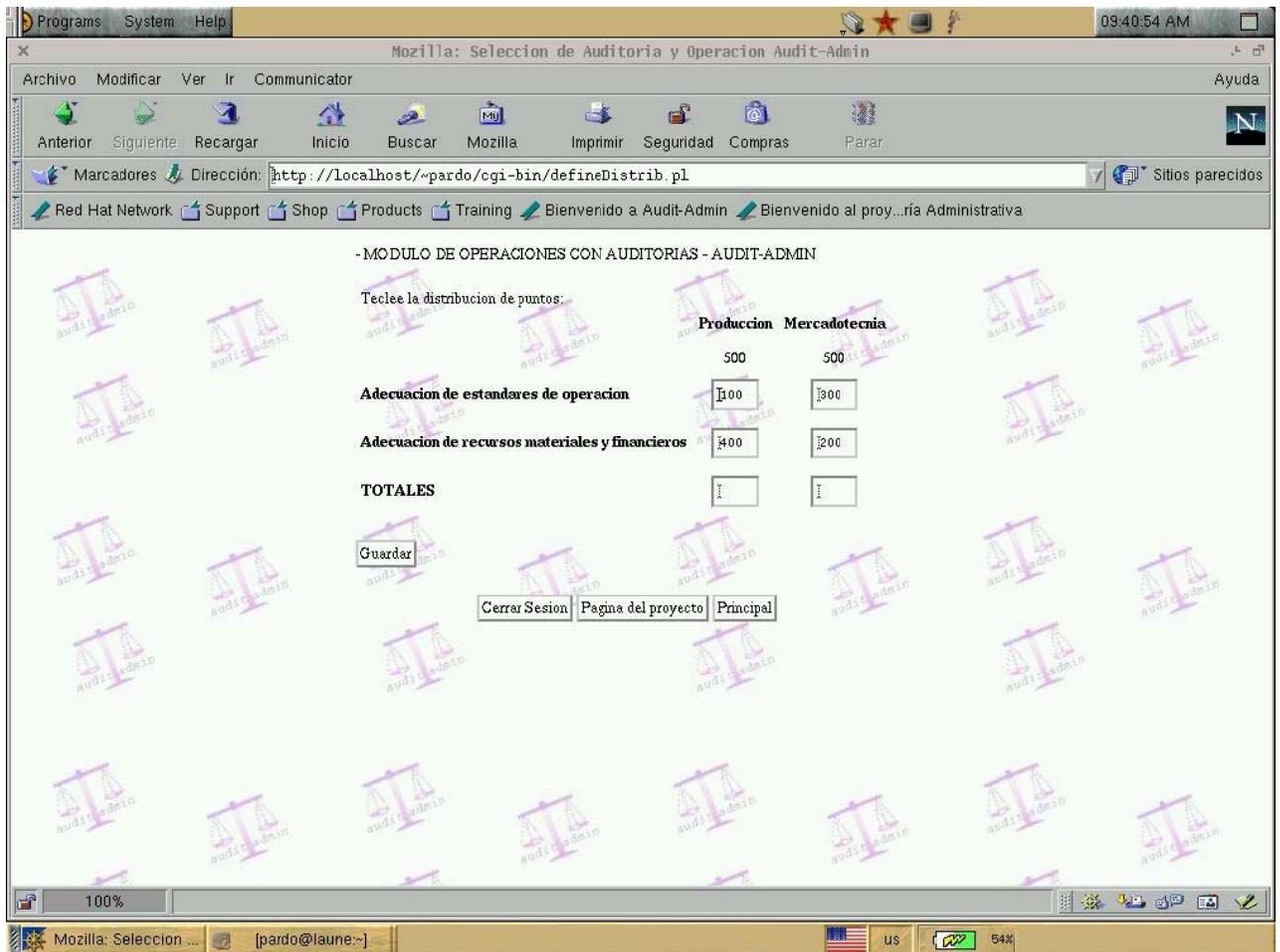




El siguiente gráfico muestra las opciones de inserción de la distribución de puntos.

Teclee la distribución de puntos:

	Produccion	Mercadotecnia
	500	500
Adecuacion de estandares de operacion	<input type="text" value="100"/>	<input type="text" value="300"/>
Adecuacion de recursos materiales y financieros	<input type="text" value="400"/>	<input type="text" value="200"/>
TOTALES	<input type="text" value=""/>	<input type="text" value=""/>



La siguiente pantalla muestra el módulo que permite realizar la elección de preguntas del catálogo para ser relacionadas a subfunciones y factores, o bien para ingresar nuevas preguntas al catálogo.

- MODULO DE OPERACIONES CON AUDITORIAS -Definicion de Preguntas - AUDIT-ADMIN -

Agregue las Preguntas que requiera su Auditoria y relacionelas a un factor-subfunción:

Cual es la base para determinar los plazos para el pago de prestamos?

Existen fuentes internas de recursos?

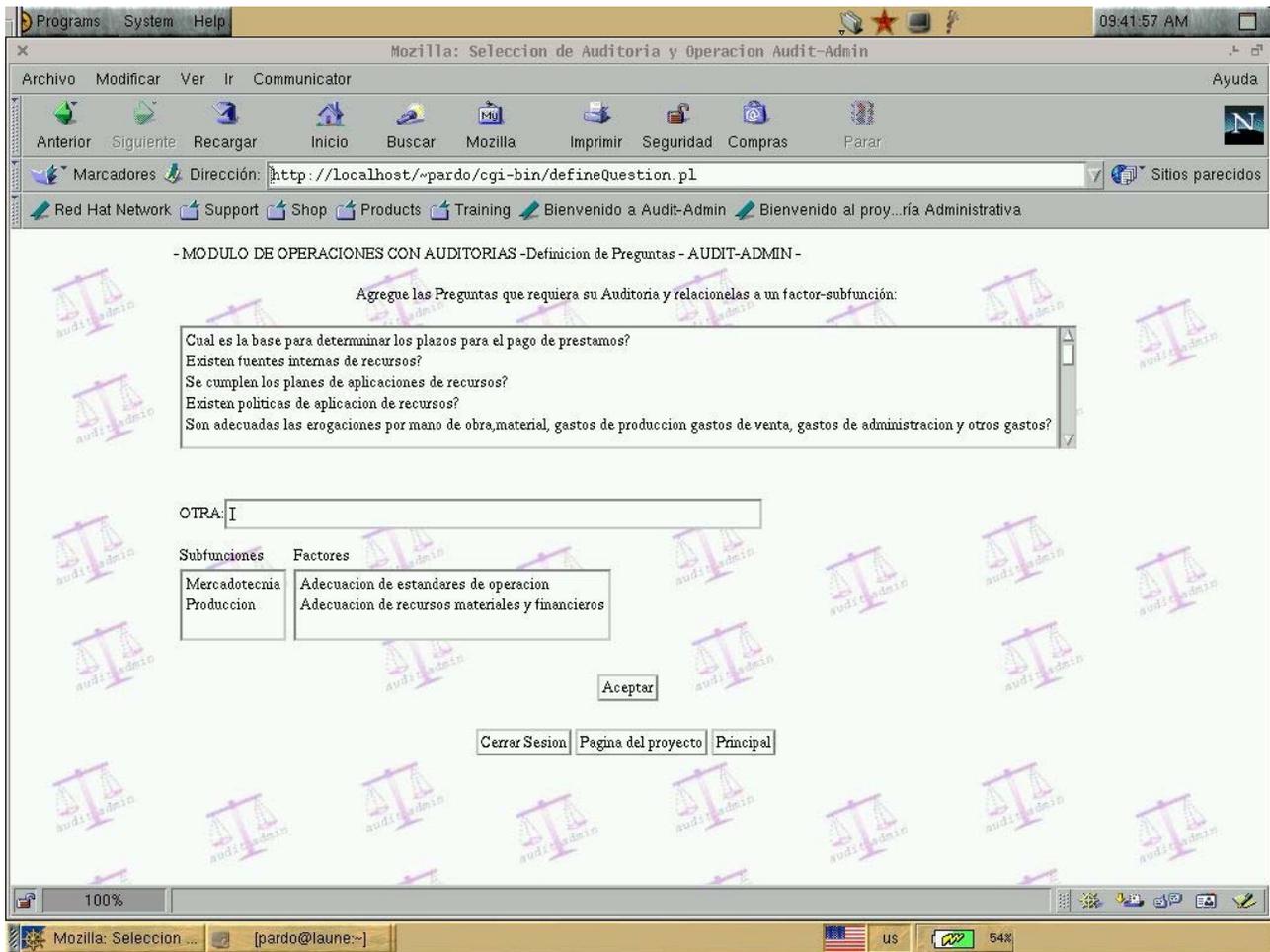
Se cumplen los planes de aplicaciones de recursos?

Existen politicas de aplicacion de recursos?

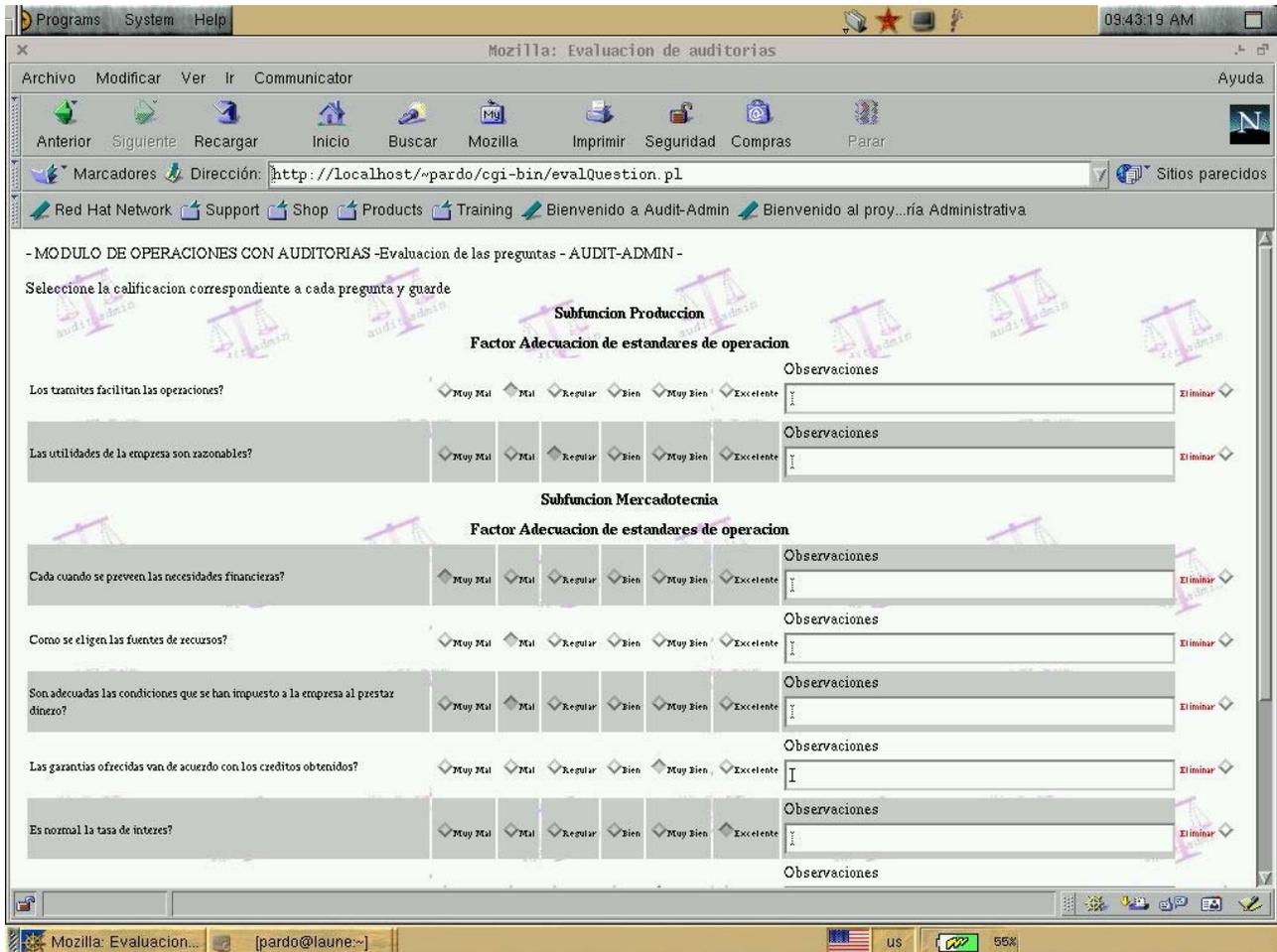
Son adecuadas las erogaciones por mano de obra,material, gastos de produccion gastos de venta, gastos de administracion y otros gastos?

OTRA:

Subfunciones	Factores
Mercadotecnia	Adecuacion de estandares de operacion
Produccion	Adecuacion de recursos materiales y financieros



La siguiente pantalla ejemplifica una evaluación hecha en el sistema donde es posible ingresar los resultados de cada pregunta así como las observaciones de cada una de ellas.



4.16 Conclusión sobre los resultados

El sistema de Auditoría Administrativa es una aportación realizada con el fin de facilitar las actividades en cada paso de una Auditoría Administrativa. El código generado es libre, modulado e independiente de plataforma, lo que permitirá realizar adaptaciones a aquellos interesados en usarlo o bien mejorarlo.

Por otro lado es posible instaurarlo como un servicio en línea dentro de una organización o bien, dentro de una escuela para fines didácticos.

Su diseño permite accesarlo desde cualquier equipo en red capaz de acceder al servicio vía http.

4.17 Aprobación o desaprobación de la hipótesis.

Con el desarrollo presentado y los resultados de los cuestionarios realizados se comprueba la hipótesis al constituir un sistema que ayuda en la automatización de auditorías administrativas y así presentar una herramienta más a aquellos que se interesen en el desarrollo de auditorías administrativas.

5. MARCO INSTRUMENTAL

5.1 – Propuestas de acción

- Se creará de una pagina Web que preste el servicio de informar acerca de el proyecto, de ofrecer el código del proyecto para evaluación, y la documentación para todo aquel interesado bajo la licencia GPL.
- Se publicará un artículo en la página de Linux en México, como una aportación de software libre al grupo de usuarios Linux. Pues el sistema se hizo en esta plataforma.
- Se ofrecerá el sistema a la Facultad de contaduría y administración para ser utilizado en las prácticas de los alumnos de la materia de Auditoría Administrativa.

A continuación se muestra la pantalla del proyecto propuesta



5.2.-Plan y programa de trabajo

1. Semana 1

Lunes - Miércoles

Terminar de desarrollar la pagina de Web para ser publicada en Internet

Jueves y viernes

Enviar información al grupo de usuarios de Linux en México para hacer publica la pagina del proyecto

2. Semana 2

Lunes

Asistir a la FCA para mostrar el sistema en el área de servicios de Internet y proponer su uso.

Autor:

Cynthia Gabriela Pardo Orozco

Othon de Mendizabal 127-302

Mexico DF

Cp.07710

Tel: (52)(55) 53681794

Email: Nekropsya@yahoo.com.mx

6. GLOSARIO

HTTP

Es la abreviatura de Hypertext Transfer Protocol. Es un conjunto de reglas o protocolo, que gobierna la transferencia de hipertexto entre dos o mas computadoras.

PERL

Es un lenguaje de programación (practical extracting and reporting language) práctico para extraer información de archivos de texto y generar informes a partir del contenido de los archivos.

SCRIPT O CÓDIGO

Es un archivo de texto que contiene una serie de instrucciones que se pueden ejecutar en la línea de ordenes o que se ejecutaran seguidas.

PLATAFORMA

Es el sistema operativo sobre el que se ejecutan las aplicaciones.

LINUX

Sistema operativo compatible con UNIX , cuya característica es que es libre, multiusuario y multitarea capaz de instalarse tanto en maquinas potentes para ofrecer servicios como computadoras personales.

GPL

(General Public License) –Especifica que se tiene la libertad de distribuir copias de software libre y de que se reciba el código fuente o que pueda conseguirlo si se requiere, de poder modificar el software de su versión original usar fragmentos de el en nuevos programas libres y no poder remover la GPL de los programas modificados ó fragmentos extraídos que ya la tenían.

CGI

Es un script especial para funcionar desde un servidor de Web.

CLIENTE

Es una máquina conectada a una red que realiza peticiones a otra maquina llamada servidor.

SERVIDOR

Es una máquina capaz de recibir peticiones de otras máquinas y ofrecerles servicios, aplicaciones o información.

7. BIBLIOGRAFÍA

José Antonio Fernández Arena. *La Auditoría Administrativa*. Diana (segunda edición). México DF.1992 ISBN: 968-13-2165-0

Rafael J. Hernández. *Análisis y Valuación de Puestos*. UNAM (quinta edición). México DF.

Brian W. Kernighan. *El Entorno de Programación Unix*. Prentice hall. isbn 968-880-067-8.

L.A. y C.P. Jorge Álvarez Anguiano. *Apuntes de Auditoria Administrativa*. UNAM.

Randal L. Schwartz. *Learning Perl*. O'Reilly. 1997.

Proyecto de Tutorial de Perl.
http://proton.ucting.udg.mx/tutorial/perl/tutorial_perl.html

Manual del usuario de PostgreSQL.
<http://es.tldp.org/Postgresql-es/web/navegable/user/user.html>

Levantar El Servicio Postgres En Linux.
http://www.postgresql.cl/colabora/inst_pg_01_cf.htm

Manual de Administración PostgreSQL.
<http://www.postgresql.org/docs/7.2/interactive/admin.html>

Auditoría Administrativa.
<http://www.benjaminfranklin.8k.com/artic09.htm>

Proyecto para compartir apuntes .
<http://www.geocities.com/gehg48/APUNTES.html>

Manuales Y Tutoriales De Programación.
<http://www.programacion.com/>

Manual de Perl en línea.
<http://www.geocities.com/SiliconValley/Station/8266/perl/>

CODIGO DEL ARCHIVO changeCat.pl

```
#!/usr/bin/perl -U
#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);

$JSCRIPT=<<EOF;

function valida() {
    var miForma=document.forma;
    if (miForma.newName.value==""){
        alert("El campo del nuevo dato en el catálogo no puede ser enviado
sin datos");
        miForma.newName.focus();
    }else{
        document.forma.submit();
    }
}

function spawnWin(options) {
    var miForma=document.forma;
    var valorOption=options[options.selectedIndex].value;
    var gatito= valorOption.split("#");
    miForma.newName.value=gatito[1];
    miForma.id.value=gatito[0];
}

}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Modificacion de Catalogos Audit-Admin',
                 -script=>$JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);
$catalogo = $query->param(catalogo);

#variables que crean el formato de la pagina que se despliega
$page_up = " <html>\n
<body bgcolor=white background=\"../Tesis/Imagenes/bglogito.jpg\">\n
<table align=center><tr><td>\n";
$page_down = "
</td></tr></table>\n";
```

```

    </body>\n
    </html>\n";
#botones para el regreso
$locacion1 = "http://cuib.unam.mx/~pardo/cgi-bin/valida.pl";
$locacion2 = "http://cuib.unam.mx/~pardo/Tesis/";
$locacion3 = "valida.pl?dato_id=$id_normal";
$botones = "<div align=center>
    <form method=post>
        <input type=\"button\" value=\"Cerrar Sesion\"
onclick=\"parent.location.href='$locacion1'\">\n
        <input type=\"button\" value=\"Pagina del proyecto\"
onclick=\"parent.location.href='$locacion2'\">\n
        <input type=\"button\" value=\"Principal\"
onclick=\"parent.location.href='$locacion3'\">\n
    </form></div>";

#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre:$datosUser[0][1]
#   AP:$datosUser[0][2]<br>
#   AM:$datosUser[0][3]<br>
#   LOG:$datosUser[0][4]<br>
#   PASS:$datosUser[0][5]<br>
#   TIPO:$datosUser[0][6]<br>
#   GRUPO:$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "3" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    <br>";

    print $page_up.$contents.$page_down;
}else{
    if($accion ne ""){
#si ya se inserto un catalogo recibe los parametros para realizar la
insercion en la base de datos
        #esto recibe
        #nombre =newName
        #catalogo
        #dato_id
        #accion
        $newName = $query->param(newName);
#nombre nuevo del elemento en el catalogo
        $id = $query->param(id);
#id del elemento a modificar
        $atributo = "descripcion";
        if($catalogo eq "subfunciones"){
            $atributo = "nombre";
            $nameCat = "cat_subfunciones";
            $idCat = "id_sub";
        }elseif($catalogo eq "factores"){
            $nameCat = "cat_factores";
            $idCat = "id_factor";

```

```

        $atributo = "nombre";
    }elseif($catalogo eq "clasificacion"){
        $nameCat = "clasificacion";
        $idCat = "id_clas";
    }elseif($catalogo eq "grupos"){
        $nameCat = "grupos";
        $idCat = "id_grupo";
    }
    $contents = "Los datos recibidos y modificados son son:<br>
    Catalogo: $catalogo<br>
    Elemento: $newName<br>
    identificador: $id<br>          ";
    print $page_up;

#IMPORTANTE EL CASO DE PREGUNTAS Y GRADOS SON CASOS ESPECIALES
#poner el que saca el valor maximo para designar el nuevo id

#hacer la insercion del nuevo usuario
Pg::doQuery($conn,"update $nameCat set $atributo='$newName' where
$idCat='$id'");
#print "insert into $nameCat values($maximo[0][0],'$newName')<br>";

#comprobacion de insercion
Pg::doQuery($conn,"select $atributo from $nameCat",\@descripcion);
#print "select descripcion from $nameCat";
$num = $#descripcion + 1;
print"<div align=center>
<form>
El cat&aacute;logo:$catalogo se ha actualizado:<br>
<SELECT NAME=\"subfunciones\" size=\"\$num\"><br>";

$x = 0;
while ( $descripcion[$x][0] ne "" ) {
    #imprime las subfunciones existentes
    print "<OPTION >$descripcion[$x][0]\n";
    $x++;
}
#referencia de lo que le mando a las ligas
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
print "</select></form></div>";
$locacion1 = "changeCat.pl?dato_id=$id_normal&catalogo=$catalogo";
$locacion2 = "opCatalogos.pl?dato_id=$id_normal";
$locacion3 = "valida.pl?dato_id=$id_normal";
$botones = "<div align=center>
<form method=post>
<input type=\"button\" value=\"Modificar otro elemento\"
onclick=\"parent.location.href='$locacion1'\">\n
<input type=\"button\" value=\"Operaciones con catalogos\"
onclick=\"parent.location.href='$locacion2'\">\n
<input type=\"button\" value=\"Principal\"
onclick=\"parent.location.href='$locacion3'\">\n
</form></div>";
print $botones.$page_down;

# print $page_up.$contents.$page_down;
}else{
#si no viene parametro es que apenas hay que elegir en que catalogo se va
a modificar y presentar la forma que permita hacerlo

```

```

        if($catalogo eq "subfunciones"){
#si se va a insertar en subfunciones

        Pg::doQuery($conn,"select id_sub,nombre FROM
cat_subfunciones",\@subfunciones);
        $num_subfunciones = $#subfunciones +1 ;
        if ($num_subfunciones > 10){
            $num = 10;
        }else{
            $num = $num_subfunciones;
        }
        $contents_up = "
        MODULO DE MODIFICACIONES -CATALOGO SUBFUNCIONES- AUDIT-ADMIN
        <form name=\"forma\" method=\"post\" action=changeCat.pl>
        Elija una la Subfunci&oacute;n a modificar:<br>
        <SELECT NAME=\"subfunciones\" size=\"\$num\"
onChange=spawnWin(this.options)><br>\n";
        print $page_up.$contents_up;
        $x = 0;
        while ( $subfunciones[$x][0] ne " " ) {
            #imprime las subfunciones existentes
            $subfunciones[$x][0] = ~ s/ //g;
            $subfunciones[$x][1] = ~ s/ //g;
            print "<OPTION
value=\"\$subfunciones[$x][0]#\$subfunciones[$x][1]\">\$subfunciones[$x][1]\n";

            $x++;
        }

        $contents_down = "
        </SELECT><br><br>
        <p>Nombre de la nueva subfunci&oacute;n<br>\n
        <input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
        <input type=\"hidden\" name=\"id\" value=\"\">\n
        <input type=\"hidden\" name=\"catalogo\" value=\"\$catalogo\">\n
        <input type=\"hidden\" name=\"dato_id\" value=\"\$id_normal\">\n
        <input type=\"hidden\" name=\"accion\" value=\"2\">\n
        <input type=\"button\" value=\"Aceptar\"
Onclick=\"valida()\"></p>
        </form>";

        print $contents_down.$botones.$page_down;
    }elseif($catalogo eq "factores"){
#si se va a insertar en factores

        Pg::doQuery($conn,"select id_factor,nombre FROM
cat_factores",\@factores);
        $num_factores = $#factores +1 ;
        if ($num_factores > 10){
            $num = 10;
        }else{
            $num = $num_factores;
        }
        $contents_up = "
        MODULO DE MODIFICACIONES -CATALOGO FACTORES- AUDIT-ADMIN
        <form name=\"forma\" method=\"post\" action=changeCat.pl>
        Los Factores existentes son:<br>
        <SELECT NAME=\"factores\" size=\"\$num\"
onChange=spawnWin(this.options)><br>\n";

```

```

print $page_up.$contents_up;
$x = 0;
while ( $factores[$x][0] ne "" ) {
    #imprime los factores existentes
    $factores[$x][0]=~ s/ //g;

    print "<OPTION
value=\"\${factores[$x][0]#\${factores[$x][1]}\>\${factores[$x][1]}\n";

    $x++;
}

$contents_down="
</SELECT><br><br>
<p>Nombre del nuevo Factor<br>\n
<input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
<input type=\"hidden\" name=\"id\" value=\"\">\n
<input type=\"hidden\" name=\"catalogo\" value=\"\${catalogo}\">\n
<input type=\"hidden\" name=\"dato_id\" value=\"\${id_normal}\">\n
<input type=\"hidden\" name=\"accion\" value=\"2\">\n
<input type=\"button\" value=\"Aceptar\"
OnClick=\"valida()\"></p>
</form>";

print $contents_down.$botones.$page_down;
#cierra elsif de factores
}elsif($catalogo eq "clasificacion"){
#si se va a insertar en clasificacion

Pg::doQuery($conn,"select id_clas,descripcion FROM
clasificacion",\@clasificacion);
$num_clasificacion = $#clasificacion +1 ;
if ($num_clasificacion > 10){
    $num = 10;
}else{
    $num = $num_clasificacion;
}
$contents_up = "
MODULO DE MODIFICACIONES -CATALOGO CLASIFICACION;N-
AUDIT-ADMIN
<form name=\"forma\" method=\"post\" action=changeCat.pl>
Las Clasificaciones existentes son:<br>
<SELECT NAME=\"clasificaciones\" size=\"\${num}\"
onChange=spawnWin(this.options)><br>\n";
print $page_up.$contents_up;
$x = 0;

while ( $clasificacion[$x][0] ne "" ) {
#imprime los factores existentes
    $clasificacion[$x][0] = ~ s/ //g;
    $clasificacion[$x][1] = ~ s/ //g;
    print "<OPTION
value=\"\${clasificacion[$x][0]#\${clasificacion[$x][1]}\>\${clasificacion[$x][1]}\n";

    $x++;
}

$contents_down = "
</SELECT><br><br>
<p>Nombre de la nueva Clasificacion<br>\n

```

```

        <input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
        <input type=\"hidden\" name=\"id\" value=\"\">\n
        <input type=\"hidden\" name=\"catalogo\" value=\"\${catalogo}\">\n
        <input type=\"hidden\" name=\"dato_id\" value=\"\${id_normal}\">\n
        <input type=\"hidden\" name=\"accion\" value=\"2\">\n
        <input type=\"button\" value=\"Aceptar\"
Onclick=\"valida()\"></p>
    </form>";

    print \$contents_down.$botones.$page_down;
#cierra elsif de clasificacion
}elsif(\${catalogo} eq "grupos"){
#si se va a insertar en clasificacion

    Pg::doQuery(\$conn,"select id_grupo,descripcion FROM
grupos",\@grupos);
    \$num_grupos = $#grupos +1 ;
    if (\$num_grupos > 10){
        \$num = 10;
    }else{
        \$num = \$num_grupos;
    }
    \$contents_up="
MODULO DE MODIFICACIONES -CATALOGO GRUPOS- AUDIT-ADMIN
    <form name=\"forma\" method=\"post\" action=changeCat.pl>
    Los Grupos existentes son:<br>
    <SELECT NAME=\"grupos\" size=\"\$num\"
onChange=spawnWin(this.options)><br>\n";
    print \$page_up.$contents_up;
    \$x = 0;
    while ( \$grupos[\$x][0] ne "" ) {
        #imprime los factores existentes
        \$grupos[\$x][0] = ~ s/ //g;
        \$grupos[\$x][1] = ~ s/ //g;
        print "<OPTION
value=\"\${grupos[\$x][0]}\#\${grupos[\$x][1]}\">\${grupos[\$x][1]}\n";

        \$x++;
    }

    \$contents_down = "
</SELECT><br><br>
    <p>Nombre del nuevo Grupo<br>\n
    <input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
    <input type=\"hidden\" name=\"id\" value=\"\">\n
    <input type=\"hidden\" name=\"catalogo\" value=\"\${catalogo}\">\n
    <input type=\"hidden\" name=\"dato_id\" value=\"\${id_normal}\">\n
    <input type=\"hidden\" name=\"accion\" value=\"2\">\n
    <input type=\"button\" value=\"Aceptar\"
Onclick=\"valida()\"></p>
    </form>";

    print \$contents_down.$botones.$page_down;
}
#cierra elsif de grupos
}
#cierre del else
exit;

```

CODIGO DEL ARCHIVO choseAudit.pl

```
#!/usr/bin/perl -U
#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);

$JSCRIPT=<<EOF;

function valida() {
    var miForma=document.forma;
    if (miForma.newName.value==""){
        alert("El campo del nuevo dato en el catálogo no puede ser enviado
sin datos");
        miForma.newName.focus();
    }else{
        document.forma.submit();
    }
}

function spawnWin(options) {
    var miForma=document.forma;
    var valorOption=options[options.selectedIndex].value;
    var gatito= valorOption.split("#");
    miForma.newName.value=gatito[1];
    miForma.id.value=gatito[0];
}

}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Modificacion de Catalogos Audit-Admin',
                -script=>$JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);
$catalogo = $query->param(catalogo);

#variables que crean el formato de la pagina que se despliega
$page_up = " <html>\n
    <body bgcolor=white background=\"../Tesis/Imagenes/bglogito.jpg\">\n
    <table align=center><tr><td>\n";
$page_down = "
    </td></tr></table>\n";
```

```

    </body>\n
    </html>\n";
#botones para el regreso
$locacion1 = "http://cuib.unam.mx/~pardo/cgi-bin/valida.pl";
$locacion2 = "http://cuib.unam.mx/~pardo/Tesis/";
$locacion3 = "valida.pl?dato_id=$id_normal";
$botones = "<div align=center>
    <form method=post>
        <input type=\"button\" value=\"Cerrar Sesion\"
onclick=\"parent.location.href='$locacion1'\">\n
        <input type=\"button\" value=\"Pagina del proyecto\"
onclick=\"parent.location.href='$locacion2'\">\n
        <input type=\"button\" value=\"Principal\"
onclick=\"parent.location.href='$locacion3'\">\n
    </form></div>";

#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre:$datosUser[0][1]
#   AP:$datosUser[0][2]<br>
#   AM:$datosUser[0][3]<br>
#   LOG:$datosUser[0][4]<br>
#   PASS:$datosUser[0][5]<br>
#   TIPO:$datosUser[0][6]<br>
#   GRUPO:$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "3" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    <br>";

    print $page_up.$contents.$page_down;
}else{
    if($accion ne ""){
#si ya se inserto un catalogo recibe los parametros para realizar la
insercion en la base de datos
        #esto recibe
        #nombre =newName
        #catalogo
        #dato_id
        #accion
        $newName = $query->param(newName);
#nombre nuevo del elemento en el catalogo
        $id = $query->param(id);
#id del elemento a modificar
        $atributo = "descripcion";
        if($catalogo eq "subfunciones"){
            $atributo = "nombre";
            $nameCat = "cat_subfunciones";
            $idCat = "id_sub";
        }elseif($catalogo eq "factores"){
            $nameCat = "cat_factores";
            $idCat = "id_factor";

```

```

        $atributo = "nombre";
    }elseif($catalogo eq "clasificacion"){
        $nameCat = "clasificacion";
        $idCat = "id_clas";
    }elseif($catalogo eq "grupos"){
        $nameCat = "grupos";
        $idCat = "id_grupo";
    }
    $contents = "Los datos recibidos y modificados son son:<br>
    Catalogo: $catalogo<br>
    Elemento: $newName<br>
    identificador: $id<br>          ";
    print $page_up;

#IMPORTANTE EL CASO DE PREGUNTAS Y GRADOS SON CASOS ESPECIALES
#poner el que saca el valor maximo para designar el nuevo id

#hacer la insercion del nuevo usuario
Pg::doQuery($conn,"update $nameCat set $atributo='$newName' where
$idCat='$id'");
#print "insert into $nameCat values($maximo[0][0],'$newName')<br>";

#comprobacion de insercion
Pg::doQuery($conn,"select $atributo from $nameCat",\@descripcion);
#print "select descripcion from $nameCat";
$num = $#descripcion + 1;
print"<div align=center>
<form>
El cat&aacute;logo:$catalogo se ha actualizado:<br>
<SELECT NAME=\"subfunciones\" size=\"\$num\"><br>";

$x = 0;
while ( $descripcion[$x][0] ne "" ) {
    #imprime las subfunciones existentes
    print "<OPTION >$descripcion[$x][0]\n";
    $x++;
}
#referencia de lo que le mando a las ligas
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
print "</select></form></div>";
$locacion1 = "changeCat.pl?dato_id=$id_normal&catalogo=$catalogo";
$locacion2 = "opCatalogos.pl?dato_id=$id_normal";
$locacion3 = "valida.pl?dato_id=$id_normal";
$botones = "<div align=center>
<form method=post>
<input type=\"button\" value=\"Modificar otro elemento\"
onclick=\"parent.location.href='$locacion1'\">\n
<input type=\"button\" value=\"Operaciones con catalogos\"
onclick=\"parent.location.href='$locacion2'\">\n
<input type=\"button\" value=\"Principal\"
onclick=\"parent.location.href='$locacion3'\">\n
</form></div>";
print $botones.$page_down;

# print $page_up.$contents.$page_down;
}else{
#si no viene parametro es que apenas hay que elegir en que catalogo se va
a modificar y presentar la forma que permita hacerlo

```

```

        if($catalogo eq "subfunciones"){
#si se va a insertar en subfunciones

        Pg::doQuery($conn,"select id_sub,nombre FROM
cat_subfunciones",\@subfunciones);
        $num_subfunciones = $#subfunciones +1 ;
        if ($num_subfunciones > 10){
            $num = 10;
        }else{
            $num = $num_subfunciones;
        }
        $contents_up = "
        MODULO DE MODIFICACIONES -CATALOGO SUBFUNCIONES- AUDIT-ADMIN
        <form name=\"forma\" method=\"post\" action=changeCat.pl>
        Elija una la Subfunci&oacute;n a modificar:<br>
        <SELECT NAME=\"subfunciones\" size=\"$num\"
onChange=spawnWin(this.options)><br>\n";
        print $page_up.$contents_up;
        $x = 0;
        while ( $subfunciones[$x][0] ne " " ) {
            #imprime las subfunciones existentes
            $subfunciones[$x][0] = ~ s/ //g;
            $subfunciones[$x][1] = ~ s/ //g;
            print "<OPTION
value=\" $subfunciones[$x][0]#$subfunciones[$x][1]\">$subfunciones[$x][1]\n";

            $x++;
        }

        $contents_down = "
        </SELECT><br><br>
        <p>Nombre de la nueva subfunci&oacute;n<br>\n
        <input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
        <input type=\"hidden\" name=\"id\" value=\"\">\n
        <input type=\"hidden\" name=\"catalogo\" value=\"$catalogo\">\n
        <input type=\"hidden\" name=\"dato_id\" value=\"$id_normal\">\n
        <input type=\"hidden\" name=\"accion\" value=\"2\">\n
        <input type=\"button\" value=\"Aceptar\"
Onclick=\"valida()\"></p>
        </form>";

        print $contents_down.$botones.$page_down;
    }elseif($catalogo eq "factores"){
#si se va a insertar en factores

        Pg::doQuery($conn,"select id_factor,nombre FROM
cat_factores",\@factores);
        $num_factores = $#factores +1 ;
        if ($num_factores > 10){
            $num = 10;
        }else{
            $num = $num_factores;
        }
        $contents_up = "
        MODULO DE MODIFICACIONES -CATALOGO FACTORES- AUDIT-ADMIN
        <form name=\"forma\" method=\"post\" action=changeCat.pl>
        Los Factores existentes son:<br>
        <SELECT NAME=\"factores\" size=\"$num\"
onChange=spawnWin(this.options)><br>\n";

```

```

print $page_up.$contents_up;
$x = 0;
while ( $factores[$x][0] ne "" ) {
    #imprime los factores existentes
    $factores[$x][0]=~ s/ //g;

    print "<OPTION
value=\"\$factores[$x][0]#\${factores[$x][1]}\>\${factores[$x][1]}\n";

    $x++;
}

$contents_down="
</SELECT><br><br>
<p>Nombre del nuevo Factor<br>\n
<input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
<input type=\"hidden\" name=\"id\" value=\"\">\n
<input type=\"hidden\" name=\"catalogo\" value=\"\$catalogo\">\n
<input type=\"hidden\" name=\"dato_id\" value=\"\$id_normal\">\n
<input type=\"hidden\" name=\"accion\" value=\"2\">\n
<input type=\"button\" value=\"Aceptar\"
OnClick=\"valida()\"></p>
</form>";

print $contents_down.$botones.$page_down;
#cierra elsif de factores
}elsif($catalogo eq "clasificacion"){
#si se va a insertar en clasificacion

Pg::doQuery($conn,"select id_clas,descripcion FROM
clasificacion",\@clasificacion);
$num_clasificacion = $#clasificacion +1 ;
if ($num_clasificacion > 10){
    $num = 10;
}else{
    $num = $num_clasificacion;
}
$contents_up = "
MODULO DE MODIFICACIONES -CATALOGO CLASIFICACION;N-
AUDIT-ADMIN
<form name=\"forma\" method=\"post\" action=changeCat.pl>
Las Clasificaciones existentes son:<br>
<SELECT NAME=\"clasificaciones\" size=\"\$num\"
onChange=spawnWin(this.options)><br>\n";
print $page_up.$contents_up;
$x = 0;

while ( $clasificacion[$x][0] ne "" ) {
#imprime los factores existentes
    $clasificacion[$x][0] = ~ s/ //g;
    $clasificacion[$x][1] = ~ s/ //g;
    print "<OPTION
value=\"\$clasificacion[$x][0]#\${clasificacion[$x][1]}\>\${clasificacion[$x][1]}\n";

    $x++;
}

$contents_down = "
</SELECT><br><br>
<p>Nombre de la nueva Clasificacion<br>\n

```

```

        <input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
        <input type=\"hidden\" name=\"id\" value=\"\">\n
        <input type=\"hidden\" name=\"catalogo\" value=\"\$catalogo\">\n
        <input type=\"hidden\" name=\"dato_id\" value=\"\$id_normal\">\n
        <input type=\"hidden\" name=\"accion\" value=\"2\">\n
        <input type=\"button\" value=\"Aceptar\"
OnClick=\"valida()\"></p>
    </form>";

    print $contents_down.$botones.$page_down;
#cierra elsif de clasificacion
}elsif($catalogo eq "grupos"){
#si se va a insertar en clasificacion

    Pg::doQuery($conn,"select id_grupo,descripcion FROM
grupos", \@grupos);
    $num_grupos = $#grupos +1 ;
    if ($num_grupos > 10){
        $num = 10;
    }else{
        $num = $num_grupos;
    }
    $contents_up="
MODULO DE MODIFICACIONES -CATALOGO GRUPOS- AUDIT-ADMIN
    <form name=\"forma\" method=\"post\" action=changeCat.pl>
    Los Grupos existentes son:<br>
    <SELECT NAME=\"grupos\" size=\"\$num\"
onChange=spawnWin(this.options)><br>\n";
    print $page_up.$contents_up;
    $x = 0;
    while ( $grupos[$x][0] ne "" ) {
        #imprime los factores existentes
        $grupos[$x][0] = ~ s/ //g;
        $grupos[$x][1] = ~ s/ //g;
        print "<OPTION
value=\"\$grupos[$x][0]#\$grupos[$x][1]\"> \$grupos[$x][1]\n";

        $x++;
    }

    $contents_down = "
    </SELECT><br><br>
    <p>Nombre del nuevo Grupo<br>\n
    <input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
    <input type=\"hidden\" name=\"id\" value=\"\">\n
    <input type=\"hidden\" name=\"catalogo\" value=\"\$catalogo\">\n
    <input type=\"hidden\" name=\"dato_id\" value=\"\$id_normal\">\n
    <input type=\"hidden\" name=\"accion\" value=\"2\">\n
    <input type=\"button\" value=\"Aceptar\"
OnClick=\"valida()\"></p>
    </form>";

    print $contents_down.$botones.$page_down;
}
#cierra elsif de grupos
}
#cierre del else
exit;

```

CODIGO DEL ARCHIVO defineDistrib.pl

```
#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$JSCRIPT=<<EOF;
var nombreInput;
var total=0;
var ids,value1;
function doString(nombreInput,valor){
    ids=nombreInput+":"+ids;
    value1=valor.toString()+":"+value1;
    valor++;
    total=valor + total;
    total=total -1 ;
}

function sumar (c) {
    var miForma;
    var v, total, lacol;
    var losIds, subfunciones;
    var eltotal, elInput, numerin;

    miForma = document.forma;
    v = total = numerin = 0;
    losIds = miForma.ids_matriz.value.split(":");
    subfunciones = miForma.ids_sub.value.split (":");
    eltotal="",elInput="";

    while (v < losIds.length) {
        lacol = losIds[v].split("x");
        if(lacol[1] == c) {
            elInput = "miForma.i" + losIds[v] + ".value";
            numerin = parseInt (eval (elInput));
            if (isNaN (numerin)) {
                numerin = 0;
            }
            total = total + numerin;
            eltotal = "miForma.totalitox" + lacol[1] + ".value=" + total;
            eval (eltotal);
        }
        v ++;
    }
}

function guardar(){
    var miForma=document.forma;
    var separator=":";

```

```

var separa="x";

var subfunciones1=miForma.ids_sub.value;
var subfunciones=subfunciones1.split(separator);

var totales1=miForma.puntosTotales.value;
var totales=totales1.split(separator);
var eltotal="",elInput="";
var losIds1=miForma.ids_matriz.value;
var losIds=losIds1.split(separator);
var
c=0,v=0,limite=subfunciones.length,limiteIds=losIds.length,total=0;
var lacol;
var numerin = 0;

while (c < limite) {
  while (v < limiteIds) {
    lacol=losIds[v].split(separa);
    if(lacol[1]==subfunciones[c]){
      elInput="miForma.i"+losIds[v] + ".value";
      numerin = parseInt (eval (elInput));
      if (numerin == 0) {
        alert ("Ningun campo puede ser 0");
        break;
      }
      total=total+numerin;
      eltotal="miForma.totalitox"+lacol[1]+".value="+total;
      eval (eltotal);
    }
    v++;
  }

  if (isNaN (total)) {
    alert ("Ningun campo puede quedar vacio");
    break;
  }
  if (total != totales[c]){
    alert ("El total es "+total+" y deberia ser "+ totales[c]+" en la
columna "+subfunciones[c]);
    break;
  }

  total=0;
  v=0;
  c++;
}

miForma.submit();

}

EOF
;

```

```

#abre conexion
$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

```

```

print $query->header;

print start_html(-title=>'Seleccion de Auditoria y Operacion Audit-
Admin',
                -script=>$JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);
$id_audit = $query->param(audit);
$ids_sub = $query->param(ids_sub);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

#####
#Este es para checar el tipo de usuario
#####

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre:$datosUser[0][1]
#   AP:$datosUser[0][2]<br>
#   AM:$datosUser[0][3]<br>
#   LOG:$datosUser[0][4]<br>
#   PASS:$datosUser[0][5]<br>
#   TIPO:$datosUser[0][6]<br>
#   GRUPO:$datosUser[0][7]<br>

#usuario no es del tipo correcto
if ($id_normal eq "" ||$datosUser[0][6] ne "1" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br> <br> ";

    print $page_up.$contents.$page_down;
}else{
    #####
    #Si accion viene sin ningun contenido
    #####
    if($accion == 1){
#si accion viene 1 o sea se ha capturado la distribucion y se esta
guardando ahora.
        $ids_matriz = $query->param(ids_matriz);

```

#este dato es importante porque contiene los nombres de los objetos por los que hay que preguntar

```

@arreglo_inputs = split(/:/, $ids_matriz);
$x = 0;
print $page_up;

```

```

#####
#verifica si hay registros en la tabla evaluacion
#####

```

```

Pg::doQuery($conn,"select id_auditoria from evaluacion where
id_auditoria=$id_audit",\@evaluadas);

```

```

#####
#cuando existen evaluaciones asignados con anterioridad
#####
if( $evaluadas[0][0] ne ""){
    print $page_up."La auditoria seleccionada cuenta ya con
evaluaciones asignadas, no es posible modificar la distribucion de
puntos.<br> \n";

```

```

    print $botones.$page_down;

```

```

} else{

```

```

#para traer el nombre de la empresa
Pg::doQuery($conn,"select nombre_emp from empresa where
id_empresa in (select id_empresa from auditoria where
id_auditoria=$id_audit)",\@nombreEmp);

```

```

print "La distribucion de puntos quedo de la siguiente
manera:<p>";

```

```

$checatotal = 0;

```

```

while ($arreglo_inputs[$x] ne ""){
    $elInput = "i$arreglo_inputs[$x]";
    @losElementos = split(/x/, $arreglo_inputs[$x]);
    $factor = $losElementos[0];
    $subfun = $losElementos[1];
    $valor_input = $query->param($elInput);

```

```

Pg::doQuery($conn,"select nombre from cat_factores where
id_factor=$factor",\@nombreFact);
Pg::doQuery($conn,"select nombre from cat_subfunciones where
id_sub=$subfun",\@nombreSub);
Pg::doQuery($conn,"update distrib_puntos set puntos=$valor_input
where id_auditoria=$id_audit and id_sub='$subfun' and
id_factor='$factor'");

```

```

    $checatotal = $checatotal + $valor_input;
    print "<li> $valor_input,factor $nombreFact[0][0], subfuncion
$nombreSub[0][0]<p>";
    $x ++;
}

```

```

Pg::doQuery($conn,"select sum(puntos) from distrib_puntos where
id_auditoria=$id_audit",\@checa);

```

```

#para checar que lo que esta en la base sea lo que esta en pantalla
    if($checatotal == $checa[0][0]){
        print "La insercion de puntos en la empresa <b>\n"
$nombreEmp[0][0] \</b> se realizo con exito<br>\n";
    }else{
        print "<font color=red>Fallo en la insercion de
valores,vuelva a intentar o bien consulte con el administrador del
sistema.</font><br>\n";
    }
    print $botones.$page_down;
}
} else {

#####
#AQUI VAN LAS SUBFUNCIONES CONTRA LOS FACTORES
#Y PONER ESPACIOS PARA DISTRIBUIR LOS PUNTOS
#####

    #trae las subfunciones existentes en la auditoria
    Pg::doQuery($conn,"select id_sub,nombre from cat_subfunciones where
id_sub in (select id_sub from ponderacion_sub where
id_auditoria=$id_audit) order by id_sub",\@subfunciones);

    Pg::doQuery($conn,"select porcentaje from ponderacion_sub where
id_auditoria=$id_audit order by id_sub",\@percentSub);

    Pg::doQuery($conn,"select id_factor,nombre from cat_factores where
id_factor in (select distinct id_factor from distrib_puntos where
id_auditoria=$id_audit)",\@factores);

    $x = 0;
    $num_subfunciones = @subfunciones ;
    if ($num_subfunciones > 10 && scalar(@factores)>0 ){
        $num = 10;
    }else{
        $num = $num_subfunciones;
        if($num > 0 && scalar(@factores)>0 ){
        }else{
            $num = 0;
        }
    }
    if($num > 0){
        $contents_up = "
- MODULO DE OPERACIONES CON AUDITORIAS - AUDIT-ADMIN \n
<p> \n
<form method=post name=\"forma\" action=\"defineDistrib.pl\">
\n
<table><tr>\n";

        print $page_up.$contents_up;

        print "<tr><td colspan=5>Teclee la distribucion de
puntos:</td></tr><tr><td>&nbsp;</td> \n";
        $x = 0;

        while ( $subfunciones[$x][0] ne "" ) {
            $subfunciones[$x][0] =~ s/ //g;
            #####
            #para sacar los valores anteriores de las subfunciones

```

```

#####
Pg::doQuery($conn,"select porcentaje from ponderacion_sub
where id_sub=$subfunciones[$x][0]
id_auditoria=$id_audit)", \@percentExist);
    $percentSub[$x][0] = $percentSub[$x][0] * 10;
    print "<td align=center><b> $subfunciones[$x][1]</b><p
align=center>$percentSub[$x][0] </td>\n";
    $lasSubfunciones = "$lasSubfunciones:$subfunciones[$x][0]";
    $losTotales = "$losTotales:$percentSub[$x][0]";
    $x ++;
}
$x = 0;
$y = 0;
$lenSub = scalar(@subfunciones);
$limite = scalar(@factores);
while($x <= $limite){
    if($x < $limite){
        print "<tr><td><b>$factores[$x][1]</b> </td>\n";
    }else{
        print "<tr><td><b>TOTALES</b> </td>\n";
    }
    while ( $subfunciones[$y][0] ne "" ) {
        $factores[$x][0]=~ s/ //g;
        $subfunciones[$y][0]=~ s/ //g;
        if($x < $limite){
            $elementos="$factores[$x][0]x$subfunciones[$y][0]";
#####
#####
#para sacar los valores anteriores de la distribucion de
puntos en caso de existir este

#####
#####
Pg::doQuery($conn,"select puntos from distrib_puntos where
id_auditoria=$id_audit and id_sub=$subfunciones[$y][0] and
id_factor=$factores[$x][0]", \@puntosAnt);

        print "<td align=center><input type=text
name=\"i$elementos\" size=4 onChange=\"sumar($subfunciones[$y][0])\"
value=$puntosAnt[0][0]> </td><!--Los nombres de estos objetos se forman
con el factor, equis (x) la subfuncion-->\n";
#        print "ELEMENTOS $elementos<br>";
        $losIds="$losIds:$elementos";
    }else{
        print "<td align=center><input type=text
name=\"totalitox$subfunciones[$y][0]\" size=4> </td><!--Los nombres de
estos objetos totales se forman del total quin la subfuncion-->\n";
    }
    $y++;
}

$y=0;
print "</tr>";
print "";
$x++;
}

$losIds =~s/^://;
#estas son las variables que le permiten al cgi saber cuantas
subfunciones o bien columnas hay que revizar y cuales son los totales a
los que tiene que llegar la suma de cada columna

```

```

$lasSubfunciones =~s/^:////;
$losTotales =~s/^:////;

$content2_down = "</tr></table><br>\n

<input class=\"sbtt\" style=\"width: 75px;\" type=\"button\"
value=\"Guardar\" name=\"Guardar\" onClick=\"guardar();\n\">
<input type=\"hidden\" name=\"dato_id\" value=\"$id_normal\">\n
<input type=\"hidden\" name=\"audit\" value=\"$id_audit\">\n
<input type=\"hidden\" name=\"ids_sub\"
value=\"$lasSubfunciones\">\n
<input type=\"hidden\" name=\"puntosTotales\"
value=\"$losTotales\">\n
<input type=\"hidden\" name=\"ids_matriz\" value=\"$losIds\">\n
<input type=\"hidden\" name=\"accion\" value=1>\n
</form>\n";

print $content2_down.$botones.$page_down;
}else {
#cuando no existen auditorias creadas.
print $page_up;
print "La auditoria seleccionada no contiene subfunciones
asignadas con pesos relacionados<br>\n
o bien, no contiene aun Factores asociados.\n";
print $empresas[$x][0];
print $botones.$page_down;
}
}
}
#cierre del else

exit;

```

CODIGO DEL ARCHIVO defineFactor.pl

```

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$JSCRIPT = <<EOF;
var destQAL = document.editQALForm.destQAL;
var ie = document.all?1:0;
var submitted = 0;

function AddToTheList()
{

```

```
var fromAB = document.editQALForm.fromAB;
var destQAL= document.editQALForm.destQAL;
var selectedValue = "";
var selectedText = "";
if ( fromAB.selectedIndex < 0 )
{
    alert("Por favor, primero seleccione un elemento de la lista");
    if (fromAB.options[0] != null)
        fromAB.options[0].selected = true;
    fromAB.focus();
    return false;
}
else
{
    selectedValue = fromAB.options[fromAB.selectedIndex].value;
    selectedText = fromAB.options[fromAB.selectedIndex].text;
    destQAL.options[destQAL.length] = new
Option(selectedText,selectedValue);
    fromAB.options[fromAB.selectedIndex] = null;
}
}
function RemoveFromTheList()
{
    var fromAB = document.editQALForm.fromAB;
    var destQAL= document.editQALForm.destQAL;
    var selectedValue = "";
    var selectedText = "";
    if ( destQAL.selectedIndex < 0 )
    {
        alert("Por favor, primero seleccione un elemento de la lista");
        if (destQAL.options[0] != null)
            destQAL.options[0].selected = true;
        destQAL.focus();
        return false;
    }
    else
    {
        selectedValue = destQAL.options[destQAL.selectedIndex].value;
        selectedText = destQAL.options[destQAL.selectedIndex].text;
        fromAB.options[fromAB.length] = new
Option(selectedText,selectedValue);
        destQAL.options[destQAL.selectedIndex] = null;
    }
}
function disapleButtons()
{
    var fromAB = document.editQALForm.fromAB;
    var destQAL= document.editQALForm.destQAL;
    if (fromAB.options.length > 0 )
    {
        if (destQAL.options.length < 5)
            document.editQALForm.AddQ.disabled = 0;
        else
            document.editQALForm.AddQ.disabled = 1;
    }
    else
        document.editQALForm.AddQ.disabled = 1;
    if (destQAL.options.length > 0)
        document.editQALForm.RemoveQ.disabled = 0;
    else
```

```

        document.editQALForm.RemoveQ.disabled = 1;
    }
function putItTogether()
{
    var destQAL= document.editQALForm.destQAL;
    var fullList = "";
    for(var i = 0; i < destQAL.length; i++)
    {
        fullList += (destQAL.options[i].text + "::");
    }
    fullList = fullList.substring(0, fullList.length-2);
    document.editQALForm.quickaddr.value = fullList;
    document.editQALForm.submit();
}
disapleButtons();

function guardaSelect(){
    var miForma = document.editQALForm;
    var longitud=miForma.destQAL.length;
    var i=0;
    var ids="";

    while (i < longitud){

        ids=miForma.destQAL.options[i].value+"::"+ids ;

        i++;

    }
    miForma.ids_fact.value=ids;

    miForma.submit();

}
EOF
;

#abre conexion
$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Seleccion de Factores Audit-Admin',
                 -script=>$JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);
$id_audit = $query->param(audit);
$ids_fact = $query->param(ids_fact);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];

```

```

$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

#####
#Este es para checar el tipo de usuario
#####

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

#usuario no es del tipo correcto
if ($id_normal eq "" ||$datosUser[0][6] ne "1" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br> <br> ";

    print $page_up.$contents.$page_down;
}else{
#####
#Si accion viene sin ningun contenido
#####
if($accion == 1){

    # para recoger los id's de las factores seleccionadas
    $ids_fact = $query->param(ids_fact);
    $cadenita = $ids_fact;
    @ids_fact1 = split(/:/, $ids_fact);
    $longitud = scalar(@ids_fact1);

    #verifica si hay puntos asignados a los factores pesos
    Pg::doQuery($conn,"select sum(puntos) from distrib_puntos where
id_auditoria=$id_audit",\@puntos);

    if( $puntos[0][0] > 0 ){
        print $page_up."Las factores seleccionadas contienen pesos
asignados, es necesario vaciar los puntos para editar nuevamente las
factores \n";

        print $botones.$page_down;

    }else{

#borra las factores existentes

```

```

Pg::doQuery($conn,"delete from distrib_puntos where
id_auditoria=$id_audit",\@puntos);
#busca las subfunciones de la auditoria
Pg::doQuery($conn,"select id_sub from ponderacion_sub where
id_auditoria=$id_audit",\@idSubAudit);
$lensubAudit = scalar(@idSubAudit);
#inserta los nuevos factores
$x = 0;
$y = 0;
while ($x < $longitud ){
    while ($y < $lensubAudit ){
        Pg::doQuery($conn,"insert into distrib_puntos
(id_auditoria,id_sub,id_factor)
values($id_audit,'$idSubAudit[$y][0]','$ids_fact1[$x]')");
        $y++;
    }
    $y=0;
    $x++;
}

#para traer el nombre de la empresa
Pg::doQuery($conn,"select nombre_emp from empresa where id_empresa
in (select id_empresa from auditoria where
id_auditoria=$id_audit)",\@nombreEmp);

print $page_up."La insercion de factores en la empresa <b>\n"
$nombreEmp[0][0] \</b> se realizo con exito<br>\n";

print $botones.$page_down;
}

}else{

#trae los factores existentes en los catalogos del sistema
Pg::doQuery($conn,"select id_factor,nombre from cat_factores where
id_factor not in (select id_factor from distrib_puntos where
id_auditoria=$id_audit) order by nombre",\@factoresCat);

#trae los factores existentes en la auditoria
Pg::doQuery($conn,"select id_factor,nombre from cat_factores where
id_factor in (select id_factor from distrib_puntos where
id_auditoria=$id_audit) order by nombre",\@factores);

$x = 0;
$num_factores = @factoresCat ;
if ($num_factores > 10){
    $num = 10;
}else{
    $num = $num_factores;
    if($num == 0){
        $num = scalar(@factores);
    }
}
if($num > 0){
#####
#trae las subfunciones existentes en la auditoria,"debe
haberlas"
#de lo contrario no es posible asignar factores
#####

```

```

Pg::doQuery($conn,"select id_auditoria from ponderacion_sub where
id_auditoria=$id_audit and id_sub > 0 and porcentaje > 0",\@subExist);

if($subExist[0][0] eq ""){
    print $page_up;
    print "<p>No es posible agregar factores sin haber agregado
subfunciones y pesos a la auditoria</p>&nbsp;";
    print $botones.$page_down;
}else{
    $contents_up = "
- MODULO DE OPERACIONES CON AUDITORIAS - AUDIT-ADMIN \n
<p>\n
<form method=post name=\"editQALForm\">\n
<table><tr>
    <td colspan=2>                <input type=\"button\"
name=\"listaPrint\" value=\"Lista de Impresion de Factores\"
onClick=\"window.open('imprime.pl?dato_id=$id_normal&tipo=fac&id_auditori
a=$id_audit','variable','width=600,height=450,scrollbars=yes,toolbar=yes'
)\"> </td></tr>
<tr>\n
    <td valign=\"middle\" align=\"center\">\n
    Agregue los Factores que requiera su Auditoria:<br> \n
<SELECT NAME=\"fromAB\" size=\"10\" width=\"300px\"
style=\"width:300px;\" ><br>\n";
    print $page_up.$contents_up;

    while ( $factoresCat[$x][0] ne " " ) {

        print "<OPTION
value=$factoresCat[$x][0]>$factoresCat[$x][1] \n";

        $x++;
    }

    $contents_down = "</SELECT></td>";

    print $contents_down;
    $contents2 = "<td valign=\"middle\" align=\"center\"><br>\n
<input class=\"sbtt\" style=\"width: 75px;\" type=\"button\"
value=\"Agregar »\" name=\"AddQ\" onClick=\"AddToTheList();\"><p>\n
<input class=\"sbtt\" style=\"width: 75px;\" type=\"button\"
value=\"&laquo; Quitar\" name=\"RemoveQ\"
onClick=\"RemoveFromTheList();\">\n
</td><td valign=\"middle\" align=\"center\">\n";

    print $contents2;

    print "Factores elegidos:<br><SELECT NAME=\"destQAL\"
size=\"10\" width=\"300px\" style=\"width:300px;\"><br>\n";
    $x = 0;
    while ( $factores[$x][0] ne " " ) {
        print "<OPTION value=$factores[$x][0]>$factores[$x][1] \n";
        $x ++;
    }

    $contents2_down = "</select></td></tr></table>\n
<input class=\"sbtt\" style=\"width: 75px;\"
type=\"button\" value=\"Guardar\" name=\"Guardar\"
onClick=\"guardaSelect();\n\">
<input type=\"hidden\" name=\"dato_id\"
value=\"$id_normal\">\n

```

```

                <input type=\"hidden\" name=\"audit\"
value=\"\$id_audit\">\n
                <input type=\"hidden\" name=\"ids_fact\" value=\"\">\n
                <input type=\"hidden\" name=\"accion\" value=1>\n
                </form>\n";

        print $contents2_down.$botones.$page_down;
    }
} else {
#cuando no existen auditorias creadas.
    print $page_up;
    print "No hay auditorias existentes<br>\n
        Requiere crear una nueva auditoria previamente";
    print $empresas[$x][0];
    print $botones.$page_down;
}
}
}
#cierre del else

exit;

CODIGO DEL ARCHIVO defineQuestion.pl

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$JSCRIPT = <<EOF;
var ie = document.all?1:0;
var submitted = 0;
var miForma=document.forma;

function validaP() {
//Esta funcion se utiliza solo para validar la insercion al catalogo de
preguntas.

        if (document.forma.preguntas.value == "" &&
document.forma.newName.value == ""){
            alert("Es necesario seleccionar una pregunta del catalogo o
insertar una nueva");
            document.forma.newName.focus();
        }else if(document.forma.subfunciones.value == ""){

```

```
        alert("Es necesario seleccionar una subfuncion a la cual relacionar
la pregunta");
    }else if(document.forma.factor.es.value == ""){
        alert("Es necesario seleccionar un factor al cual relacionar la
pregunta");
    }else{
        document.forma.submit();
    }
}
function guardaSelect(){
//    var longitud=document.forma.destQAL.length;
    var i=0;
    var ids="";
    while (i < longitud){
//    ids=document.forma.destQAL.options[i].value+" "+ids ;
        i++;
    }
    document.forma.ids_preg.value=ids;
    document.forma.submit();
}

function regresa(){
    if( confirm('Desea ingresar otra pregunta?') ){
        history.go(-1);
    }else{}
}
EOF
;

#abre conexion
$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Seleccion de Auditoria y Operacion Audit-
Admin',
                -script=>$JSCRIPT);

$action = $query->param(accion);
$id_normal = $query->param(dato_id);
```

```

$Sid_audit = $query->param(audit);

$ids_preg = $query->param(ids_sub);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

#####
#Este es para checar el tipo de usuario
#####

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
#AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
#PASS$datosUser[0][5]<br>
#TIPO$datosUser[0][6]<br>
#GRUPO$datosUser[0][7]<br>

#usuario no es del tipo correcto
if ($id_normal eq "" ||$datosUser[0][6] ne "1" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl>vuelva a
intentar.</a>\n<br> <br> ";

    print $page_up.$contents.$page_down;
}else{
#####
#Si accion viene sin ningun contenido
#####
    if($accion == 1){
        #cuando es llamado por segunda ocasion para guardar los datos
        #
        # para recoger los id's de las subfunciones seleccionadas
        @id_preg = $query->param('preguntas');
#####PENDIENTE ESTA VERIFICAR QUE CUANDO SE INSERTA NUEVA PREGUNTA PRIMERO
SE GUARDE ESTA EN EL CATALOGO DE PREGUNTAS
### VERIFICAR QUE SOLO SE ESTA ENVIANDO EL MENSAJE DE QUE SE INSERTARON
EXISTOSAMENTE CUANDO ASI SEA
### QUE NO BORRE LAS PREGUNTAS ACTUALES , YA QUE NO SE BORRAN DESDE AQUI
SI NO DESDE LA VENTANA DE EVALUACION.

```

```

$hid_sub = $query->param(subfunciones);
$hid_factor = $query->param(factoros);
$newQuestion = $query->param(newQuestion);

if ($newQuestion ne ""){
    Pg::doQuery($conn,"select (max(id_pregunta)) FROM
cat_preguntas", \@maximo);
    $maximo[0][0] ++;
    Pg::doQuery($conn,"insert into cat_preguntas values
($maximo[0][0], '$newQuestion')");
}

#me quede en que hay que hacer la validacion para insertar mas
preguntas y
#que hay que insertar las preguntas que se seleccionaron para
posteriormente calificarlas
Pg::doQuery($conn,"select respuesta, id_grado from evaluacion
where id_auditoria=$hid_audit", \@evaluadas);
if( $evaluadas[0][0] > 0 || $evaluadas[0][1] > 0 ){
    print $page_up."La auditoria seleccionada contiene preguntas
evaluadas, no es posible agregar preguntas una vez que ha comenzado la
evaluacion \n";

    print $botones.$page_down;

}else{

    #para traer el nombre de la empresa
    Pg::doQuery($conn,"select nombre_emp,id_empresa from empresa
where id_empresa in (select id_empresa from auditoria where
id_auditoria=$hid_audit)", \@nombreEmp);
    #borra las subfunciones existentes
    Pg::doQuery($conn,"delete from evaluacion where
id_auditoria=$hid_audit and respuesta='' and id_grado='' ");
    #inserta las nuevas subfunciones
    ##hay que adaptar los cambios que hice para las preguntas
    $x = 0;
    while ($hid_preg[$x] > 0 ){
        #linea que verifica que no se inserten preguntas repetidas en
una auditoria.
        Pg::doQuery($conn,"select id_auditoria from evaluacion
where id_auditoria=$hid_audit and id_pregunta=$hid_preg[$x] and
id_sub=$hid_sub and id_factor=$hid_factor and
id_empresa=$nombreEmp[0][1]", \@pregExist);
        if (scalar(@pregExist)>0){

        }else{
            Pg::doQuery($conn,"insert into evaluacion
(id_auditoria,id_pregunta,id_sub,id_factor,id_empresa)
values($hid_audit,$hid_preg[$x],$hid_sub,$hid_factor,$nombreEmp[0][1])");
        }

        $x++;
    }

    $page_up = &hazOnload('regresa');
    print $page_up."La insercion de preguntas en la auditoria a la
empresa <b>\n $nombreEmp[0][0] \n</b> se realizo con exito aqui esta
preguntas <br>\n";

    print $botones.$page_down;
}

```

```

    }
}else{

    #trae las subfunciones y factores que se eligieron con anterioridad
    Pg::doQuery($conn,"select distinct id_sub from distrib_puntos
where id_auditoria=$id_audit;",\@subfun);
#para hacer una cadena que ayude a formar el query que solo traiga las
preguntas en el catalogo correspondientes a las subfunciones y factores
anteriormente seleccionados

    $limite = scalar(@subfun);
    $x = 0;
    $first = 0;
    while($x < $limite){
        if ($first == 0){
            $subfun[$x][0] =~ s/ //g;
            $cadenaSub="id_sub='$subfun[$x][0]';";
            $first = 1;
        }else{
            $cadenaSub="$cadenaSub or id_sub='$subfun[$x][0]';";
        }
        $x ++;
    }

    Pg::doQuery($conn,"select distinct id_factor from distrib_puntos
where id_auditoria=$id_audit;",\@factores);

    $limite = scalar(@factores);
    $x = 0;
    $first = 0;
    while($x < $limite){
        if ($first == 0){
            $factores[$x][0] =~ s/ //g;
            $cadenaFact="id_factor='$factores[$x][0]';";
            $first = 1;
        }else{
            $cadenaFact="$cadenaFact or id_factor='$factores[$x][0]';";
        }
        $x ++;
    }

    Pg::doQuery($conn,"select id_pregunta, descripcion from
cat_preguntas where id_pregunta not in (select id_pregunta from
evaluacion where id_auditoria=$id_audit)",\@catPreguntas);

#    print "select id_pregunta, descripcion from cat_preguntas where
id_pregunta not in (select id_pregunta from evaluacion where
id_auditoria=$id_audit)";

#####
#####
# Trae todas las preguntas y solo las subfunciones y factores elegidos,
es entonces #
# cuando el usuario define que una pregunta pertenece a una subfuncion y
factor #
#####
#####
#    Pg::doQuery($conn,"select id_pregunta, descripcion from
cat_preguntas where id_pregunta in (select id_pregunta from evaluacion
where id_auditoria=$id_audit)",\@lasPreguntas);

```

```

$x = 0;
$num_preguntas = @catPreguntas ;
if ($num_preguntas > 5){
    $num = 5;
}else{
    $num = $num_preguntas;
    if($num == 0){
        $num = scalar(@catPreguntas);
    }
}
if($num > 0){
    Pg::doQuery($conn,"select nombre,id_sub FROM cat_subfunciones
where id_sub in(select id_sub from distrib_puntos where
id_auditoria=$id_audit)order by nombre",\@catSubfunciones);
    Pg::doQuery($conn,"select nombre,id_factor FROM cat_factores
where id_factor in (select id_factor from distrib_puntos where
id_auditoria=$id_audit)order by nombre",\@catFactores);

    $contents_up = "
- MODULO DE OPERACIONES CON AUDITORIAS -Definicion de
Preguntas - AUDIT-ADMIN - \n
<p>\n
<form name=\"forma\" method=post>\n
<table><tr>\n
<td valign=\"middle\" align=\"center\">\n
Agregue las Preguntas que requiera su Auditoria y
relacionelas a un factor-subfunción:<br> \n
<SELECT NAME=\"preguntas\" size=\"5\" width=\"300px\"
style=\"width:300px;\" multiple><br>\n";
    print $page_up.$contents_up;

    while ( $catPreguntas[$x][0] ne "" ) {
        print "<OPTION
value=$catPreguntas[$x][0]>$catPreguntas[$x][1] \n";
        $x++;
    }
    print "</SELECT></td></tr>";

    $contents2 = "<tr><td valign=\"middle\" align=\"center\"><br>\n
</td></tr><tr><td valign=\"middle\" align=\"center\">\n";
    print $contents2;

    print "</td></tr>\n
<tr><td>OTRA:<input type=\"text\" name=\"newQuestion\"
size=\"70\"></td></tr>
</table>\n";

    print "<table><tr><td>Subfunciones<br>\n
<SELECT NAME=\"subfunciones\" size=\"3\" >";

    $x = 0;
    while ( $catSubfunciones[$x][0] ne "" ) {
#imprime las subfunciones existentes en el catalogo
        $catSubfunciones[$x][1] =~ s/ //g;

```

```

        print "<OPTION
value=\"\$catSubfunciones[$x][1]\">\"$catSubfunciones[$x][0]\"";
        $x ++;
    }
    print"</SELECT></td>
        <td>Factores<br>
<SELECT NAME=\"factores\" size=\"3\">";
    $x = 0;
    while ( $catFactores[$x][0] ne " " ) {
#imprime los factores existentes en el catalogo
        $catFactores[$x][1] =~ s/ //g;
        print "<OPTION
value=\"\$catFactores[$x][1]\">\"$catFactores[$x][0]\"";
        $x ++;
    }
    print"
        </SELECT>
</td></tr></table>\n
        <p><div align=center><input type=\"button\" value=\"Aceptar\"
Onclick=\"validaP()\"></div></p>

        <input type=\"hidden\" name=\"dato_id\"
value=\"\$id_normal\">\n
        <input type=\"hidden\" name=\"audit\" value=\"\$id_audit\">\n
        <input type=\"hidden\" name=\"accion\" value=1>\n
        </form>";
        print $botones.$page_down;
    }else {
#cuando no existen auditorias creadas.
        print $page_up;
        print "<p align=center>No hay auditorias existentes<br>0 que
cumplan con los contenidos necesarios para realizar este paso,Verifique
haber llevado a cabo los pasos anteriores para la realizacion de
auditorias o bien \n
        Requiere crear una auditoria previamente</p>";
        print $empresas[$x][0];
        print $botones.$page_down;
    }
}
}
#cierre del else
exit;

```

CODIGO DEL ARCHIVO defineQuestionA.pl

```

#!/usr/bin/perl -U
#####
#Proyecto de Tesis Auditoria Administrativa
#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

```

```
#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$JSCRIPT = <<EOF;
var destQAL = document.editQALForm.destQAL;
var ie = document.all?1:0;
var submitted = 0;

function validaP() {
//Esta funcion se utiliza solo para validar la insercion al catalogo de
preguntas.

    var miForma = document.editQALForm;

    if (miForma.preguntas.value == ""){
        alert("Es necesario seleccionar una pregunta del catalogo o
insertar una nueva");
        miForma.newName.focus();

    }else if(miForma.subfuncion.value == ""){

        alert("Es necesario seleccionar una subfuncion a la cual relacionar
la pregunta");

    }else if(miForma.factor.value == ""){

        alert("Es necesario seleccionar un factor al cual relacionar la
pregunta");
    }else{

        document.miForma.submit();

    }
}

function guardaSelect(){
    var miForma = document.editQALForm;
    var longitud=miForma.destQAL.length;
    var i=0;
    var ids="";

    while (i < longitud){

        ids=miForma.destQAL.options[i].value+":"+ids ;

        i++;

    }
    miForma.ids_preg.value=ids;

    miForma.submit();

}
EOF
;
```

```

#abre conexion
$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Seleccion de Auditoria y Operacion Audit-
Admin',
                -script=>$JSCRIPT);

$accion = $query->param(accion);

$id_normal = $query->param(dato_id);

$id_audit = $query->param(audit);

$sids_preg = $query->param(ids_sub);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

#####
#Este es para checar el tipo de usuario
#####

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre:$datosUser[0][1]
#   AP$datosUser[0][2]<br>
#AM$datosUser[0][3]<br>
#   LOG$datosUser[0][4]<br>
#PASS$datosUser[0][5]<br>
#TIPO$datosUser[0][6]<br>
#GRUPO$datosUser[0][7]<br>

#usuario no es del tipo correcto
if ($id_normal eq "" || $datosUser[0][6] ne "1" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el

```

```

administrador del sistema o <a href=valida.pl>vuelva a
intentar.</a>\n<br> <br> ";

    print $page_up.$contents.$page_down;
}else{
#####
#Si accion viene sin ningun contenido
#####
    if($accion == 1){
        #cuando es llamado por segunda ocasion para guardar los datos
        #
        # para recoger los id's de las subfunciones seleccionadas
        $sid_preg = $query->param(preguntas);
        $sid_sub = $query->param(subfunciones);
        $sid_factor = $query->param(factores);
        $newQuestion = $query->param(newQuestion);

        if ($newQuestion ne ""){
            Pg::doQuery($conn,"select (max(id_pregunta)) FROM
cat_preguntas",\@maximo);
            $maximo[0][0] ++;
            Pg::doQuery($conn,"insert into cat_preguntas values
($maximo[0][0], '$newQuestion')");
        }

        #me quede en que hay que hacer la validacion para insertar mas
preguntas y
        #que hay que insertar las preguntas que se seleccionaron para
posteriormente calificarlas
        Pg::doQuery($conn,"select respuesta, id_grado from evaluacion
where id_auditoria=$sid_audit",\@evaluadas);
        if( $evaluadas[0][0] > 0 || $evaluadas[0][1] > 0 ){
            print $page_up."La auditoria seleccionada contiene preguntas
evaluadas, no es posible agregar preguntas una vez que ha comenzado la
evaluacion \n";

            print $botones.$page_down;

        }else{

            #borra las subfunciones existentes
            Pg::doQuery($conn,"delete from evaluacion where
id_auditoria=$sid_audit and respuesta='' and id_grado='' ");
            #inserta las nuevas subfunciones
            ##hay que adaptar los cambios que hice para las preguntas
            $x = 0;
            while ($x < $longitud ){
                Pg::doQuery($conn,"insert into evaluacion
(id_auditoria,id_pregunta) values($sid_audit,$sids_preg[$x])");

                $x++;
            }

            #para traer el nombre de la empresa
            Pg::doQuery($conn,"select nombre_emp from empresa where
id_empresa in (select id_empresa from auditoria where
id_auditoria=$sid_audit)",\@nombreEmp);

            print $page_up."La insercion de preguntas en la auditoria a la
empresa <b>\ " $nombreEmp[0][0] \ "</b> se realizo con exito<br>\n";

            print $botones.$page_down;

```

```

    }
}else{

    #trae las subfunciones y factores que se eligieron con anterioridad
    Pg::doQuery($conn,"select distinct id_sub from distrib_puntos
where id_auditoria=$id_audit;","\@subfun);
#para hacer una cadena que ayude a formar el query que solo traiga las
preguntas en el catalogo correspondientes a las subfunciones y factores
anteriormente seleccionados

    $limite = scalar(@subfun);
    $x = 0;
    $first = 0;
    while($x < $limite){
        if ($first == 0){
            $subfun[$x][0] =~ s/ //g;
            $cadenaSub="id_sub='$subfun[$x][0]';";
            $first = 1;
        }else{
            $cadenaSub="$cadenaSub or id_sub='$subfun[$x][0]';";
        }
        $x ++;
    }

    Pg::doQuery($conn,"select distinct id_factor from distrib_puntos
where id_auditoria=$id_audit;","\@factores);

    $limite = scalar(@factores);
    $x = 0;
    $first = 0;
    while($x < $limite){
        if ($first == 0){
            $factores[$x][0] =~ s/ //g;
            $cadenaFact="id_factor='$factores[$x][0]';";
            $first = 1;
        }else{
            $cadenaFact="$cadenaFact or id_factor='$factores[$x][0]';";
        }
        $x ++;
    }

    Pg::doQuery($conn,"select id_pregunta, descripcion from
cat_preguntas where id_pregunta not in (select id_pregunta from
evaluacion where id_auditoria=$id_audit)",\@catPreguntas);

#    print "select id_pregunta, descripcion from cat_preguntas where
id_pregunta not in (select id_pregunta from evaluacion where
id_auditoria=$id_audit)";

#####
#####
# Trae todas las preguntas y solo las subfunciones y factores elegidos,
es entonces #
# cuando el usuario define que una pregunta pertenece a una subfuncion y
factor #
#####
#####
#    Pg::doQuery($conn,"select id_pregunta, descripcion from
cat_preguntas where id_pregunta in (select id_pregunta from evaluacion
where id_auditoria=$id_audit)",\@lasPreguntas);

```

```

$x = 0;
$num_preguntas = @catPreguntas ;
if ($num_preguntas > 5){
    $num = 5;
}else{
    $num = $num_preguntas;
    if($num == 0){
        $num = scalar(@catPreguntas);
    }
}
if($num > 0){
    Pg::doQuery($conn,"select nombre,id_sub FROM cat_subfunciones
where id_sub in(select id_sub from distrib_puntos where
id_auditoria=$id_audit)order by nombre",\@catSubfunciones);
    Pg::doQuery($conn,"select nombre,id_factor FROM cat_factores
where id_factor in (select id_factor from distrib_puntos where
id_auditoria=$id_audit)order by nombre",\@catFactores);

    $contents_up = "
- MODULO DE OPERACIONES CON AUDITORIAS -Definicion de
Preguntas - AUDIT-ADMIN - \n
<p>\n
<form method=post name=\"editQALForm\">\n
<table><tr>\n
<td valign=\"middle\" align=\"center\">;

    $contents2 = "<tr><td valign=\"middle\" align=\"center\"><br>\n
</td></tr><tr><td valign=\"middle\" align=\"center\">\n";

    print $contents2;

#    print "Preguntas elegidas:<br><SELECT NAME=\"destQAL\"
size=\"5\" width=\"300px\" style=\"width:300px;\"><br>\n";
#    $x = 0;

#    while ( $lasPreguntas[$x][0] ne " " ) {
#        print "<OPTION
value=$lasPreguntas[$x][0]>$lasPreguntas[$x][1] \n";
#        $x++;
#    }

#    print "</select></td></tr></table>\n";
#    print "</td></tr>\n
<tr><td>OTRA:<input type=\"text\" name=\"newQuestion\"
size=\"70\"></td></tr>
</table>\n";

    print "<table><tr><td>Subfunciones<br>\n
<SELECT NAME=\"subfunciones\" size=\"3\"
onChange=daleValue(this.options,'sub')>";

    $x = 0;
    while ( $catSubfunciones[$x][0] ne " " ) {
#imprime las subfunciones existentes en el catalogo
        $catSubfunciones[$x][1] =~ s/ //g;
        print "<OPTION
value=\"$catSubfunciones[$x][1]\">$catSubfunciones[$x][0]\n";

        $x ++;

```

```

    }

    print "</SELECT></td>
      <td>Factores<br>
      <SELECT NAME=\"factores\" size=\"3\"
onChange=daleValue(this.options,'fact')>";

    $x = 0;

    while ( $catFactores[$x][0] ne " " ) {
#imprime los factores existentes en el catalogo
    $catFactores[$x][1] =~ s/ //g;
    print "<OPTION
value=\" $catFactores[$x][1]\">$catFactores[$x][0]\n";

    $x ++;
}

    print "
      </SELECT>
    </td></tr></table>\n
    <p><div align=center><input type=\"button\" value=\"Aceptar\"
Onclick=\"validaP()\"></div></p>

      <input type=\"hidden\" name=\"dato_id\"
value=\" $id_normal\">\n
      <input type=\"hidden\" name=\"audit\" value=\" $id_audit\">\n
      <input type=\"hidden\" name=\"accion\" value=1>\n
    </form>";

    print $botones.$page_down;
  }else {
#cuando no existen auditorias creadas.
    print $page_up;
    print "<p align=center>No hay auditorias existentes<br>0 que
cumplan con los contenidos necesarios para realizar este paso,Verifique
haber llevado a cabo los pasos anteriores para la realizacion de
auditorias o bien \n
    Requiere crear una auditoria previamente</p>";
    print $empresas[$x][0];
    print $botones.$page_down;
  }
}
}
#cierre del else

exit;

```

CODIGO DEL ARCHIVO defineSub.pl

```
#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$JSCRIPT = <<EOF;
var destQAL = document.editQALForm.destQAL;
var ie = document.all?1:0;
var submitted = 0;

function AddToTheList()
{
    var fromAB = document.editQALForm.fromAB;
    var destQAL = document.editQALForm.destQAL;
    var selectedValue = "";
    var selectedText = "";
    if ( fromAB.selectedIndex < 0 )
    {
        alert("Por favor, primero seleccione un elemento de la lista");
        if (fromAB.options[0] != null)
            fromAB.options[0].selected = true;
        fromAB.focus();
        return false;
    }
    else
    {
        selectedValue = fromAB.options[fromAB.selectedIndex].value;
        selectedText = fromAB.options[fromAB.selectedIndex].text;
        destQAL.options[destQAL.length] = new
Option(selectedText,selectedValue);
        fromAB.options[fromAB.selectedIndex] = null;
    }
}

function RemoveFromTheList()
{
    var fromAB = document.editQALForm.fromAB;
    var destQAL= document.editQALForm.destQAL;
    var selectedValue = "";
    var selectedText = "";
    if ( destQAL.selectedIndex < 0 )
    {
        alert("Por favor, primero seleccione un elemento de la lista");
        if (destQAL.options[0] != null)
            destQAL.options[0].selected = true;
        destQAL.focus();
        return false;
    }
}

```

```

    }
    else
    {
        selectedValue = destQAL.options[destQAL.selectedIndex].value;
        selectedText = destQAL.options[destQAL.selectedIndex].text;
        fromAB.options[fromAB.length] = new
Option(selectedText,selectedValue);
        destQAL.options[destQAL.selectedIndex] = null;
    }
}
function disapleButtons()
{
    var fromAB = document.editQALForm.fromAB;
    var destQAL= document.editQALForm.destQAL;
    if (fromAB.options.length > 0 )
    {
        if (destQAL.options.length < 5)
            document.editQALForm.AddQ.disabled = 0;
        else
            document.editQALForm.AddQ.disabled = 1;
    }
    else
        document.editQALForm.AddQ.disabled = 1;
    if (destQAL.options.length > 0)
        document.editQALForm.RemoveQ.disabled = 0;
    else
        document.editQALForm.RemoveQ.disabled = 1;
}
function putItTogether()
{
    var destQAL= document.editQALForm.destQAL;
    var fullList = "";
    for(var i = 0; i < destQAL.length; i++)
    {
        fullList += (destQAL.options[i].text + "::");
    }
    fullList = fullList.substring(0, fullList.length-2);
    document.editQALForm.quickaddr.value = fullList;
    document.editQALForm.submit();
}
disapleButtons();

function guardaSelect(){
    var miForma = document.editQALForm;
    var longitud=miForma.destQAL.length;
    var i=0;
    var ids="";

    while (i < longitud){

        ids=miForma.destQAL.options[i].value+"::"+ids ;

        i++;

    }
    miForma.ids_sub.value=ids;

    miForma.submit();

}
EOF

```

```

;

#abre conexion
$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Seleccion de Auditoria y Operacion Audit-
Admin',
                -script=>$JSCRIPT);

$accion = $query->param(accion);

$id_normal = $query->param(dato_id);

$id_audit = $query->param(audit);

$ids_sub = $query->param(ids_sub);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

#####
#Este es para checar el tipo de usuario
#####

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

# si no llego el id o llego pero no es un usuario del tipo
administrador, se desecha
# el id es:$datosUser[0][0]
# el nombre:$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

#usuario no es del tipo correcto
if ($id_normal eq "" ||$datosUser[0][6] ne "1" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el

```

```

administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br> <br> ";

    print $page_up.$contents.$page_down;
}else{

#####
# Si accion viene sin ningun contenido #
#####

    if($accion == 1){
# if ($id_audit eq ""){
#     print $page_up."Los datos de la auditoria no llegaron completos,
verifique con el administrador del sistema o vuelva a intentar";
#     print $botones.$page_down;
# }else{

# para recoger los id's de las subfunciones seleccionadas
    $ids_sub = $query->param(ids_sub);
    $cadenita = $ids_sub;
    @ids_sub = split(/:/, $ids_sub);
    $longitud = @ids_sub;

# verifica si hay subfunciones con pesos asignados
    Pg::doQuery($conn,"select sum(porcentaje) from ponderacion_sub
where id_auditoria=$id_audit",\@porcentajes);
    if( $porcentajes[0][0] > 0 ){
        print $page_up."Las subfunciones seleccionadas contienen pesos
asignados, es necesario vaciar los porcentajes para editar nuevamente las
subfunciones \n";

        print $botones.$page_down;

    }else{

#borra las subfunciones existentes
    Pg::doQuery($conn,"delete from ponderacion_sub where
id_auditoria=$id_audit",\@porcentajes);
#inserta las nuevas subfunciones
    $x = 0;
    while ($x < $longitud ){
        Pg::doQuery($conn,"insert into ponderacion_sub
(id_sub,id_auditoria) values('$ids_sub[$x]', $id_audit)");
#para verificar si se realizo la insercion sin errores
        $x++;
    }

    #para traer el nombre de la empresa
    Pg::doQuery($conn,"select nombre_emp from empresa where
id_empresa in (select id_empresa from auditoria where
id_auditoria=$id_audit)",\@nombreEmp);

    print $page_up."La insercion de subfunciones en la empresa
<b>\ " $nombreEmp[0][0] \ "</b> se realizo con exito<br>\n";

    print $botones.$page_down;
}

}else{

#trae las subfunciones existentes en los catalogos del sistema

```

```

Pg::doQuery($conn,"select id_sub,nombre from cat_subfunciones where
id_sub not in (select id_sub from ponderacion_sub where
id_auditoria=$id_audit) order by nombre",\@subfuncionesCat);

#trae las subfunciones existentes en la auditoria
Pg::doQuery($conn,"select id_sub,nombre from cat_subfunciones where
id_sub in (select id_sub from ponderacion_sub where
id_auditoria=$id_audit) order by nombre",\@subfunciones);

$x = 0;
$num_subfunciones = @subfuncionesCat ;
if ($num_subfunciones > 10){
    $num = 10;
}else{
    $num = $num_subfunciones;
    if($num == 0){
        $num = scalar(@subfunciones);
    }
}
if($num > 0){
    $contents_up = "
- MODULO DE OPERACIONES CON AUDITORIAS - AUDIT-ADMIN \n
<p>\n
<form method=post name=\"editQALForm\">\n
<table><tr>\n
    <td colspan=2>                <input type=\"button\"
name=\"listaPrint\" value=\"Lista de Impresion de Subfunciones\"
onClick=\"window.open('imprime.pl?dato_id=$id_normal&tipo=sub&id_auditori
a=$id_audit','variable','width=600,height=450,scrollbars=yes,toolbar=yes'
)\> </td></tr>
    <tr>
    <td valign=\"middle\" align=\"center\">\n

    Agregue las Subfunciones que requiera su Auditoria:<br> \n
    <SELECT NAME=\"fromAB\" size=\"10\" width=\"300px\"
style=\"width:300px;\><br>\n";
    print $page_up.$contents_up;

    while ( $subfuncionesCat[$x][0] ne "" ) {

        print "<OPTION
value=$subfuncionesCat[$x][0]>$subfuncionesCat[$x][1] \n";

        $x++;
    }

    $contents_down = "</SELECT></td>";
    print $contents_down;

    $contents2 = "<td valign=\"middle\" align=\"center\"><br>\n
    <input class=\"sbtttn\" style=\"width: 75px;\>\" type=\"button\"
value=\"Agregar >\" name=\"AddQ\" onClick=\"AddToTheList();\"><p>\n
    <input class=\"sbtttn\" style=\"width: 75px;\>\" type=\"button\"
value=\"&laquo; Quitar\" name=\"RemoveQ\"
onClick=\"RemoveFromTheList();\">\n
    </td><td valign=\"middle\" align=\"center\">\n";

    print $contents2;

    print "Subfunciones elegidas:<br><SELECT NAME=\"destQAL\"
size=\"10\" width=\"300px\" style=\"width:300px;\><br>\n";

```

```

    $x = 0;

    while ( $subfunciones[$x][0] ne " " ) {
        print "<OPTION
value=$subfunciones[$x][0]>$subfunciones[$x][1] \n";
        $x++;
    }

    $contents2_down = "</select></td></tr></table>\n
    <input class=\"sbttt\" style=\"width: 75px;\" type=\"button\"
value=\"Guardar\" name=\"Guardar\" onClick=\"guardaSelect();\n\">
    <input type=\"hidden\" name=\"dato_id\"
value=\"${id_normal}\">\n
    <input type=\"hidden\" name=\"audit\" value=\"${id_audit}\">\n
    <input type=\"hidden\" name=\"ids_sub\" value=\"\">\n
    <input type=\"hidden\" name=\"accion\" value=1>\n
    </form>\n";

    print $contents2_down.$botones.$page_down;
} else {
#cuando no existen auditorias creadas.
    print $page_up;
    print "No hay auditorias existentes<br>\n
    Requiere crear una nueva auditoria previamente";
    print $empresas[$x][0];
    print $botones.$page_down;
}
}
}
#cierre del else

exit;

```

CODIGO DEL ARCHIVO defineWeight.pl

```

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

#abre conexion
$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

$JSCRIPT = <<EOF;

```

```

var nombreInput;
var total = 0;
var ids,value1;

function doString(nombreInput,valor){
    ids = nombreInput+":"+ids;
    value1 = valor.toString()+":"+value1;
    valor++;
    total = valor + total;
    total = total -1 ;
}

function guardar(){
    miForma = document.formal;
    var cadena = miForma.auxiliar.value;
    var separator = ":";
    var arregloIds = cadena.split(separator);
    var limite = arregloIds.length -1;
    var losContenidos = "";
    for (var i = 0; i < limite; i++){
        elInput = "miForma"+"."+i+arregloIds[i]+".value;";
        numerito = parseInt (eval(elInput));
        total = total + numerito;
        losContenidos += numerito + ":";
    }
    if (total < 100 && total > 0){
        alert("La suma de las subfunciones debe dar 100% o bien 0 si se
desea mantener en blanco");
        total = 0;
        losContenidos = "";
    }else if(eval(total) > 100){
        alert("La suma de las subfunciones no puede ser mayor al 100%");
        total = 0;
        losContenidos = "";
    }else{
        losContenidos = losContenidos+"0";
        miForma.porcentajes.value = losContenidos;
        cadena = cadena+"0";
        miForma.ids_sub.value = cadena;
        //alert(losContenidos);
        miForma.submit();
    }
}

EOF
;

print start_html(-title=>'Seleccion de Auditoria y Operacion Audit-
Admin',
                -script=>$JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);
$id_audit = $query->param(audit);
$sids_sub = $query->param(ids_sub);

```

```

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

#####
#Este es para checar el tipo de usuario
#####

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre:$datosUser[0][1]
# AP$datosUser[0][2]<br>
#AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
#PASS$datosUser[0][5]<br>
#TIPO$datosUser[0][6]<br>
#GRUPO$datosUser[0][7]<br>

#usuario no es del tipo correcto
if ($id_normal eq "" ||$datosUser[0][6] ne "1" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br> <br> ";

    print $page_up.$contents.$page_down;
}else{
#####
#Si accion viene sin ningun contenido
#####
    if($accion == 1){
# para recoger los id's de las subfunciones seleccionadas
        $ids_sub = $query->param(ids_sub);
        $cadenita = $ids_sub;
        @ids_sub = split(/:/, $ids_sub);
        $lengIds = @ids_sub;
        $porcentajes = $query->param(porcentajes);
# $porcientos=$porcentajes;
        @porcentajes = split(/:/, $porcentajes);
        $lengPercent = @porcentajes;
        $lengPercent = $lengPercent-1;

#####
#verifica si hay subfunciones con pesos asignados
#####

```

```

Pg::doQuery($conn,"select puntos from distrib_puntos where
id_auditoria=$id_audit",\@puntos);

#####
#cuando existen puntos asignados con anterioridad
#####
if( $puntos[0][0] ne ""){
    print $page_up."La auditoria seleccionada cuenta ya con puntos
asignados, que son el comienzo de una evaluacion, no es posible insertar
nuevos pesos cuando se ha realizado la distribucion de puntos.<br> \n";

    print $botones.$page_down;

}

}else{

#borra las pesos existentes
Pg::doQuery($conn,"update ponderacion_sub set porcentaje=0 where
id_auditoria=$id_audit");
#inserta las nuevas subfunciones
$x = 0;
while ($x < $lengPercent ){

    Pg::doQuery($conn,"update ponderacion_sub set
porcentaje=$porcentajes[$x] where id_sub='$ids_sub[$x]' and
id_auditoria=$id_audit");

#para verificar si se realizo la insercion sin errores
    $x++;
}

#para traer el nombre de la empresa
Pg::doQuery($conn,"select nombre_emp from empresa where
id_empresa in (select id_empresa from auditoria where
id_auditoria=$id_audit)",\@nombreEmp);

    print $page_up."La insercion de pesos en la empresa <b>\n
$nombreEmp[0][0] \n"</b> se realizo con exito<br>\n";

    print $botones.$page_down;
}

}

}else{
#####
#AQUI HAY QUE PONER LAS SUBFUNCIONES DE LA AUDITORIA
#Y DESPUES DAR PARA PONER LOS PORCENTAJES
#####

#trae las subfunciones existentes en la auditoria
Pg::doQuery($conn,"select id_sub,nombre from cat_subfunciones where
id_sub in (select id_sub from ponderacion_sub where
id_auditoria=$id_audit)",\@subfunciones);

    $x = 0;
$num_subfunciones=@subfunciones ;
if ($num_subfunciones > 10){
    $num = 10;
}else{
    $num = $num_subfunciones;
}
if($num > 0){
    $contents_up = "
- MODULO DE OPERACIONES CON AUDITORIAS - AUDIT-ADMIN \n

```

```

        <p> \n
        <form method=post name=\"formal\" action=\"defineWeight.pl\">
\n
        <table><tr>\n";

    print $page_up.$contents_up;

    print "<tr><td colspan=2>Teclee el peso de cada subfuncion
asignada:</td></tr> \n";

    $x = 0;

    $losIds;

    while ( $subfunciones[$x][0] ne "" ) {

        $subfunciones[$x][0] =~s/ //g;

#        Pg::doQuery($conn,"select porcentaje from ponderacion_sub where
id_sub=$subfunciones[$x][0] id_auditoria=$id_audit)","@\percentExist);

        print "<td> $subfunciones[$x][1] </td>
<td> <input type=\"text\" maxlength=\"6\"
name=\"i$subfunciones[$x][0]\"></td></tr> \n";
        $vari = $subfunciones[$x][0];

        $losIds = "$vari:$losIds";

        $x ++;
    }

    $numerito = scalar(@subfunciones);
    $contents2_down = "</table><br>\n
<input class=\"sbttt\" style=\"width: 75px;\" type=\"button\"
value=\"Guardar\" name=\"Guardar\" onClick=\"guardar();\n\">
<input type=\"hidden\" name=\"dato_id\" value=\"$id_normal\">\n
<input type=\"hidden\" name=\"audit\" value=\"$id_audit\">\n
<input type=\"hidden\" name=\"ids_sub\" value=\"\">\n
<input type=\"hidden\" name=\"auxiliar\" value=\"$losIds\">\n
<input type=\"hidden\" name=\"porcentajes\" value=\"\">\n
<input type=\"hidden\" name=\"accion\" value=1>\n
</form>\n";

    print $contents2_down.$botones.$page_down;

}else {
#cuando no existen auditorias creadas.
    print $page_up;

    print "La auditoria seleccionada no contiene subfunciones
asignadas<br>\n
Requiere asociar subfunciones a una auditoria con anterioridad
para realizar esta operacion \n";

    print $empresas[$x][0];

    print $botones.$page_down;
}
}

```

```

}
#cierre del else

exit;

CODIGO DEL ARCHIVO delUser.pl

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print $query->start_html;

$action = $query->param(accion);

$id_normal = $query->param(dato_id);

$usuario = $query->param(id);

#variables que crean el formato de la pagina que se despliega
$page_up = " <html>\n
  <body bgcolor=white background=\"../Tesis/Imagenes/bglogito.jpg\">\n
  <table align=center><tr><td>\n";
$page_down = "
</td></tr></table>\n
</body>\n
</html>\n";

#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal", \@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>

```

```

# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "3"){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    llego $id_normal<br>
    enc $id_encriptado<br>
tipo $datosUser[0][6]";

    print $page_up.$contents.$page_down;
}else{

    Pg::doQuery($conn,"delete from usuarios where id_usuario
=$usuario");

    $contents="El usuario se elimino con exito <br>
<form><input type=\"button\" value=\"cerrar\"
onclick=\"window.close()\" ></form>";

    print $page_up.$contents.$page_down;

}#cierre del else

exit;

```

CODIGO DEL ARCHIVO evalQuestion.pl

```

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$JSCRIPT = <<EOF;
var ie = document.all?1:0;
var submitted = 0;
var miForma=document.forma;

function validaP() {

    document.forma.action='evalQuestion.pl';
    document.forma.submit();

}

```

```
function guardaSelect(){
//   var longitud=document.forma.destQAL.length;
  var i=0;
  var ids="";

  while (i < longitud){
//   ids=document.forma.destQAL.options[i].value+":"+ids ;
    i++;
  }
  document.forma.ids_preg.value=ids;

  document.forma.submit();
}

function regresa(){
  if( confirm('Desea ingresar otra pregunta?') ){
    history.go(-1);
  }else{}
}
EOF
;

#abre conexion
$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Evaluacion de auditorias',
                 -script=>$JSCRIPT);

$action = $query->param(accion);

$id_normal = $query->param(dato_id);

$id_audit = $query->param(audit);

@variables= &prueba();

#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
```

```

#terminan los formatos de la pagina

#####
#Este es para checar el tipo de usuario
#####

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

#usuario no es del tipo correcto
if ($id_normal eq "" ||$datosUser[0][6] ne "1" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl>vuelva a
intentar.</a>\n<br> <br> ";

    print $page_up.$contents.$page_down;
}else{

    if($accion == 1){
#####
#cuando es llamado por segunda ocasion para guardar los datos
#####
        $i=1;
        $numDatos=$query->param("numDatos");
        #recibiendo las calificaciones
        print $page_up;

        while($i <= $numDatos){
            $dato=$query->param("pregResp".$i)."-".$query->param("obs".$i);
            if (length($dato)>0){
                push(@lasPregResp,$dato);
            }else{print "No se selecciono ni se agrego ningun dato en el
campo numero $i";}

            $i++;
        }

        $limite=scalar(@lasPregResp);
        #print scalar(@lasPregResp);
        #para guardar las calificaciones recibidas
        $i=0;

        while($i < $limite){

            @registro=split(/-/, $lasPregResp[$i]);

```

```

#el registro trae el id de la subfuncion-idfactor-idpregunta-
idgrado-idEmpresa-observaciones
# print " se recibio id de la subfuncion $registro[0]-idfactor
$registro[1]-idpregunta $registro[2]-idgrado $registro[3]-observaciones
$registro[5]<p>" ;
if ($registro[3] == "666"){#cuando se selecciono eliminar
Pg::doQuery($conn,"delete from evaluacion where
id_sub=$registro[0] and id_factor=$registro[1] and
id_pregunta=$registro[2] and id_empresa=$registro[4] and
id_auditoria=$id_audit;");
# print "delete from evaluacion where id_sub=$registro[0] and
id_factor=$registro[1] and id_pregunta=$registro[2] and
id_empresa=$registro[4] and id_auditoria=$id_audit;";
}elseif ( length($registro[3])>0 || length($registro [5])>0){
Pg::doQuery($conn," update evaluacion set
id_grado=$registro[3],observaciones='$registro[5]' where
id_sub=$registro[0] and id_factor=$registro[1] and
id_pregunta=$registro[2] and id_empresa=$registro[4] and
id_auditoria=$id_audit;");

# print "update evaluacion set
id_grado=$registro[3],observaciones='$registro[5]' where
id_sub=$registro[0] and id_factor=$registro[1] and
id_pregunta=$registro[2] and id_empresa=$registro[4] and
id_auditoria=$id_audit;";
}else{
# print "no hice nada $registro[6] <br>";
}

$i++;
}
#Pg::doQuery($conn," select
id_grado,apreciacion,cump_superior,cump_inferior from
grados;","\@losGrados);
$content_up = "
- MODULO DE OPERACIONES CON AUDITORIAS -Evaluacion de las
preguntas - AUDIT-ADMIN - \n
<p align="center">\n Los cambios se realizaron con
exito<br> \n";

print $content_up.$botones.$page_down;

}else{
#####
#####
#Si accion viene sin ningun contenido es la primera vez que se llama
a la evaluacion

#####
#####

#trae las subfunciones y factores que se eligieron con anterioridad
Pg::doQuery($conn," select
id_grado,id_auditoria,id_empresa,e.id_pregunta,p.descripcion,e.id_sub,cs.
nombre,e.id_factor,cf.nombre from evaluacion e,cat_preguntas
p,cat_subfunciones cs,cat_factores cf where p.id_pregunta=e.id_pregunta
and cs.id_sub=e.id_sub and cf.id_factor=e.id_factor and
id_auditoria=$id_audit order by e.id_sub,e.id_factor;","\@dataEval);

```

#para hacer una cadena que ayude a formar el query que solo traiga las preguntas en el catalogo correspondientes a las subfunciones y factores anteriormente seleccionados

```

    $sqlIdGrado;
    $sqlIdAuditoria=1;
    $sqlIdEmpresa=2;
    $sqlIdPregunta=3;
    $sqlDescrip=4;
    $sqlIdSub=5;
    $sqlSubNombre=6;
    $sqlIdFactor=7;
    $sqlFacNombre=8;

    Pg::doQuery($conn," select
id_grado,apreciacion,cump_superior,cump_inferior from
grados;" , \@losGrados);

    $gradId=0;
    $gradAprec=1;
    $gradSup=2;
    $gradInf=3;

    $contents_up = "
        - MODULO DE OPERACIONES CON AUDITORIAS -Evaluacion de las
preguntas - AUDIT-ADMIN - \n
        <p>\n
        <form name=\"forma\" method=post>\n
        Seleccione la calificacion correspondiente a cada pregunta y
guarde<br> \n
        <table align='center' cellpadding=0 cellspacing=0>\n";

    print $page_up.$contents_up;

    $limite = scalar(@dataEval);
    $x = 0;
    $first = 0;

    while($x < $limite){
        $oscuro="#D1D1D1";
        $claro="#FFFFFF";
        if ($color eq $oscuro){
            $color=$claro;
        }else{
            $color=$oscuro;
        }
        if ($x == 0 || ($dataEval[$x][$sqlIdSub]!= $dataEval[$x-
1][$sqlIdSub]) ){
            print "<tr><th colspan=9><table cellpadding=0
cellspacing=0><tr><th>Subfuncion
$dataEval[$x][$sqlSubNombre]</th></tr></table> </th></tr> \n";
            if ($x == 0 || ($dataEval[$x][$sqlIdFactor]!= $dataEval[$x-
1][$sqlIdFactor]) ){
                print "<tr><th colspan=9><table cellpadding=0
cellspacing=0><tr><th>Factor
$dataEval[$x][$sqlFacNombre]</th></tr></table> </th></tr> \n";
            }
        }else{
            print "<tr> <td bgcolor=\" $color\"><font
size=\"2\">$dataEval[$x][$sqlDescrip]</font> </td>\n";
        }
    }

```

```

        $limGrados=scalar(@losGrados);
        $y=0;

        while($y < $limGrados){
            #el value trae el id de la subfuncion-idfactor-
            idpregunta-idgrado-empresa
            if (length($dataEval[$x][$sqlIdGrado])>0){
                if($dataEval[$x][$sqlIdGrado] ==
                $y){$chechado="checked";}else {$chechado="";}
            }

            print "<td bgcolor=\"\$color\"><input type=\"radio\"
            name=\"pregResp$x\" value=\"\$dataEval[$x][$sqlIdSub]-
            $dataEval[$x][$sqlIdFactor]-$dataEval[$x][$sqlIdPregunta]-
            $losGrados[$y][$gradId]-$dataEval[$x][$sqlIdEmpresa]\" \$chechado><font
            size=\"1\">\"$losGrados[$y][$gradAprec]</font></td>\n";

            $y++;
        }

        print "<td bgcolor=\"\$color\">Observaciones<br> <input
        type=\"text\" name=\"obs$x\" size=\"50\"></td>
        <td><font color=\"red\" size=\"1\">Eliminar</font><input
        type=\"radio\" name=\"pregResp$x\" value=\"\$dataEval[$x][$sqlIdSub]-
        $dataEval[$x][$sqlIdFactor]-$dataEval[$x][$sqlIdPregunta]-666-
        $dataEval[$x][$sqlIdEmpresa]\"></td>\n</tr>\n";

    }
    $x++;
}

print "</table>";

print"<p><div align=center><input type=\"button\"
value=\"Aceptar\" onclick=\"validaP()\"></div></p>
<input type=\"hidden\" name=\"dato_id\"
value=\"\$id_normal\">\n
<input type=\"hidden\" name=\"audit\" value=\"\$id_audit\">\n
<input type=\"hidden\" name=\"numDatos\" value=\"\$x\">\n
<input type=\"hidden\" name=\"accion\" value=1>\n
</form>";

print $botones.$page_down;

}#else{
#cuando no existen auditorias creadas.
# print $page_up;
# print "<p align=center>No hay auditorias existentes<br>0 que
cumplan con los contenidos necesarios para realizar este paso,Verifique
haber llevado a cabo los pasos anteriores para la realizacion de
auditorias o bien \n
# Requiere crear una auditoria previamente</p>";
# print $empresas[$x][0];
# print $botones.$page_down;
#}
}

#cierre del else

```

```
exit;
```

```
CODIGO DEL ARCHIVO  imprime.pl
```

```
#!/usr/bin/perl  -U
```

```
#####  
#Proyecto de Tesis Auditoria Administrativa
```

```
#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM
```

```
#This program is free software; you can redistribute it and/or  
#modify it under the terms of the GNU General Public License  
#as published by the Free Software Foundation; either version 2  
#of the License, or (at your option) any later version.  
#####
```

```
use Pg;  
use CGI qw(:standard);  
use pemes::inicializar;
```

```
#abre conexion  
$conn = Pg::connectdb("dbname=auditoria user=pardo");
```

```
$query = new CGI;
```

```
print $query->header;
```

```
print start_html(-title=>'Seleccion de Auditoria y Operacion Audit-  
Admin',  
                 -script=>$JSCRIPT);
```

```
$accion = $query->param(accion);
```

```
$id_normal = $query->param(dato_id);
```

```
$id_audit = $query->param(audit);
```

```
$tipo = $query->param(tipo);
```

```
@variables = &prueba();  
#variables que crean el formato de la pagina que se despliega  
$page_up = $variables[0];  
$page_down = $variables[1];
```

```
#####  
#Este es para checar el tipo de usuario  
#####
```

```
Pg::doQuery($conn,"select * from usuarios where id_usuario  
=$id_normal", \@datosUser);
```

```

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre$datosUser[0][1]
#   AP$datosUser[0][2]<br>
#AM$datosUser[0][3]<br>
#   LOG$datosUser[0][4]<br>
#PASS$datosUser[0][5]<br>
#TIPO$datosUser[0][6]<br>
#GRUPO$datosUser[0][7]<br>

#usuario no es del tipo correcto
if ($id_normal eq "" ||$datosUser[0][6] ne "1" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br> <br> ";

    print $page_up.$contents.$page_down;
}else{
#####
#Si accion viene sin ningun contenido
#####
    if($accion == 1){
#   if ($id_audit eq ""){
#       print $page_up."Los datos de la auditoria no llegaron completos,
verifique con el administrador del sistema o vuelva a intentar";
#   print $botones.$page_down;
#   }else{

# para recoger los id's de las subfunciones seleccionadas
    $ids_sub = $query->param(ids_sub);
    $cadenita = $ids_sub;
    @ids_sub = split(/:/, $ids_sub);
    $longitud = @ids_sub;

#verifica si hay subfunciones con pesos asignados
    Pg::doQuery($conn,"select sum(porcentaje) from ponderacion_sub
where id_auditoria=$id_audit",\@porcentajes);
    if( $porcentajes[0][0] > 0 ){
        print $page_up."Las subfunciones seleccionadas contienen pesos
asignados, es necesario vaciar los porcentajes para editar nuevamente las
subfunciones \n";

        print $botones.$page_down;

    }else{

#borra las subfunciones existentes
    Pg::doQuery($conn,"delete from ponderacion_sub where
id_auditoria=$id_audit",\@porcentajes);
#inserta las nuevas subfunciones
    $x = 0;
    while ($x < $longitud ){
        Pg::doQuery($conn,"insert into ponderacion_sub
(id_sub,id_auditoria) values('$ids_sub[$x]', $id_audit)");
#para verificar si se realizo la insercion sin errores
        $x++;
    }

    #para traer el nombre de la empresa

```

```

Pg::doQuery($conn,"select nombre_emp from empresa where
id_empresa in (select id_empresa from auditoria where
id_auditoria=$id_audit)"," \@nombreEmp);

    print $page_up."La insercion de subfunciones en la empresa
<b>\ " $nombreEmp[0][0] \ "</b> se realizo con exito<br>\n";

    print $page_down;
}

}else{

    #trae las subfunciones existentes en los catalogos del sistema
    if ($tipo eq 'sub'){
        Pg::doQuery($conn,"select id_sub,nombre from cat_subfunciones
order by nombre", \@Catalogo);
    }elseif($tipo eq 'fac'){
        Pg::doQuery($conn,"select id_factor,nombre from cat_factores
order by nombre", \@Catalogo);
    }

    $x = 0;
    $num = scalar(@Catalogo) ;

    if($num > 0){
        $contents_up = "
        - MODULO DE OPERACIONES CON AUDITORIAS - AUDIT-ADMIN \n
        <p>\n
        <form>\n
        <table border=1>
        <td valign=\ "middle\ " align=\ "center\ ">\n

        Agregue las Subfunciones que requiera su Auditoria:<br> \n
        <table
border=2><tr><td>Nombre</td><td>Marcar<br>Importantes</td><td>Ordenar</td
><td>\n</td><td>\n</td><td>\n</td></tr>\n";
        print $page_up.$contents_up;

        while ( $Catalogo[$x][0] ne " " ) {

            print
            "<tr><td>$Catalogo[$x][1]</td><td>\n</td><td>\n</td><td>\n</td></tr> \n";

            $x++;
        }

        $contents_down = "</table></form></td></tr></table>\n";;

        print $contents_down.$page_down;

    }else {
#cuando no existen auditorias creadas.
    print $page_up;
    print "No hay auditorias existentes<br>\n
    Requiere crear una nueva auditoria previamente";
    print $empresas[$x][0];
    print $page_down;
    }
}
#cierre del else

```

```
exit;
```

```
CODIGO DEL ARCHIVO insertAudit.pl
```

```
#!/usr/bin/perl -U
```

```
#####  
#Proyecto de Tesis Auditoria Administrativa
```

```
#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM
```

```
#This program is free software; you can redistribute it and/or  
#modify it under the terms of the GNU General Public License  
#as published by the Free Software Foundation; either version 2  
#of the License, or (at your option) any later version.  
#####
```

```
use Pg;  
use CGI qw(:standard);  
use pemes::inicializar
```

```
$conn = Pg::connectdb("dbname=auditoria user=pardo");
```

```
$query = new CGI;
```

```
print $query->header;
```

```
print start_html(-title=>'Insercion de Auditorias Audit-admin');
```

```
$accion = $query->param(accion);
```

```
$id_normal = $query->param(dato_id);
```

```
@variables = &prueba();  
#variables que crean el formato de la pagina que se despliega  
$page_up = $variables[0];  
$page_down = $variables[1];  
#botones para el regreso  
$locacion1 = $variables[2];  
$locacion2 = $variables[3];  
$locacion3 = $variables[4];  
$botones = $variables[5];  
#terminan los formatos de la pagina
```

```
Pg::doQuery($conn,"select * from usuarios where id_usuario  
=$id_normal", \@datosUser);
```

```
#si no llego el id o llego pero no es un usuario del tipo administrador,  
se desecha
```

```
# el id es:$datosUser[0][0]  
# el nombre$datosUser[0][1]  
# AP$datosUser[0][2]<br>  
# AM$datosUser[0][3]<br>
```

```

# LOG${datosUser[0][4]}<br>
# PASS${datosUser[0][5]}<br>
# TIPO${datosUser[0][6]}<br>
# GRUPO${datosUser[0][7]}<br>

if ($id_normal eq "" || ${datosUser[0][6]} ne "1"){
    $contents="Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>";

    print $page_up.$contents.$page_down;
}else{
    if($accion eq "2"){

        $id_empresa = $query->param(id_emp);
        $n_auditor = $query->param(nombreAuditor);
        $aPaterno = $query->param(aPaterno);
        $aMaterno = $query->param(aMaterno);
        $horasDedicadas = $query->param(horasDedicadas);
        $fechaInicio = $query->param(fechaInicio);
        $comentarios = $query->param(comentarios);
        Pg::doQuery($conn,"select nombre_emp from empresa where
id_empresa=${id_empresa"},\@empresa);
        $contents = "Los datos recibidos e insertados son:<br>
<table>
<tr><td>Nombre de la empresa:</td><td> $empresa[0][0] </tr>
<tr><td>Fecha de Inicio:</td><td> $fechaInicio </tr>
<tr><td>Horas Dedicadas:</td><td> $horasDedicadas </tr>
<tr><td>Nombre del Auditor:</td><td> $n_auditor </tr>
<tr><td>Apellido Paterno del Auditor:</td><td> $aPaterno </tr>
<tr><td>Apellido Materno del Auditor:</td><td> $aMaterno </tr>
<tr><td>Comentarios:</td><td> $comentarios </tr>
</table>";

#poner el que saca el valor maximo para designar el nuevo id
Pg::doQuery($conn,"select (MAX(id_auditoria)) FROM
auditoria",\@maximo);
$maximo[0][0]=$maximo[0][0]+1;

#hacer la insercion de la auditoria
Pg::doQuery($conn,"insert into auditoria (id_empresa, id_auditoria,
id_usuario, n_auditor, ap_auditor, am_auditor, hr_dedicadas,
fecha_inicio, fecha_final, comentarios)
values('${id_empresa}', '$maximo[0][0]', '${id_normal}', '$n_auditor', '$aPaterno
', '$aMaterno', '$horasDedicadas', '$fechaInicio', null, '$comentarios')");
    print"insert into auditoria (id_empresa, id_auditoria, id_usuario,
n_auditor, ap_auditor, am_auditor, hr_dedicadas, fecha_inicio,
fecha_final, comentarios)
values('${id_empresa}', '$maximo[0][0]', '${id_normal}', '$n_auditor', '$aPaterno
', '$aMaterno', '$horasDedicadas', '$fechaInicio', null, '$comentarios')";

#    print "insert into empresa
values($maximo[0][0], '$clasificacion', '$giro', '$estateEmpresa', '$delegaci
onEmpresa', '$nameContact', '$nameEmpresa', '$phoneContact', '$firstContact',
'$secondContact', '$dateFundation', '$streetEmpresa', '$numberEmpresa', '$col
oniaEmpresa', '$cpEmpresa', '$municipioEmpresa')";

#para cambiar el valor del estatus
#    Pg::doQuery($conn,"select estatus from grupos where
id_grupo='${grupoUser}',\@estatus);

```

```

#   $estatus=$estatus[0][0]+1;

#   Pg::doQuery($conn,"update grupos set estatus='$estatus' where
id_grupo='$grupoUser'");
    $boton = "<form><input type=\"button\" value=\"Insertar Otra
Auditoria\"
onclick=\"parent.location.href='insertAudit.pl?dato_id=$id_normal'\"
></form>";

    print $page_up.$contents.$boton.$botones.$page_down;
}else{#cierre del if de accion

    Pg::doQuery($conn,"select id_empresa,nombre_emp from
empresa",\@empresas);

    $contents_up = "
MODULO DE INSERCIÓN DE DATOS NUEVAS AUDITORIAS AUDIT-ADMIN
<form name=\"forma\" action=\"insertAudit.pl\">

    Para insertar auditorias es necesario llenar los siguientes
datos:<br>
<table>
<tr><td><p>Elija empresa a auditar<br>
<select name=\"id_emp\">";

    print $page_up.$contents_up;

    $x = 0;

    while($empresas[$x][0] ne ""){

        print "<option value=$empresas[$x][0]>$empresas[$x][1]\n";

        $x ++;

    }

    $contents_middle = "</select></p></td>
<td><p>Fecha de inicio de auditoria \dd/mm/aaaa\"<br><input
type=\"text\" name=\"fechaInicio\" maxlength=10 size=10></p></td></tr>";

    print $contents_middle;

    $contents_down = "<tr><td><hr><p>Nombre del Auditor<br>
<input type=\"text\" name=\"nombreAuditor\" maxlength=\"50\"
size=\"30\"></p>
</td><td><hr><p>Apellido Paterno<br>
<input type=\"text\" name=\"aPaterno\" maxlength=\"50\"
size=\"30\"></p>
</td></tr><tr><td><p>Apellido Materno<br>
<input type=\"text\" name=\"aMaterno\" maxlength=\"50\"
size=\"30\"></p>
</td><td><p>Horas dedicadas<br>
<input type=\"text\" name=\"horasDedicadas\" maxlength=\"5\"
size=\"5\">
<input type=\"hidden\" name=\"accion\" value=\"2\">
<input type=\"hidden\" name=\"dato_id\" value=\"\$id_normal\">
</td></tr>
<tr><td align=center colspan=2>

```

```

                <textarea name=comentarios rows=15 cols=45
wrap=hard></textarea>
            </td></tr>
            <tr><td align=center colspan=2>
                <input type=\ "submit\ " value=\ "aceptar\ " value=\ "2\ ">
            </form>
            </td></tr></table>";

        print $contents_down.$botones.$page_down;
    }
#cierre del else de adentro
}
#cierre del else

exit;

CODIGO DEL ARCHIVO insertCat.pl

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$JSCRIPT=<<EOF;

function valida() {
    var miForma=document.formal;

    if (miForma.newName.value == ""){
        alert("El campo del nuevo dato en el catálogo no puede ser enviado
sin datos");
        miForma.newName.focus();
    }else{
        document.formal.submit();
    }
}

function validaP() {
//Esta funcion se utiliza solo para validar la insercion al catalogo de
preguntas.
    var miForma=document.formal;

    if (miForma.newName.value == ""){
        alert("El campo del nuevo dato en el catálogo no puede ser enviado
sin datos");
        miForma.newName.focus();
    }
}

```

```
//     }else if(miForma.subfuncion.value == ""){
//     alert("Es necesario seleccionar una subfuncion a la cual relacionar
la pregunta");
//     }else if(miForma.factor.value == ""){
//     alert("Es necesario seleccionar un factor al cual relacionar la
pregunta");
    }else{
        document.formal.submit();
    }
}

function daleValue(options,nombre) {
    var miForma=document.formal;
    var elElemento=nombre;
    var valorOption=options[options.selectedIndex].value;

    if (elElemento == "sub"){
        miForma.subfuncion.value=valorOption;
    }
    if(elElemento == "fact"){
        miForma.factor.value=valorOption;
    }
}

EOF
;

#
$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;
$action = $query->param(accion);
$id_normal = $query->param(dato_id);
$catalogo = $query->param(catalogo);

print start_html(-title=>'Insercion de Catalogos Audit-Admin',
                -script=>$JSCRIPT);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];#no se usan
#terminan los formatos de la pagina
```

```

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "3" ){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
<br>";

    print $page_up.$contents.$botones.$page_down;
}else{
    if($accion ne ""){
#si ya se inserto un catalogo recibe los parametros para realizar la
insercion en la base de datos
#esto recibe
#nombre =newName
#catalogo
#dato_id
#accion
#nombre del nuevo elemento en el catalogo
$newName = $query->param(newName);
$atributo = "descripcion";
if($catalogo eq "subfunciones"){
    $atributo = "nombre";
    $nameCat = "cat_subfunciones";
    $idCat = "id_sub";
}elsif($catalogo eq "factores"){
    $nameCat = "cat_factores";
    $idCat = "id_factor";
    $atributo = "nombre";
}elsif($catalogo eq "clasificacion"){
    $nameCat = "clasificacion";
    $idCat = "id_clas";
}elsif($catalogo eq "grupos"){
    $nameCat = "grupos";
    $idCat = "id_grupo";
}elsif($catalogo eq "preguntas"){
    $nameCat = "cat_preguntas";
    $idCat = "id_pregunta";
}
}
$contents = "Los datos recibidos e insertados son:<br>
Nombre: $nameUser<br>
Apellido Paterno: $firstUser<br>
Apellido Materno: $secondUser<br>
Login: $loginUser <br>
Pass: $passwordUser <br>

```

```

    Tipo de usuario :$tipoUser<br>
    Grupo: $grupoUser <br>
    ";
    print $page_up;
#IMPORTANTE EL CASO DE PREGUNTAS Y GRADOS SON CASOS ESPECIALES
#poner el que saca el valor maximo para designar el nuevo id

#   print "select (MAX($idCat)) FROM $nameCat<br>";

#hacer la insercion del nuervo elemento en el catalogo
if($catalogo eq "preguntas"){#antes preguntas
    Pg::doQuery($conn,"select (max($idCat)) FROM $nameCat",\@maximo);
    $maximo[0][0] ++;

    $subfuncion = $query->param(subfuncion);
    $factor = $query->param(factor);
    Pg::doQuery($conn,"insert into $nameCat values
($maximo[0][0], '$newName')");

    }else{
        Pg::doQuery($conn,"select max(to_number($idCat, '9999999999'))
FROM $nameCat",\@maximo);
        $maximo[0][0] ++;
#        print"insert into $nameCat values($maximo[0][0], '$newName')";
        Pg::doQuery($conn,"insert into $nameCat
values($maximo[0][0], '$newName')");
    }
    #print "insert into $nameCat values($maximo[0][0], '$newName')<br>";

#comprobacion de insercion
Pg::doQuery($conn,"select $atributo from $nameCat order by
$atributo",\@descripcion);
#print "select descripcion from $nameCat";
$num = $#descripcion + 1;
if ($num > 15){
    $num=15;
}

print"<div align=center>
<form>
El cat&aacute;logo:$catalogo se ha actualizado:<br>
<SELECT NAME=\"subfunciones\" size=\" $num\"><br>";

$x = 0;
while ( $descripcion[$x][0] ne "" ) {
    #imprime las subfunciones existentes
    print "<OPTION >$descripcion[$x][0]\n";

    $x++;
}
#referencia de lo que le mando a las ligas
#   LOG$datosUser[0][4]<br>
#   PASS$datosUser[0][5]<br>
print "</select></form></div>";
$locacion1 = "insertCat.pl?dato_id=$id_normal&catalogo=$catalogo";
$locacion2 = "opCatalogos.pl?dato_id=$id_normal";
$locacion3 = "valida.pl?dato_id=$id_normal";
$botones = "<div align=center>
<form method=post>

```

```

        <input type=\"button\" value=\"Insertar otro elemento\"
onclick=\"parent.location.href='$locacion1'\">\n
        <input type=\"button\" value=\"Operaciones con catalogos\"
onclick=\"parent.location.href='$locacion2'\">\n
        <input type=\"button\" value=\"Principal\"
onclick=\"parent.location.href='$locacion3'\">\n
    </form></div>";
    print$botones.$page_down;

# print $page_up.$contents.$page_down;
}else{

#si no viene parametro es que apenas hay que elegir en que catalogo se va
a insertar y presentar la forma que permita hacerlo

    if($catalogo eq "subfunciones"){

#si se va a insertar en subfunciones

        Pg::doQuery($conn,"select nombre FROM cat_subfunciones order by
nombre",\@subfunciones);

        $num_subfunciones = $#subfunciones +1 ;

        if ($num_subfunciones > 10){

            $num = 10;

        }else{

            $num = $num_subfunciones;

        }
        $contents_up="
MODULO DE ALTA -CATALOGO SUBFUNCIONES- AUDIT-ADMIN
<form name=\"formal\" method=\"post\" action=insertCat.pl>
Las Subfunciones existentes son:<br>
<SELECT NAME=\"subfunciones\" size=\"$num\"><br>";

print $page_up.$contents_up;

$x = 0;
while ( $subfunciones[$x][0] ne "" ) {
    #imprime las subfunciones existentes
    print "<OPTION >$subfunciones[$x][0]\n";

    $x++;
}

$contents_down = "
</SELECT><br><br>
<p>Nombre de la nueva subfuncion<br>\n
size=\"100\">\n
    <input type=\"text\" name=\"newName\" maxlength=\"100\"
value=\"$catalogo\">\n
    <input type=\"hidden\" name=\"dato_id\"
value=\"$id_normal\">\n
    <input type=\"hidden\" name=\"accion\" value=\"2\">\n
    <input type=\"button\" value=\"Acepta\"
Onclick=\"valida()\"></p>

```

```

        </form>";

        print $contents_down.$botones.$page_down;
#si se va a insertar en factores
        }elseif($catalogo eq "factores"){

                Pg::doQuery($conn,"select nombre FROM cat_factores order by
nombre",\@factores);
                $num_factores = $#factores +1 ;
                if ($num_factores > 15){
                        $num = 15;
                }else{
                        $num = $num_factores;
                }
                $contents_up = "
                MODULO DE ALTA -CATALOGO FACTORES- AUDIT-ADMIN
                <form name=\"formal\" method=\"post\" action=insertCat.pl>
                Los Factores existentes son:<br>
                <SELECT NAME=\"factores\" size=\"\$num\"><br>";

                print $page_up.$contents_up;

                $x = 0;

                while ( $factores[$x][0] ne "" ) {
                        #imprime los factores existentes
                        print "<OPTION >$factores[$x][0]\n";

                        $x++;
                }

                $contents_down = "
                </SELECT><br><br>
                <p>Nombre del nuevo Factor<br>\n
                <input type=\"text\" name=\"newName\" maxlength=\"50\"
size=\"50\">\n
                <input type=\"hidden\" name=\"catalogo\"
value=\"\$catalogo\">\n
                <input type=\"hidden\" name=\"dato_id\"
value=\"\$id_normal\">\n
                <input type=\"hidden\" name=\"accion\" value=\"2\">\n
                <input type=\"button\" value=\"Aceptar\"
Onclick=\"valida()\"></p>
                </form>";

                print $contents_down.$page_down;
#cierra elsif de factores
        }elseif($catalogo eq "clasificacion"){
#si se va a insertar en clasificacion

                Pg::doQuery($conn,"select descripcion FROM
clasificacion",\@clasificacion);
                $num_clasificacion = $#clasificacion +1 ;
                if ($num_clasificacion > 10){
                        $num = 10;
                }else{
                        $num = $num_clasificacion;
                }
                $contents_up = "
                MODULO DE ALTA -CATALOGO CLASIFICACION- AUDIT-ADMIN
                <form name=\"formal\" method=\"post\" action=insertCat.pl>
                Las Clasificaciones existentes son:<br>

```

```

        <SELECT NAME=\ "clasificaciones\ " size=\ "$num\ "><br>";
print $page_up.$contents_up;

$x = 0;

while ( $clasificacion[$x][0] ne "" ) {
    #imprime los factores existentes
    print "<OPTION >$clasificacion[$x][0]\n";

    $x++;
}

$contents_down = "
</SELECT><br><br>
<p>Nombre de la nueva Clasificacion<br>\n
<input type=\ "text\ " name=\ "newName\ " maxlength=\ "50\ "
size=\ "50\ ">\n
<input type=\ "hidden\ " name=\ "catalogo\ "
value=\ "$catalogo\ ">\n
<input type=\ "hidden\ " name=\ "dato_id\ "
value=\ "$id_normal\ ">\n
<input type=\ "hidden\ " name=\ "accion\ " value=\ "2\ ">\n
<input type=\ "button\ " value=\ "Aceptar\ "
OnClick=\ "valida()\ "></p>
</form>";

    print $contents_down.$page_down;
#cierra elsif de clasificacion
#si se va a insertar en clasificacion
}elsif($catalogo eq "grupos"){

    Pg::doQuery($conn,"select descripcion FROM grupos order by
nombre", \@grupos);
    $num_grupos = $#grupos +1 ;
    if ($num_grupos > 10){
        $num = 10;
    }else{
        $num = $num_grupos;
    }
    $contents_up = "
MODULO DE ALTA -CATALOGO GRUPOS- AUDIT-ADMIN
<form name=\ "formal\ " method=\ "post\ " action=insertCat.pl>
Los Grupos existentes son:<br>
<SELECT NAME=\ "grupos\ " size=\ "$num\ "><br>";

print $page_up.$contents_up;

$x = 0;

while ( $grupos[$x][0] ne "" ) {
#imprime los factores existentes
    print "<OPTION >$grupos[$x][0]\n";

    $x++;
}

$contents_down="
</SELECT><br><br>
<p>Nombre del nuevo Grupo<br>\n
<input type=\ "text\ " name=\ "newName\ " maxlength=\ "50\ "
size=\ "50\ ">\n

```

```

        <input type=\"hidden\" name=\"catalogo\"
value=\"${catalogo}\">\n
        <input type=\"hidden\" name=\"dato_id\"
value=\"${id_normal}\">\n
        <input type=\"hidden\" name=\"accion\" value=\"2\">\n
        <input type=\"button\" value=\"Aceptar\"
onclick=\"valida()\"></p>
        </form>";

        print $contents_down.$page_down;
#cierra elsif de grupos
    }
#si se va a insertar en clasificacion
    elsif(${catalogo} eq "preguntas"){

        Pg::doQuery($conn,"select descripcion FROM cat_preguntas order by
descripcion",\@preguntas);
#        Pg::doQuery($conn,"select nombre,id_sub FROM cat_subfunciones
order by nombre",\@catSubfunciones);
#        Pg::doQuery($conn,"select nombre,id_factor FROM cat_factores
order by nombre",\@catFactores);

        $num_preguntas = $#preguntas +1 ;

        if ($num_preguntas > 10){

            $num = 10;

        }else{

            $num = $num_preguntas;

        }

        $contents_up = "
MODULO DE ALTA -CATALOGO PREGUNTAS- AUDIT-ADMIN
<form name=\"formal\" method=\"post\" action=insertCat.pl>
Las Preguntas existentes son:<br>
<SELECT NAME=\"preguntas\" size=\"${num}\"><br>";

        print $page_up.$contents_up;

        $x = 0;

        while ( $preguntas[$x][0] ne "" ) {
#imprime las preguntas existentes
            print "<OPTION >${preguntas[$x][0]}\n";

            $x ++;
        }
        print "                </SELECT><br><br>";

        $contents_down = "
<p>Nombre de la nueva Pregunta<br>\n
        <input type=\"text\" name=\"newName\" size=\"70\">\n
        <input type=\"hidden\" name=\"catalogo\"
value=\"${catalogo}\">\n
        <input type=\"hidden\" name=\"dato_id\"
value=\"${id_normal}\">\n
        <input type=\"hidden\" name=\"subfuncion\" value=\"\">\n
        <input type=\"hidden\" name=\"factor\" value=\"\">\n

```

```

        <input type="hidden" name="accion" value="2">\n
        ";

        print $contents_down;

#         print "<table><tr><td>Subfunciones<br>\n
#             <SELECT NAME="subfunciones" size="3"
#             onChange=daleValue(this.options,'sub')>";

#         $x = 0;
#         while ( $catSubfunciones[$x][0] ne "" ) {
#imprime las subfunciones existentes en el catalogo
#             $catSubfunciones[$x][1] =~ s/ //g;
#             print "<OPTION
#             value=" $catSubfunciones[$x][1] \">$catSubfunciones[$x][0]\n";
#
#             $x ++;
#         }

#         print"</SELECT></td>
#             <td>Factores<br>
#             <SELECT NAME="factores" size="3"
#             onChange=daleValue(this.options,'fact')>";

#         $x = 0;

#         while ( $catFactores[$x][0] ne "" ) {

#imprime los factores existentes en el catalogo
#             $catFactores[$x][1] =~ s/ //g;
#             print "<OPTION
#             value=" $catFactores[$x][1] \">$catFactores[$x][0]\n";

#             $x ++;
#         }

#         print"
#             </SELECT>
#             </td></tr></table>\n
#             <p><div align=center><input type="button"
#             value="Aceptar" onclick="validaP()"></div></p>
#             </form>";

        print $botones.$page_down;
    }
#cierra elsif de preguntas
}
#cierre del else

exit;

```

CODIGO DEL ARCHIVO insertUser.pl

```
#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####
use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;
print $query->header;

print start_html(-title=>'Insercion de Usuarios Audit-admin');

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];#no se usa
$cerrar_ventana = $variables[6];
#terminan los formatos de la pagina

$action = $query->param(accion);
$id_normal = $query->param(dato_id);

Pg::doQuery($conn,"select *from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "3"){
    $contents="Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
```

```

administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    llego $id_normal<br>
    enc $id_encryptado<br>
tipo $datosUser[0][6]
";

    print $page_up.$contents.$cerrar_ventana.$page_down;
}else{

    $nameUser = $query->param(nombre);
    $firstUser = $query->param(apaterno);
    $secondUser = $query->param(amaterno);
    $loginUser = $query->param(login);
    $passwordUser = $query->param(password);
    $tipoUser = $query->param(TipoUsuario);
    $grupoUser = $query->param(grupo);
Pg::doQuery($conn,"select login from usuarios",\@logins);
    $x=0;
    $loginUser =~ s/ //g;

    $limite= @logins;

    while( $x < $limite ){

        $logins[$x][0]=~ s/ //g;

        if ($logins[$x][0] eq $loginUser){

            $mensaje = "<font color=red>El login insertado \"ya
existe\"</font>,<br>
                la operacion fue cancelada.<br>
                Intente insertar nuevamente el usuario con un nuevo login
\n";

            print $page_up.$mensaje.$page_down;

            $pasa="no";

            $x=$limite;
        }
        $x ++;
    }

    if($pasa ne "no"){
        $contents="Los datos recibidos e insertados son:<br>
        Nombre: $nameUser<br>
        Apellido Paterno: $firstUser<br>
        Apellido Materno: $secondUser<br>
        Login: $loginUser <br>
        Pass: $passwordUser <br>
        Tipo de usuario :$tipoUser<br>
        Grupo: $grupoUser <br>";
        #poner el que saca el valor maximo para designar el nuevo id
Pg::doQuery($conn,"select (MAX(id_usuario)) FROM
usuarios",\@maximo);
        $maximo[0][0] = $maximo[0][0]+1;

        #hacer la insercion del nuevo usuario

```

```

Pg::doQuery($conn,"insert into usuarios
(id_usuario,nombre,apaterno,amaterno,login,password,grupo)
values($maximo[0][0],$nameUser,$firstUser,$secondUser,$loginUser,
$passwordUser,$tipoUser,$grupoUser)");

```

```

    print $page_up.$contents.$cerrar_ventana.$page_down;
}
}#cierre del else
exit;

```

CODIGO DEL ARCHIVO modifUser.pl

```

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print $query->start_html;

$accion = $query->param(accion);

$id_normal = $query->param(dato_id);

$usuario = $query->param(id);

#variables que crean el formato de la pagina que se despliega
$page_up = " <html>\n
<body bgcolor=white background=\"../Tesis/Imagenes/bglogito.jpg\">\n
<table align=center><tr><td>\n";
$page_down = "
</td></tr></table>\n
</body>\n
</html>\n";

#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal", \@datosUser);

```

```

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre$datosUser[0][1]
#   AP$datosUser[0][2]<br>
#   AM$datosUser[0][3]<br>
#   LOG$datosUser[0][4]<br>
#   PASS$datosUser[0][5]<br>
#   TIPO$datosUser[0][6]<br>
#   GRUPO$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "3"){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    llego $id_normal<br>
    enc $id_encryptado<br>
tipo $datosUser[0][6]";

    print $page_up.$contents.$page_down;
}else{
    if ($accion eq ""){
        Pg::doQuery($conn,"select * from usuarios where id_usuario
=$usuario",\@datosUser);
        Pg::doQuery($conn,"select * from grupos where id_grupo
=$datosUser[0][7]",\@grupoUser);
        Pg::doQuery($conn,"select * from grupos where id_grupo
<>$datosUser[0][7]",\@nogruposUser);

        Pg::doQuery($conn,"select * from tipousuario where id_tipo
=$datosUser[0][6]",\@tipoUser);
        Pg::doQuery($conn,"select * from tipousuario where id_tipo <>
$datosUser[0][6]",\@notipoUser);
#   el id es:$datosUser[0][0]
    $datosUser[0][0]=~ s/ //g;
#   el nombre$datosUser[0][1]
    $datosUser[0][1]=~ s/ //g;
#   AP$datosUser[0][2]<br>
    $datosUser[0][2]=~ s/ //g;
#   AM$datosUser[0][3]<br>
    $datosUser[0][3]=~ s/ //g;
#   LOG$datosUser[0][4]<br>
    $datosUser[0][4]=~ s/ //g;
#   PASS$datosUser[0][5]<br>
    $datosUser[0][5]=~ s/ //g;
#   TIPO$datosUser[0][6]<br>
    $datosUser[0][6]=~ s/ //g;
#   GRUPO$datosUser[0][7]<br>
    $datosUser[0][7]=~ s/ //g;
    $contents= "
        <form name=InsertaUser action=modifUser.pl>
        Actualice los campos con los nuevos valores:<br>
        <table>
        <tr><td>
        <p>Nombre del usuario<br>
        <input type="text" name="nombre" maxlength="30" size="30"
value="\ $datosUser[0][1]\ "></p>
        </td><td>
        <p>Apellido Paterno<br>

```

```

        <input type=\"text\" name=\"apaterno\" maxlength=\"30\"
size=\"30\" value=\"\${datosUser[0][2]}\"></p>
        </td></tr>
        <tr><td><p>Apellido Materno<br>
        <input type=\"text\" name=\"amaterno\" maxlength=\"30\"
size=\"30\" value=\"\${datosUser[0][3]}\"></p>
        </td><td><p>Login<br>
        <input type=\"text\" name=\"login\" maxlength=\"20\" size=\"20\"
value=\"\${datosUser[0][4]}\"></p>
        </td></tr>
        <tr><td><p>Password<br>
        <input type=\"password\" name=\"password\" maxlength=\"30\"
size=\"30\" value=\"\${datosUser[0][5]}\">
        </td></tr>
        <tr><td>;

        print $page_up.$contents;

        print \"<SELECT NAME=\"grupo\"><br>\";

        if (\$grupoUser[0][0] ne \"\") {

                print \"<OPTION
value=\"\${grupoUser[0][0]}\">\${grupoUser[0][1]}\"\\n\";
        }
        $x = 0;

        while ( $nogrupouser[$x][0] ne \"\" ) {

                print \"<OPTION
value=\"\${nogrupouser[$x][0]}\">\${nogrupouser[$x][1]}\"\\n\";

                $x++;
        }

        print \"</SELECT><br><br></td><td>\";

        print \"<SELECT NAME=\"TipoUsuario\"><br>\";

        if (\$tipoUser[0][0] ne \"\") {

                print \"<OPTION
value=\"\${tipoUser[0][0]}\">\${tipoUser[0][1]}\"\\n\";
        }

        $x = 0;

        while ( $notipoUser[$x][0] ne \"\" ) {

                print \"<OPTION
value=\"\${notipoUser[$x][0]}\">\${notipoUser[$x][1]}\"\\n\";

                $x++;
        }

        print \"</SELECT></td></tr><br><br>\";

        $contents_down=\"<tr><td>
        <input type=\"hidden\" name=\"dato_id\" value=$id_normal>
        <input type=\"hidden\" name=\"accion\" value=1>
        <input type=\"hidden\" name=\"id\" value=$usuario>
        <input type=\"submit\" value=\"Actualizar\">

```

```

        </form></td></tr></table>";

    print $contents_down.$page_down;

}else{

    $nameUser = $query->param(nombre);
#quita los espacios
    $nameUser =~ s/ //g;
    $firstUser = $query->param(apaterno);
    $firstUser =~ s/ //g;
    $secondUser = $query->param(amaterno);
    $secondUser =~ s/ //g;
    $loginUser = $query->param(login);
    $loginUser =~ s/ //g;
    $passwordUser = $query->param(password);
    $passwordUser =~ s/ //g;
    $grupoUser = $query->param(grupo);
    $grupoUser =~ s/ //g;
    $tipoUser = $query->param(TipoUsuario);
    $tipoUser =~ s/ //g;
    Pg::doQuery($conn,"select login from usuarios where id_usuario not
in(select id_usuario from usuarios where id_usuario=$usuario"),\@logins);
    $x = 0;
    $loginUser =~ s/ //g;

    $limite = @logins;

    while( $x < $limite ){

        $logins[$x][0] =~ s/ //g;

        if ($logins[$x][0] eq $loginUser){

            $mensaje = "<font color=red>El login insertado \"ya
existe\"</font>,<br>
la operacion fue cancelada.<br>
Intente modificar nuevamente el usuario con un nuevo login \n";

            print $page_up.$mensaje.$page_down;
            $pasa = "no";
            $x = $limite;
        }
        $x ++;
    }

    if ($pasa ne "no"){

        $contents = "Los datos recibidos y actualizados son:<br>
Nombre: $nameUser<br>
Apellido Paterno: $firstUser<br>
Apellido Materno: $secondUser<br>
Login: $loginUser <br>
Pass: $passwordUser <br>
Grupo: $grupoUser <br>
Tipo de usuario :$tipoUser<br>";

        #poner el que saca el valor maximo para designar el nuevo id
        Pg::doQuery($conn," update usuarios set nombre='$nameUser',
apaterno='$firstUser', amaterno='$secondUser', login='$loginUser',

```

```

password='$passwordUser', grupo='$grupoUser', tipo='$tipoUser' where
id_usuario='$usuario' ");

    print " update usuarios set nombre='$nameUser',
apaterno='$firstUser', amaterno='$secondUser', login='$loginUser',
password='$passwordUser', grupo='$grupoUser', tipo='$tipoUser' where
id_usuario='$usuario'";

    print $page_up.$contents.$page_down;
}
}

}#cierre del else

exit;

```

CODIGO DEL ARCHIVO opAuditorias.pl

```

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

#####
#la siguiente funcion javascript sirve para cambiar la pagina de
#manera automatica una vez que el usuario ha elegido una operacion
#a realizar, enviando los datos necesarios.
#####
$JSCRIPT=<<EOF;

var nombreCgi = "";
var operacion = "";

function spawnWin(options) {
    var url = options[options.selectedIndex].value;
    parent.location.href=url;
}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Operaciones con auditorias',

```

```

                                -script=>${JSCRIPT});
#
#   alert(" checale url=" + nombreCgi + url);

$accion = $query->param(accion);

$sid_normal = $query->param(dato_id);

@variables = &prueba2("miForma");
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$sid_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre:$datosUser[0][1]
#   AP:$datosUser[0][2]<br>
#   AM:$datosUser[0][3]<br>
#   LOG:$datosUser[0][4]<br>
#   PASS:$datosUser[0][5]<br>
#   TIPO:$datosUser[0][6]<br>
#   GRUPO:$datosUser[0][7]<br>

if ($sid_normal eq "" || $datosUser[0][6] ne "1"){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    llego $sid_normal<br>
tipo $datosUser[0][6]";

    print $page_up.$contents.$page_down;

}else{

#esto despliega el menu de las operaciones con el usuario si aun no se ha
definido una accion a realizar
#el select le pasa a la variable del javascript el nombre del catalogo a
modificar y el id del usuario
    $contents="
    <table align=center><tr>
    <td>
    <form method=post name=\"miForma\">
    <select name=\"act_Usuario\" onChange=\"spawnWin(this.options)\\"
size=4>\n
    <option value=\"insertAudit.pl?dato_id=$sid_normal\">Nueva
Auditoria</option>\n
    <option value=\"workAudit.pl?dato_id=$sid_normal\">Trabajar con
auditorias existentes</option>\n

```

```

        <option
value="\modifAudit.pl?dato_id=$id_normal\ ">Modificaci&oacute;n de datos
de la auditoria</option\n>
        <option value="\closeAudit?dato_id=$id_normal\ ">Cerrar
auditoria</option\n>

        </select>
        </form>
</td></tr></table>";

        print $page_up.$contents.$botones.$page_down;
}
#cierre del else principal

exit;

```

CODIGO DEL ARCHIVO opCatalogos.pl

```

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####
use Pg;
use CGI qw(:standard);
use pemes::inicializar;

#####
#la siguiente funcion javascript sirve para cambiar la pagina de
#manera automatica una vez que el usuario ha elegido una operacion
#a realizar, enviando los datos necesarios.
#####

$JSCRIPT=<<<EOF;

var nombreCgi = "";
var operacion = "";

function spawnWin(options) {
    var url = options[options.selectedIndex].value;
    if( nombreCgi != ""){
        parent.location.href = nombreCgi+url;

    }else{
        alert("Requiere elegir una operacion de ALTA, BAJA o CAMBIO
previamente");
    }
}

}

```

```

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;

print $query->header;

print start_html(-title=>'Modificaciones a Catalogos Audit-Admin',
                 -script=>$JSCRIPT);

$accion = $query->param(accion);

$id_normal = $query->param(dato_id);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre:$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "3"){
    $contents="Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    llego $id_normal<br>
tipo $datosUser[0][6]";

    print $page_up.$contents.$page_down;
}else{

    #esto despliega el menu de las operaciones con el usuario si aun no se
ha definido una accion a realizar
    #el select le pasa a la variable del javascript el nombre del catalogo
a modificar y el id del usuario
    $contents="

```

```

<table align=center><tr>
<td>
<form method=post>
<p>Elija la accion a realizar con el cat&acute;logo:</p>
<input type=\"hidden\" name=\"dato_id\" value=\"\${id_normal}\">
<input type=\"radio\" name=\"accionCatalogo\"
onClick=\"nombreCgi='insertCat.pl?catalogo='\">Alta<br>
<input type=\"radio\" name=\"accionCatalogo\"
onClick=\"nombreCgi='delCat.pl?catalogo='\">Baja<br>
<input type=\"radio\" name=\"accionCatalogo\"
onClick=\"nombreCgi='changeCat.pl?catalogo='\">Cambio
</td>
<td>

<select name=\"act_Usuario\" onChange=\"spawnWin(this.options)\"
size=5>\n
<option value=\"subfunciones&dato_id=\${id_normal}\">Cat&acute;logo de
subfunciones</option>\n
<option value=\"factores&dato_id=\${id_normal}\">Cat&acute;logo de
factores</option>\n
<option value=\"clasificacion&dato_id=\${id_normal}\">Cat&acute;logo
de clasificaci&oacute;n</option>\n
<option value=\"grupos&dato_id=\${id_normal}\">Cat&acute;logo de
grupos</option>\n
<option value=\"preguntas&dato_id=\${id_normal}\">Cat&acute;logo de
preguntas</option>\n

</select>
</form>
</td></tr></table>";

print $page_up.$contents.$botones.$page_down;
}
#cierre del else principal

exit;

CODIGO DEL ARCHIVO opEmpChange.pl

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

#####
#la siguiente funcion javascript sirve para cambiar la pagina de
#manera automatica una vez que el usuario ha elegido una operacion
#a realizar, enviando los datos necesarios.

```

```
#####

$JSCRIPT=<<EOF;

function spawnWin(options) {
  var url = options[
    options.selectedIndex
  ].value;
  parent.location.href= url;
}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;
print $query->header;
print start_html(-title=>'Personal Profile',
  -script=>$JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);

@variables= &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up=$variables[0];
$page_down=$variables[1];
#botones para el regreso
$locacion1=$variables[2];
$locacion2=$variables[3];
$locacion3=$variables[4];
$botones=$variables[5];
#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre:$datosUser[0][1]
#   AP:$datosUser[0][2]<br>
#   AM:$datosUser[0][3]<br>
#   LOG:$datosUser[0][4]<br>
#   PASS:$datosUser[0][5]<br>
#   TIPO:$datosUser[0][6]<br>
#   GRUPO:$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "1"){
  $contents="Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
";

```

```

    print $page_up.$contents.$botones.$page_down;
}else{
    if($accion ne "2" && $accion ne "3"){
        #modificacion de los datos dela empresa

        Pg::doQuery($conn,"select id_empresa,nombre_emp from empresa where
id_empresa not in (select id_empresa from auditoria)",\@empresas);
        print $page_up;

        if($dimension=@empresas > 0){
            if($dimension < 10){
                $longitud=$dimension;
            }else {
                $longitud=10;
            }
            print "<form>El siguiente listado solo presenta empresas a las que
no se han asignado auditorias. Si la empresa que busca no se encuentra en
el listado puede significar que tiene auditorias asignadas. Seleccione
empresa a modificar:
<div align=center>
<SELECT NAME=\"usuarios\" size=\"\$longitud\"
onChange=\"spawnWin(this.options)\><br>";
            print "<OPTION>Elija:\n";
            $x = 0;
            while ( $empresas[$x][0] ne " " ) {

                print "<OPTION
value=opEmpChange.pl?dato_id=$id_normal\&id=$empresas[$x][0]&accion=2>\\"$
empresas[$x][1]\\"<br>";

                $x++;
            }
            print "</SELECT>
</div>

</form><br><br>";

            print $botones.$page_down;

            }else{#si no hay empresas para modificar

                print "Por el momento no existen empresas con permiso para ser
editadas, verifique que la empresa por seleccionar no este siendo
utilizada por una auditoria";
                print $botones.$page_down;
            }#cierre else de empresas a modificar

        }elseif($accion eq "2"){
            print $page_up;

            $id = $query->param(id);
            # print "has llegado camarada el id de la empresa es $id";
            Pg::doQuery($conn,"select
id_clas,id_giro,id_est,id_del,nombre,nombre_emp,telefono,a_paterno,a_mate
rno,fecha_fundacion,calle,numero,colonia,cp,municipio from empresa where
id_empresa='$id'",\@datos_empresa);

            $nameEmpresa = $datos_empresa[0][5];
            $dateFundation = $datos_empresa[0][9];
            $clasificacion1 =$datos_empresa[0][0];

```

```

Pg::doQuery($conn,"select id_clas,descripcion from clasificacion
where id_clas='$clasificacion1',\@clasifEmp);

$giro1 =$datos_empresa[0][1];
Pg::doQuery($conn,"select id_giro,descripcion from giro where
id_giro='$giro1',\@giroEmp);
$nameContact =$datos_empresa[0][4];
$firstContact =$datos_empresa[0][7];
$secondContact =$datos_empresa[0][8];
$phoneContact =$datos_empresa[0][6];
$streetEmpresa =$datos_empresa[0][10];
$numberEmpresa =$datos_empresa[0][11];
$coloniaEmpresa =$datos_empresa[0][12];
$cpEmpresa =$datos_empresa[0][13];
$municipioEmpresa =$datos_empresa[0][14];
$stateEmpresal =$datos_empresa[0][2];
Pg::doQuery($conn,"select id_est,nombre from estado where
id_est='$stateEmpresal',\@estateEmp);
$delegacionEmpresal =$datos_empresa[0][3];
Pg::doQuery($conn,"select id_del,nombre from delegacion where
id_del='$delegacionEmpresal',\@delEmp);

#/////////
Pg::doQuery($conn,"select id_clas,descripcion from
clasificacion",\@clasificacion);
Pg::doQuery($conn,"select id_giro,descripcion from giro",\@giro);
Pg::doQuery($conn,"select id_est,nombre from estado",\@estado);
Pg::doQuery($conn,"select id_del,nombre from
delegacion",\@delegacion);

$content_up="
MODULO DE MODIFICACION DE DATOS EMPRESAS AUDIT-ADMIN
<form name=\"forma\" action=\"opEmpChange.pl\">
Para dar insertar empresas es necesario llenar los siguientes
datos:<br>
<table>
<tr><td><p>Nombre de la Empresa<br>
<textarea name=\"nombreEmp\" wrap=\"hard\" rows=\"3\"
cols=\"50\">$nameEmpresa</textarea></p></td>
<td><p>Fecha de fundaci&oacute;n \dd/mm/aaaa\"<br><input
type=\"text\" value=\"$dateFundacion\" name=\"fechaFundacion\"
maxlength=10 size=10></p></td></tr>
<tr><td>Clasificaci&oacute;n<br>
<select name=\"clasificacion\">
";

print $content_up;
print "<option value=$clasifEmp[0][0]>$clasifEmp[0][1] \n";

$x=0;
while($clasificacion[$x][0] ne ""){

print "<option
value=$clasificacion[$x][0]>$clasificacion[$x][1]\n";

$x= $x + 1;
}
print "</select></td><td>Giro<br><select name=\"giro\">";
$x=0;
print "<option value=$giroEmp[0][0]>$giroEmp[0][1]\n";
while($giro[$x][0] ne ""){

```

```

print "<option value=$giro[$x][0]>$giro[$x][1]\n";
    $x= $x + 1;
}
print "</select></td></tr>";

$contents_down="<tr><td><hr><p>Nombre del Contacto<br>
<input type=\"text\" name=\"nombreContacto\" value=\"\$nameContact\"
maxlength=\"50\" size=\"50\"></p>
</td><td><hr><p>Apellido Paterno<br>
<input type=\"text\" name=\"aPaterno\" value=\"\$firstContact\"
maxlength=\"50\" size=\"50\"></p>
</td></tr><tr><td><p>Apellido Materno<br>
<input type=\"text\" name=\"aMaterno\" value=\"\$secondContact\"
maxlength=\"50\" size=\"50\"></p>
</td><td><p>Tel&eacute;fono<br>
<input type=\"text\" name=\"telefono\" value=\"\$phoneContact\"
maxlength=\"21\" size=\"21\">
<input type=\"hidden\" name=\"dato_id\" value=$id_normal>
<input type=\"hidden\" name=\"accion\" value=3></td></tr>
<input type=\"hidden\" name=\"id\" value=$id></td></tr>
<tr><td>
<hr>Domicilio de la Empresa<br>Calle<br>
<input type=\"text\" name=\"calle\" value=\"\$streetEmpresa\"
maxlength=\"50\" size=\"50\"></td>
<td><hr>N&uacute;mero<br>
<input type=\"text\" name=\"numero\" value=\"\$numberEmpresa\"
maxlength=\"4\" size=\"4\"></td>
</tr><tr><td>Colonia<br>
<input type=\"text\" name=\"colonia\" value=\"\$coloniaEmpresa\"
maxlength=\"50\" size=\"50\"></td>
<td>C&oacute;digo Postal<br>
<input type=\"text\" name=\"cp\" value=\"\$cpEmpresa\"
maxlength=\"5\" size=\"5\"></td></tr>
<tr><td>Municipio<br>
<input type=\"text\" name=\"municipio\" value=\"\$municipioEmpresa\"
maxlength=\"50\" size=\"50\"></td></tr>";

print $contents_down;
print " <tr><td>Estado<br>
<select name=\"estado\">
";
print "<option value=$estateEmp[0][0]>$estateEmp[0][1]\n";
    $x=0;
    while($estado[$x][0] ne ""){
print "<option value=$estado[$x][0]>$estado[$x][1]\n";
        $x= $x + 1;
    }
print "</select></td><td>Delegaci&oacute;n<br><select
name=\"delegacion\">
<option value=$delEmp[0][0]>$delEmp[0][1]
";
    $x=0;
    while($delegacion[$x][0] ne ""){
print "<option value=$delegacion[$x][0]>$delegacion[$x][1]\n";
        $x= $x + 1;
    }

```

```

    }
    print "</select></td></tr>
    <tr><td colspan=2 align=\"center\"><input type=\"submit\"
value=\"Guardar Cambios\"> </td></tr>
</form></table>";

    #////////

    print $botones.$page_down;

    }elseif($accion eq "3"){
    $sid = $query->param(id);
    $nameEmpresa = $query->param(nombreEmp);
    $dateFundation = $query->param(fechaFundacion);
    $clasificacion = $query->param(clasificacion);
    $giro = $query->param(giro);
    $nameContact = $query->param(nombreContacto);
    $firstContact = $query->param(aPaterno);
    $secondContact = $query->param(aMaterno);
    $phoneContact = $query->param(telefono);
    $streetEmpresa = $query->param(calle);
    $numberEmpresa = $query->param(numero);
    $coloniaEmpresa = $query->param(colonia);
    $cpEmpresa = $query->param(cp);
    $municipioEmpresa = $query->param(municipio);
    $estateEmpresa = $query->param(estado);
    $delegacionEmpresa = $query->param(delegacion);

    $contents="Los datos recibidos y actualizados son:<br>
<table>
<tr><td>ID:</td><td> $sid </tr>
<tr><td>Nombre de la empresa:</td><td> $nameEmpresa </tr>
<tr><td>Fecha de fundacion:</td><td> $dateFundation </tr>
<tr><td>Clasificacion:</td><td> $clasificacion </tr>
<tr><td>Giro :</td><td>$giro </tr>
<tr><td>Nombre del contacto:</td><td> $nameContact </tr>
<tr><td>Apellido Paterno del contacto:</td><td> $firstContact </tr>
<tr><td>Apellido Materno del contacto:</td><td> $secondContact </tr>
<tr><td colspan=2>DOMICILIO DE LA ORGANIZACION:</tr>
<tr><td>Calle: </td><td>$streetEmpresa </tr>
<tr><td>Numero:</td><td> $numberEmpresa </tr>
<tr><td>Colonia:</td><td> $coloniaEmpresa </tr>
<tr><td>Cp:</td><td> $cpEmpresa </tr>
<tr><td>Municipio:</td><td> $municipioEmpresa </tr>
<tr><td>Estado:</td><td> $estateEmpresa </tr>
<tr><td>Delegacion:</td><td> $delegacionEmpresa </tr>
</table>
";

    # $nameEmpresa <br>
    # $dateFundation <br>
    # $clasificacion <br>
    # $giro <br>
    # $nameContact <br>
    # $firstContact <br>
    # $secondContact <br>
    # $phoneContact <br>
    # $streetEmpresa <br>
    # $numberEmpresa <br>
    # $coloniaEmpresa <br>

```

```

#      $cpEmpresa <br>
#      $municipioEmpresa <br>
#      $estateEmpresa <br>
#      $delegacionEmpresa

Pg::doQuery($conn,"update empresa set
id_clas='$clasificacion',id_giro='$giro',id_est='$estateEmpresa',id_del='
$delegacionEmpresa',nombre='$nameContact',nombre_emp='$nameEmpresa',telef
ono='$phoneContact',a_paterno='$firstContact',a_materno='$secondContact',
fecha_fundacion='$dateFoundation',calle='$streetEmpresa',numero='$numberEm
presa',colonia='$coloniaEmpresa',cp='$cpEmpresa',municipio='$municipioEmp
resa' where id_empresa='$id'");
    $nameEmpresa = "";
    $dateFoundation = "";
    $clasificacion1 = "";
    $giro1 = "";
    $nameContact = "";
    $firstContact = "";
    $secondContact = "";
    $phoneContact = "";
    $streetEmpresa = "";
    $numberEmpresa = "";
    $coloniaEmpresa = "";
    $cpEmpresa = "";
    $municipioEmpresa = "";
    $estateEmpresal = "";
#      print "update empresa set
id_clas='$clasificacion',id_giro='$giro',id_est='$estateEmpresa',id_del='
$delegacionEmpresa',nombre='$nameContact',nombre_emp='$nameEmpresa',telef
ono='$phoneContact',a_paterno='$firstContact',a_materno='$secondContact',
fecha_fundacion='$dateFoundation',calle='$streetEmpresa',numero='$numberEm
presa',colonia='$coloniaEmpresa',cp='$cpEmpresa',municipio='$municipioEmp
resa' where id_empresa='$id'";

#para cambiar el valor del estatus
#      Pg::doQuery($conn,"select estatus from grupos where
id_grupo='$grupoUser',\@estatus);
#      $estatus=$estatus[0][0]+1;

#      Pg::doQuery($conn,"update grupos set estatus='$estatus' where
id_grupo='$grupoUser'");
    $boton="<form><input type=\"button\" value=\"Insertar Otra empresa\"
onclick=\"parent.location.href='opEmpInsert.pl?dato_id=$id_normal'\"
></form>";

    print $page_up.$contents.$boton.$botones.$page_down;

}

}#cierre del else

exit;

CODIGO DEL ARCHIVO opEmpDel.pl

```

```
#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

#####
#la siguiente funcion javascript sirve para cambiar la pagina de
#manera automatica una vez que el usuario ha elegido una operacion
#a realizar, enviando los datos necesarios.
#####

$JSCRIPT=<<EOF;

function spawnWin(options) {
  var url = options[
    options.selectedIndex
  ].value;
  parent.location.href= url;
}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;
print $query->header;
print start_html(-title=>'Personal Profile',
                 -script=>$JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);

@variables= &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up=$variables[0];
$page_down=$variables[1];
#botones para el regreso
$locacion1=$variables[2];
$locacion2=$variables[3];
$locacion3=$variables[4];
$botones=$variables[5];
#terminan los formatos de la pagina
```

```

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre:$datosUser[0][1]
#   AP:$datosUser[0][2]<br>
#   AM:$datosUser[0][3]<br>
#   LOG:$datosUser[0][4]<br>
#   PASS:$datosUser[0][5]<br>
#   TIPO:$datosUser[0][6]<br>
#   GRUPO:$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "1"){
    $contents="Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
";

    print $page_up.$contents.$botones.$page_down;
}else{
    if($accion ne "2" && $accion ne "3"){
        #modificacion de los datos dela empresa

        Pg::doQuery($conn,"select id_empresa,nombre_emp from empresa where
id_empresa not in (select id_empresa from auditoria)",\@empresas);
        print $page_up;

        if($dimension=@empresas > 0){
            if($dimension < 10){
                $longitud=$dimension;
            }else {
                $longitud=10;
            }
            print "<form>El siguiente listado solo presenta empresas a las que
no se han asignado auditorias. Si la empresa que busca no se encuentra en
el listado puede significar que tiene auditorias asignadas y no puede ser
borrada del Catalogo con el fin de conservar la integridad de los
datos.<br>
Seleccione la empresa a eliminar:
<div align=center>
<SELECT NAME=\"usuarios\" size=\"$longitud\"
onChange=\"spawnWin(this.options)\><br>";
            print "<OPTION>Elija:\n";
            $x = 0;
            while ( $empresas[$x][0] ne "" ) {

                print "<OPTION
value=opEmpDel.pl?dato_id=$id_normal&id=$empresas[$x][0]&accion=2>\n"$emp
resas[$x][1]\n\n";

                $x++;
            }
            print "</SELECT>
</div>

</form><br><br>";

            print $botones.$page_down;

```

```

}else{#si no hay empresas para modificar

    print "Por el momento no existen empresas con permiso para ser
editadas, verifique que la empresa por seleccionar no este siendo
utilizada por una auditoria";
    print $botones.$page_down;
}#cierre else de empresas a modificar

}elseif($accion eq "2"){
    print $page_up;

    $sid = $query->param(id);

    Pg::doQuery($conn,"delete from empresa where id_empresa='$sid'");

    $contents="La empresa se elimino con exito <br>
<form><input type=\"button\" value=\"Eliminar Otra empresa\"
onclick=\"parent.location.href='opEmpDel.pl?dato_id=$id_normal'\"
></form>";

    #////////

    print $contents.$botones.$page_down;

}elseif($accion eq "3"){
    $sid = $query->param(id);
    $nameEmpresa = $query->param(nombreEmp);
    $dateFundation = $query->param(fechaFundacion);
    $clasificacion = $query->param(clasificacion);
    $giro = $query->param(giro);
    $nameContact = $query->param(nombreContacto);
    $firstContact = $query->param(aPaterno);
    $secondContact = $query->param(aMaterno);
    $phoneContact = $query->param(telefono);
    $streetEmpresa = $query->param(calle);
    $numberEmpresa = $query->param(numero);
    $coloniaEmpresa = $query->param(colonia);
    $cpEmpresa = $query->param(cp);
    $municipioEmpresa = $query->param(municipio);
    $estateEmpresa = $query->param(estado);
    $delegacionEmpresa = $query->param(delegacion);

    $contents="Los datos recibidos y actualizados son:<br>
<table>
<tr><td>ID:</td><td> $sid </tr>
<tr><td>Nombre de la empresa:</td><td> $nameEmpresa </tr>
<tr><td>Fecha de fundacion:</td><td> $dateFundation </tr>
<tr><td>Clasificacion:</td><td> $clasificacion </tr>
<tr><td>Giro :</td><td>$giro </tr>
<tr><td>Nombre del contacto:</td><td> $nameContact </tr>
<tr><td>Apellido Paterno del contacto:</td><td> $firstContact </tr>
<tr><td>Apellido Materno del contacto:</td><td> $secondContact </tr>
<tr><td>Telefono del contacto:</td><td> $phoneContact </tr>
<tr><td colspan=2>DOMICILIO DE LA ORGANIZACION:</td>
<tr><td>Calle: </td><td>$streetEmpresa </tr>
<tr><td>Numero:</td><td> $numberEmpresa </tr>
<tr><td>Colonia:</td><td> $coloniaEmpresa </tr>
<tr><td>Cp:</td><td> $cpEmpresa </tr>
<tr><td>Municipio:</td><td> $municipioEmpresa </tr>

```

```

<tr><td>Estado:</td><td> $estateEmpresa </tr>
<tr><td>Delegacion:</td><td> $delegacionEmpresa </tr>
</table>
";

#   $nameEmpresa <br>
#   $dateFundation <br>
#   $clasificacion <br>
#   $giro <br>
#   $nameContact <br>
#   $firstContact <br>
#   $secondContact <br>
#   $phoneContact <br>
#   $streetEmpresa <br>
#   $numberEmpresa <br>
#   $coloniaEmpresa <br>
#   $cpEmpresa <br>
#   $municipioEmpresa <br>
#   $estateEmpresa <br>
#   $delegacionEmpresa

Pg::doQuery($conn,"update empresa set
id_clas='$clasificacion',id_giro='$giro',id_est='$estateEmpresa',id_del='
$delegacionEmpresa',nombre='$nameContact',nombre_emp='$nameEmpresa',telef
ono='$phoneContact',a_paterno='$firstContact',a_materno='$secondContact',
fecha_fundacion='$dateFundation',calle='$streetEmpresa',numero='$numberEm
presa',colonia='$coloniaEmpresa',cp='$cpEmpresa',municipio='$municipioEmp
resa' where id_empresa='$id'");
    $nameEmpresa = "";
    $dateFundation = "";
    $clasificacion1 = "";
    $giro1 = "";
    $nameContact = "";
    $firstContact = "";
    $secondContact = "";
    $phoneContact = "";
    $streetEmpresa = "";
    $numberEmpresa = "";
    $coloniaEmpresa = "";
    $cpEmpresa = "";
    $municipioEmpresa = "";
    $estateEmpresal = "";
#   print "update empresa set
id_clas='$clasificacion',id_giro='$giro',id_est='$estateEmpresa',id_del='
$delegacionEmpresa',nombre='$nameContact',nombre_emp='$nameEmpresa',telef
ono='$phoneContact',a_paterno='$firstContact',a_materno='$secondContact',
fecha_fundacion='$dateFundation',calle='$streetEmpresa',numero='$numberEm
presa',colonia='$coloniaEmpresa',cp='$cpEmpresa',municipio='$municipioEmp
resa' where id_empresa='$id'";

#para cambiar el valor del estatus
#   Pg::doQuery($conn,"select estatus from grupos where
id_grupo='$grupoUser',\@estatus);
#   $estatus=$estatus[0][0]+1;

#   Pg::doQuery($conn,"update grupos set estatus='$estatus' where
id_grupo='$grupoUser'");

```

```

    $boton="<form><input type=\"button\" value=\"Insertar Otra empresa\"
onclick=\"parent.location.href='opEmpInsert.pl?dato_id=$id_normal'\"
></form>";

```

```

    print $page_up.$contents.$boton.$botones.$page_down;

```

```

    }

```

```

}#cierre del else

```

```

exit;

```

CODIGO DEL ARCHIVO opEmpInsert.pl

```

#!/usr/bin/perl -U

```

```

#####
#Proyecto de Tesis Auditoria Administrativa

```

```

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

```

```

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

```

```

use Pg;

```

```

use CGI qw(:standard);

```

```

use pemes::inicializar;

```

```

#####

```

```

#funciones de javascript par filtrar la aparicion de los
#giros comerciales que estan repartidos en 3 grupos

```

```

#####

```

```

$JSCRIPT=<<EOF;

```

```

function traeClas(tipoClas) {
if (tipoClas == 'ninguna') {
    document.forma.accion.value=2;
    document.forma.submit();
}else{
    document.forma.accion.value=1;
    document.forma.submit();
    //var = options[options.selectedIndex].value;
    //var winArgs = \"top=10,left=10,\";
    //winArgs+= \"width=300,height=100\";
    //parent.location.href= url;
}
}

```

```

EOF

```

```

;

```

```

$conn = Pg::connectdb("dbname=auditoria user=pardo");

```

```

$query = new CGI;

print $query->header;

print start_html(-title=>'Insercion de Empresas Audit-admin',
                -script=>$JSCRIPT);

$accion = $query->param(accion);

$id_normal = $query->param(dato_id);

###
$nameEmpresa = $query->param(nombreEmp);
$dateFundation = $query->param(fechaFundacion);
$clasificacion = $query->param(clasificacion);
$giro = $query->param(giro);
$nameContact = $query->param(nombreContacto);
$firstContact = $query->param(aPaterno);
$secondContact = $query->param(aMaterno);
$phoneContact = $query->param(telefono);
$streetEmpresa = $query->param(calle);
$numberEmpresa = $query->param(numero);
$coloniaEmpresa = $query->param(colonia);
$cpEmpresa = $query->param(cp);
$municipioEmpresa = $query->param(municipio);
$estateEmpresa = $query->param(estado);
$delegacionEmpresa = $query->param(delegacion);

$fax = $query->param(fax);
$email = $query->param(email);
$puesto = $query->param(puesto);
$clasGen = $query->param(clasGen);
$clasGrupal = $query->param(clasGrupal);
$clasDet = $query->param(clasDet);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>

```

```

# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "1"){

    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>";

    print $page_up.$contents.$page_down;
}else{
    if($accion eq "2"){

#     $nameEmpresa <br>
#     $dateFundation <br>
#     $clasificacion <br>
#     $giro <br>
#     $nameContact <br>
#     $firstContact <br>
#     $secondContact <br>
#     $phoneContact <br>
#     $streetEmpresa <br>
#     $numberEmpresa <br>
#     $coloniaEmpresa <br>
#     $cpEmpresa <br>
#     $municipioEmpresa <br>
#     $estateEmpresa <br>
#     $delegacionEmpresa
#poner el que saca el valor maximo para designar el nuevo id

        Pg::doQuery($conn,"select (MAX(id_empresa)) FROM
empresa",\@maximo);
        $maximo[0][0] = $maximo[0][0]+1;

#hacer la insercion de la empresa
        Pg::doQuery($conn,"insert into empresa
values($maximo[0][0],'$clasificacion','$giro','$estateEmpresa','$delegaci
onEmpresa','$nameContact','$nameEmpresa','$phoneContact','$firstContact',
'$secondContact','$dateFundation','$streetEmpresa','$numberEmpresa','$col
oniaEmpresa','$cpEmpresa','$municipioEmpresa','$clasGen','$clasGrupal','$
clasDet','$fax','$email','$puesto')");

        Pg::doQuery($conn,"select id_empresa from empresa where
id_empresa=$maximo[0][0]",\@verifica);

        if (length($verifica[0][0])>0){
            $contents="Los datos recibidos e insertados son:<br>
<table>
<tr><td>Nombre de la empresa:</td><td> $nameEmpresa </td></tr>
<tr><td>Fecha de fundacion:</td><td> $dateFundation </td></tr>
<tr><td>Clasificacion:</td><td> $clasificacion </td></tr>
<tr><td>Giro :</td><td>$giro </td></tr>
<tr><td>Nombre del contacto:</td><td> $nameContact </td></tr>
<tr><td>Apellido Paterno del contacto:</td><td> $firstContact
</td></tr>
<tr><td>Apellido Materno del contacto:</td><td> $secondContact
</td></tr>
<tr><td>Telefono del contacto:</td><td> $phoneContact </td></tr>

```

```

<tr><td>Fax:</td><td> $fax </td></tr>
<tr><td>Email:</td><td> $email </td></tr>
<tr><td>Puesto:</td><td> $puesto </td></tr>

<tr><td colspan=2>DOMICILIO DE LA ORGANIZACION:</td></tr>
<tr><td>Calle: </td><td>$streetEmpresa </td></tr>
<tr><td>Numero:</td><td> $numberEmpresa </td></tr>
<tr><td>Colonia:</td><td> $coloniaEmpresa </td></tr>
<tr><td>Cp:</td><td> $cpEmpresa </td></tr>
<tr><td>Municipio:</td><td> $municipioEmpresa </td></tr>
<tr><td>Estado:</td><td> $estateEmpresa </td></tr>
<tr><td>Delegacion:</td><td> $delegacionEmpresa </td></tr>
</table>";
    }else{
        $contents="Ocurrio un error que impidio el guardado de datos,
verifique los los datos insertados cumplan con las reglas de
formato<br>";
    }

#    print "insert into empresa
values($maximo[0][0], '$clasificacion', '$giro', '$estateEmpresa', '$delegaci
onEmpresa', '$nameContact', '$nameEmpresa', '$phoneContact', '$firstContact',
'$secondContact', '$dateFundation', '$streetEmpresa', '$numberEmpresa', '$col
oniaEmpresa', '$cpEmpresa', '$municipioEmpresa')";

#para cambiar el valor del estatus
#    Pg::doQuery($conn,"select estatus from grupos where
id_grupo='$grupoUser', \@estatus);
#    $estatus=$estatus[0][0]+1;

#    Pg::doQuery($conn,"update grupos set estatus='$estatus' where
id_grupo='$grupoUser'");
    $boton = "<form><input type=\"button\" value=\"Insertar Otra
empresa\"
onclick=\"parent.location.href='opEmpInsert.pl?dato_id=$id_normal'\"
></form>";

    print $page_up.$contents.$boton.$botones.$page_down;

}else{
#cierre del if de accion
#para recibir los parametros temporales de la clasificacion de empresas

    Pg::doQuery($conn,"select id_clas,descripcion from clasificacion order
by id_clas", \@clasificacion);
    Pg::doQuery($conn,"select id_giro,descripcion from giro", \@giro);
    Pg::doQuery($conn,"select id_est,nombre from estado", \@estado);
    Pg::doQuery($conn,"select id_del,nombre from delegacion", \@delegacion);
    Pg::doQuery($conn,"select division,descripcion from
clasif_general", \@lasclasGen);

#terminar la parte que envia estos datos y los recibe para poner las
opciones
#    $nameEmpresa <br>
#    $dateFundation <br>
#    $clasificacion <br>
#    $giro <br>

```

```

$contentups_up = "
MODULO DE INSERCIÓN DE DATOS NUEVAS EMPRESAS AUDIT-ADMIN
<form name=\"forma\" action=\"opEmpInsert.pl\">
Para dar insertar empresas es necesario llenar los siguientes
datos:<br>
<table>
<tr><td><p>Nombre de la Empresa<br>
<textarea name=\"nombreEmp\" wrap=\"hard\" rows=\"3\"
cols=\"50\">$nameEmpresa</textarea></p></td>
<td><p>Fecha de fundaci&oacute;n \dd/mm/aaaa\"<br><input
type=\"text\" name=\"fechaFundacion\" maxlength=10 size=10
value=\"\$dateFundation\"></p></td></tr>";

print $page_up.$contentups_up;

print "<tr><td colspan=2>";
print "Clasificación General<br><select name=\"clasGen\"
onChange=\"javascript:traeClas('General');\">";

$x = 0;

if (length($clasGen)>0){
Pg::doQuery($conn,"select division,descripcion from clasif_general
where division='$clasGen',\@laclasGen");
print "<option value=$laclasGen[0][0] selected>$laclasGen[0][1]\n";
}
while($lasclasGen[$x][0] ne ""){

print "<option value=$lasclasGen[$x][0]>$lasclasGen[$x][1]\n";

$x ++;
}
print "</select></td></tr>";

#para las clasificaciones grupales y detalladas
if (length($clasGen)>0 ){
Pg::doQuery($conn,"select grupo,descripcion from clasif_grupal where
division='$clasGen',\@lasclasGrupal);

print "<tr><td colspan=2>";
print "Clasificación Grupal<br><select name=\"clasGrupal\"
onChange=\"javascript:traeClas('Grupal');\">";

$x = 0;
if (length($clasGrupal)>0){
Pg::doQuery($conn,"select grupo,descripcion from clasif_grupal
where grupo='$clasGrupal',\@laclasGrupal);
print "<option value=$laclasGrupal[0][0]
selected>$laclasGrupal[0][1]\n";
}
while($lasclasGrupal[$x][0] ne ""){

print "<option
value=$lasclasGrupal[$x][0]>$lasclasGrupal[$x][1]\n";

$x ++;
}
print "</select></td></tr>";
}

print "<tr><td colspan=2>";

```

```

if (length($clasGrupal)>0){
    Pg::doQuery($conn,"select id,descripcion from clasif_detalle where
grupo='$clasGrupal',\@lasclasDetalle);

    print "Clasificacion Detallada<br><select name=\"clasDet\">";

    $x = 0;

    while($lasclasDetalle[$x][0] ne ""){

        print "<option
value=$lasclasDetalle[$x][0]>$lasclasDetalle[$x][1]\n";

        $x ++;
    }
    print "</select></td>";
}

print "</tr>";
print "    <tr><td colspan=2>Clasificaci&oacute;n Tama&ntilde;o<br>
    <select name=\"clasificacion\">";

    $x = 0;

    while($clasificacion[$x][0] ne ""){

        print "<option
value=$clasificacion[$x][0]>$clasificacion[$x][1]\n";

        $x ++;
    }
    print "</select></td></tr>";

#    $nameContact <br>
#    $firstContact <br>
#    $secondContact <br>
#    $phoneContact <br>
#    $streetEmpresa <br>
#    $numberEmpresa <br>
#    $coloniaEmpresa <br>
#    $cpEmpresa <br>
#    $municipioEmpresa <br>
#    $estateEmpresa <br>
#    $delegacionEmpresa
    $contents_down = "<tr><td><hr><p>Nombre del Contacto<br>
    <input type=\"text\" name=\"nombreContacto\" maxlength=\"50\"
size=\"50\" value=\"$nameContact\"></p>
    </td><td><hr><p>Apellido Paterno<br>
    <input type=\"text\" name=\"aPaterno\" maxlength=\"50\" size=\"50\"
value=\"$firstContact\"></p>
    </td></tr><tr><td><p>Apellido Materno<br>
    <input type=\"text\" name=\"aMaterno\" maxlength=\"50\" size=\"50\"
value=\"$secondContact\"></p>
    </td><td><p>Tel&eacute;fono<br>
    <input type=\"text\" name=\"telefono\" maxlength=\"21\" size=\"21\"
value=\"$phoneContact\">
    <input type=\"hidden\" name=\"dato_id\" value=$id_normal>
    <input type=\"hidden\" name=\"accion\" value=\"\"></td></tr>
    <input type=\"hidden\" name=\"tipoClas\" value=\"\"></td></tr>

```

```

        <tr><td><p>Fax<br>
        <input type=\"text\" name=\"fax\" maxlength=\"21\" size=\"21\"
value=\"\${fax}\"></p>
        </td><td><p>E-mail<br>
        <input type=\"text\" name=\"email\" maxlength=\"300\" size=\"50\"
value=\"\${email}\">
        </td></tr>
        <tr><td colspan=2><p>Puesto<br>
        <input type=\"text\" name=\"puesto\" maxlength=\"255\" size=\"50\"
value=\"\${puesto}\"></p>
        </td></tr>

        <tr><td>
        <hr>Domicilio de la Empresa<br>Calle<br>
        <input type=\"text\" name=\"calle\" maxlength=\"50\" size=\"50\"
value=\"\${streetEmpresa}\"></td>
        <td><hr>Nº<br>
        <input type=\"text\" name=\"numero\" maxlength=\"4\" size=\"4\"
value=\"\${numberEmpresa}\"></td>
        </tr><tr><td>Colonia<br>
        <input type=\"text\" name=\"colonia\" maxlength=\"50\" size=\"50\"
value=\"\${coloniaEmpresa}\"></td>
        <td>Código Postal<br>
        <input type=\"text\" name=\"cp\" maxlength=\"5\" size=\"5\"
value=\"\${cpEmpresa}\"></td></tr>
        <tr><td>Municipio<br>
        <input type=\"text\" name=\"municipio\" maxlength=\"50\" size=\"50\"
value=\"\${municipioEmpresa}\"></td></tr>";

print \$contents_down;
print " <tr><td>Estado<br>
        <select name=\"estado\">";

        $x = 0;
        while(\$estado[$x][0] ne ""){

            print "<option value=\${estado[$x][0]}>\${estado[$x][1]}\n";

            $x ++;
        }

        print "</select></td><td>Delegaci&eacute;n<br><select
name=\"delegacion\">
        <option value=\"0\">--solo en DF--";
        $x = 0;

        while(\$delegacion[$x][0] ne ""){

            print "<option value=\${delegacion[$x][0]}>\${delegacion[$x][1]}\n";

            $x ++;
        }

        print "</select></td></tr>
        <tr><td colspan=2 align=\"center\"><input type=\"button\"
value=\"Insertar Empresa\" onClick=\"javascript:traeClas('ninguna');\">
</td></tr>
        </form></table>";
        print \$botones.$page_down;
    }#cierre del else de adentro
}#cierre del else

```

exit;

CODIGO DEL ARCHIVO opEmpresas.pl

```
#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

#####
#la siguiente funcion javascript sirve para cambiar la pagina de
#manera automatica una vez que el usuario ha elegido una operacion
#a realizar, enviando los datos necesarios.
#####

$JSCRIPT=<<<EOF;

var nombreCgi="";
var operacion="";

function spawnWin(options) {
  var url = options[
    options.selectedIndex
  ].value;
  if( nombreCgi!=""){
    parent.location.href=nombreCgi+url;

  }else{
    alert("Requiere elegir una operacion de ALTA, BAJA o CAMBIO
previamente");
  }
}

}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;
print $query->header;
print start_html(-title=>'Eleccion de Operaciones con Empresas',
                 -script=>$JSCRIPT);

$action = $query->param(accion);
$id_normal = $query->param(dato_id);
```

```

@variables= &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up=$variables[0];
$page_down=$variables[1];
#botones para el regreso
$locacion1=$variables[2];
$locacion2=$variables[3];
$locacion3=$variables[4];
$botones=$variables[5];
#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "1"){
    $contents="Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>

    ";

    print $page_up.$contents.$page_down;
}else{

    #esto despliega el menu de las operaciones con el usuario si aun no se
ha definido una accion a realizar
#el select le pasa a la variable del javascript el nombre del catalogo
a modificar y el id del usuario
    $contents="
    <table align=center><tr>
    <td>
    <form method=post>
    <p>Elija la accion a realizar:</p>
    <input type=\"hidden\" name=\"dato_id\" value=\"'$id_normal'\">
    <input type=\"radio\" name=\"accionEmpresa\"
onClick=\"nombreCgi='insertEmpresa.pl'\\">Alta<br>
    <input type=\"radio\" name=\"accionEmpresa\"
onClick=\"nombreCgi='delEmpresa.pl'\\">Baja<br>
    <input type=\"radio\" name=\"accionEmpresa\"
onClick=\"nombreCgi='change Empresa.pl'\\">Cambio
    </td>
    <td>

    <select name=\"act_usuario\" onChange=\"spawnWin(this.options)\"
size=4>\n

```

```

        <option value=\"subfunciones&dato_id=$id_normal\">Cat&aacute;logo de
subfunciones</option>\n
        <option value=\"factores&dato_id=$id_normal\">Cat&aacute;logo de
factores</option>\n
        <option value=\"clasificaci&oacute;n&dato_id=$id_normal\">Cat&aacute;logo
de clasificaci&oacute;n</option>\n
        <option value=\"grupos&dato_id=$id_normal\">Cat&aacute;logo de
grupos</option>\n
    </select>
</form>
</td></tr></table>
";

```

```

    print $page_up.$contents.$botones.$page_down;
}#cierre del else principal
exit;

```

CODIGO DEL ARCHIVO opUsuarios.pl

```

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

#####
#la siguiente funcion javascript sirve para cambiar la pagina de
#manera automatica una vez que el usuario ha elegido una operacion
#a realizar, enviando los datos necesarios.
#####
$JSCRIPT=<<EOF;

function spawnWin(options) {
    var url = options[options.selectedIndex].value;
    parent.location.href= url;
}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;
print $query->header;

```

```

print start_html(-title=>'Personal Profile',
                 -script=>$_JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);

@variables = &prueba();
#variables que crean el formato de la pagina que se despliega
$page_up = $variables[0];
$page_down = $variables[1];
#botones para el regreso
$locacion1 = $variables[2];
$locacion2 = $variables[3];
$locacion3 = $variables[4];
$botones = $variables[5];
#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal",\@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>

if ($id_normal eq "" || $datosUser[0][6] ne "3"){
    $contents = "Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    llego $id_normal<br>
tipo $datosUser[0][6]";

    print $page_up.$contents.$botones.$page_down;
}else{
    if($accion ne "1" && $accion ne "2" && $accion ne "3"){
        #esto despliega el menu de las operaciones con el usuario si aun no
se ha definido una accion a realizar

        $contents="<table align=center><tr><td>
        <li><A HREF="\#"\"
OnClick="\window.open('opUsuarios.pl?accion=1\&dato_id=$id_normal','varia
ble','width=600,height=450,scrollbars=yes,toolbar=no')\">Nuevo
Usuario</A><br>
        <li><A HREF="\#"\"
OnClick="\window.open('opUsuarios.pl?accion=2\&dato_id=$id_normal','varia
ble','width=600,height=450,scrollbars=yes,toolbar=no')\">Modificar datos
del usuario</A><br>
        <li><A HREF="\#"\"
OnClick="\window.open('opUsuarios.pl?accion=3\&dato_id=$id_normal','varia
ble','width=600,height=450,scrollbars=yes,toolbar=no')\">Borrar
Usuario</A><br>

```

```

</td></tr></table>\n";

    print $page_up.$contents.$botones.$page_down;
}elsif ($accion eq "1"){
#insercion de usuario
Pg::doQuery($conn,"select id_grupo,descripcion from grupos where
estatus <> 0",\@grupos);
Pg::doQuery($conn,"select * from tipoUsuario",\@tiposUsuario);

print $page_up;
$contents_up="<table align=center><tr><td>
MODULO DE ALTA DE USUARIOS AUDIT-ADMIN
<form name=InsertaUser action=insertUser.pl>
Para dar de alta nuevos usuarios es necesario insertar los
siguientes datos:<br>
<table>
<tr><td><p>Nombre del usuario<br>
<input type=\"text\" name=\"nombre\" maxlength=\"30\"
size=\"30\"></p></td>
<td><p>Apellido Paterno<br>
<input type=\"text\" name=\"apaterno\" maxlength=\"30\"
size=\"30\"></p>
</td></tr><tr><td><p>Apellido Materno<br>
<input type=\"text\" name=\"amaterno\" maxlength=\"30\"
size=\"30\"></p>
</td><td><p>Login<br>
<input type=\"text\" name=\"login\" maxlength=\"20\"
size=\"20\"></p>
</td></tr><tr><td><p>Password<br>
<input type=\"password\" name=\"password\" maxlength=\"30\"
size=\"30\">
<input type=\"hidden\" name=\"dato_id\" value=$id_normal>
</td><td>";
print $contents_up;
#seleccion de grupo para el usuario
print "Seleccione un grupo al que el usuario pertenecer&aacute;:";
<SELECT NAME=\"grupo\"><br>";

$x = 0;
while ( $grupos[$x][0] ne "" ) {

    print "<OPTION value=$grupos[$x][0]>$grupos[$x][1]\n";

    $x ++;
}
print "</SELECT><br><br>";
#seleccion de tipo de usuario
print "</td></tr><td>Seleccione el tipo de usuario que se
dar&aacute; de alta:
<SELECT NAME=\"TipoUsuario\"><br>";

$x = 0;
while ( $tiposUsuario[$x][0] ne "" ) {

    print "<OPTION
value=$tiposUsuario[$x][0]>$tiposUsuario[$x][1]\n";

    $x ++;
}
print "</SELECT><br><br>";
$contents_down = "</td></tr>

```

```

        <tr><td align=right><input type=submit value=\"Insertar
Usuario\"></td>
        <td align=left><input type=button value='Cancelar'
onclick='self.close()''>
        </td></tr></table>
    </form>
    </td></tr></table>\n";

    print $contents_down.$page_down;

}elsif ($accion eq "2"){
    #modificacion de los datos del usuario

    Pg::doQuery($conn,"select apaterno,amaterno,nombre,id_usuario from
usuarios order by apaterno",\@usuarios);

    print $page_up;

    if($dimension = @usuarios < 10){
        $longitud = $dimension;
    }else {
        $longitud = 10;
    }

    print "<form>Seleccione el usuario a modificar:
        <SELECT NAME=\"usuarios\" size=\"$$longitud\"
onChange=\"spawnWin(this.options)\\" ><br>";

    $x = 0;

    while ( $usuarios[$x][3] ne "" ) {

        print "<OPTION
value=modifUser.pl?dato_id=$id_normal&id=$usuarios[$x][3]>\"$usuarios[$x
][0] $usuarios[$x][1] $usuarios[$x][2]\"<br>";

        $x ++;
    }

    print "</SELECT></form><br><br>";

    print $page_down;

}elsif ($accion eq "3"){
    Pg::doQuery($conn,"select apaterno,amaterno,nombre,id_usuario from
usuarios order by apaterno",\@usuarios);

    print $page_up;

    if($dimension = @usuarios < 10){
        $longitud = $dimension;
    }else {
        $longitud = 10;
    }

    print "<form>Seleccione el usuario a modificar:
        <SELECT NAME=\"usuarios\" size=\"$$longitud\"
onChange=\"spawnWin(this.options)\\" ><br>";

    $x = 0;

```

```

        while ( $usuarios[$x][3] ne "" ) {
            print "<OPTION
value=delUser.pl?dato_id=$id_normal&id=$usuarios[$x][3]>\$usuarios[$x][
0] $usuarios[$x][1] $usuarios[$x][2]\`\n";
            $x ++;
        }
        print "</SELECT></form><br><br>";
        print $page_down;
    }
}#cierre del else
exit;

```

CODIGO DEL ARCHIVO valida.pl

```

#!/usr/bin/perl -U
#####
#Proyecto de Tesis Auditoria Administrativa
#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM
#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####
use Pg;
use CGI qw(:standard);

#####
#la siguiente funcion javascript sirve para cambiar la pagina
#de manera automatica una vez que el usuario ha elegido una
#operacion a realizar, enviando los datos necesarios.
#####
$JSCRIPT=<<EOF;

function spawnWin(options) {
    var url = options[
        options.selectedIndex
    ].value;
    var winArgs = \ "top=10,left=10,\ ";
    winArgs+= \ "width=300,height=100\ ";
    parent.location.href= url;
}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;
print $query->header;

```

```

print start_html(-title=>'Bienvenido a Audit-Admin',
                 -script=>$_JSCRIPT);

$cid_normal= $query->param(dato_id);

$accion = $query->param(accion);
$page_up= " <html>\n
  <body onload=\"document.forma.reset()\" bgcolor=white
background=\"../Tesis/Imagenes/bglogito.jpg\">\n
  <table align=center><tr><td>\n";
$page_down="
  </td></tr></table>\n
  </body>\n
  </html>\n";
#botones para regresar
$locacion1="valida.pl";
$locacion2="../Tesis/";

$botones="<div align=center>
  <form method=post>
  <input type=\"button\" value=\"Cerrar Sesion\"
onclick=\"parent.location.href='$locacion1'\">\n
  <input type=\"button\" value=\"Pagina del proyecto\"
onclick=\"parent.location.href='$locacion2'\">\n
  </form></div>";

#si la accion es igual a 1 es que apenas hay que recoger el login y el
password
if ($accion ne "2" && $cid_normal eq ""){
  $contents=" <form name=forma method=post action=valida.pl> \n
  Escriba su login :<br>\n
  <input type=text name=login maxlength=\"20\"><br> \n
  Escriba su password :<br>\n
  <input type=password name=password maxlength=\"40\"><br>\n
  <input type=submit value=\"Enviar\" >\n
  <input type=\"hidden\" name=accion value=\"2\" >\n
  </form>\n";

  print $page_up.$contents.$page_down;
}elseif ($accion eq "2" || $cid_normal ne ""){
  $login = $query->param(login);
  Pg::doQuery($conn,"select * from usuarios where login = '$login'
",\@datosUser);
  if ($cid_normal ne ""){
    Pg::doQuery($conn,"select * from usuarios where id_usuario
='$cid_normal' ",\@datosUser);
  }
  if ($datosUser[0][0] eq ""){
    $contents="No fue encontrado ningun usuario con esos
datos<br>\n
  <a href=valida.pl?accion=1>Intentelo de nuevo</a> <br>
  ";
    print $page_up.$contents.$botones.$page_down;
    exit;
  }
}

$pass = $query->param(password);
if ($cid_normal ne ""){

```

```

    $pass=$datosUser[0][5];
    $pass=~ s/ //g;#para quitarle los espacios
}

$datosUser[0][5]=~ s/ //g;#para quitarle los espacios

if($pass ne $datosUser[0][5]){
    $contents="No fue encontrado ning&uacute;n usuario con esos
datos<br>\n
llego \"$pass\" y era \"$datosUser[0][5]\n"
<a href=valida.pl?accion=1>Intentelo de nuevo</a><br>
llego$id_normal ";
    print $page_up.$contents.$botones.$page_down;
    exit;
}
# el id es:$datosUser[0][0]
# el nombre$datosUser[0][1]
# AP$datosUser[0][2]<br>
# AM$datosUser[0][3]<br>
# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>
# $id_encryptado=rsa::encripta($datosUser[0][0]);#encripto el id del
usuario que es lo unoco que requiero para rescatar sus datos en todo el
proceso de la auditoria.
    $id_encryptado=$datosUser[0][0];

    if ($datosUser[0][6] eq "1" ){
        $actividades="<select name=\"act_Usuario\"
onChange=\"spawnWin(this.options)\" size=5>\n
<option value=\"opEmpInsert.pl?dato_id=$id_encryptado\">Alta de
empresas</option>\n
<option
value=\"opEmpChange.pl?dato_id=$id_encryptado\">Modificacion de datos de
empresas</option>\n
<option value=\"opEmpDel.pl?dato_id=$id_encryptado\">Baja de
empresas</option>\n
<option
value=\"opAuditorias.pl?dato_id=$id_encryptado\">Operaciones con
auditorias</option>\n
<option
value=\"opReportes.pl?dato_id=$id_encryptado\">Generaci&oacute;n de
Reportes</option>\n
</select>";
    }elseif ($datosUser[0][6] eq "2" ){
        $actividades="<select name=\"act_Usuario\"
onChange=\"spawnWin(this.options)\" size=4>\n
<option
value=\"opAuditores.pl?accion='lista'&dato_id=$id_encryptado\">Ver lista
de Aditores</option>\n
<option value=\"opAuditorias.pl?dato_id=$id_encryptado\">Ver
Auditor&iacute;as</option>\n
<option
value=\"opReportes.pl?dato_id=$id_encryptado\">Generaci&oacute;n de
Reportes</option>\n
<option
value=\"obsevaciones.pl?dato_id=$id_encryptado\">Observaciones al
auditor</option>\n

```

```

        </select>";
    }elseif ($datosUser[0][6] eq "3" ){
        $actividades="<select name=\"act_Usuario\"
onChange=\"spawnWin(this.options)\\" size=4>\n
        <option
value=\"opUsuarios.pl?dato_id=$id_encriptado\">Operaciones con
usuarios</option>\n
        <option
value=\"opCatalogos.pl?dato_id=$id_encriptado\">Modificaciones a los
cat&aacute;logos</option>\n
        <option
value=\"management.pl?accion=1&dato_id=$id_encriptado\">Modificaciones a
empresas y auditorias</option>\n
        <option
value=\"management.pl?accion=2&dato_id=$id_encriptado\">Mantenimiento del
sistema</option>\n
        </select>";
    }

    $contents="Bienvenido a audit-admin:
<br>$datosUser[0][1] $datosUser[0][2] $datosUser[0][3]<br>
<form name=forma method=post action=.pl> \n
Elija una de las siguientes actividades a realizar:<br>
$actividades<br>

<input type=hidden name=\"dato_id\" value=\"\$id_encriptado\">

</form>\n
";
    print $page_up.$contents.$botones.$page_down;

}#cierre del else if cuando accion es distinto de 1
exit;

CODIGO DEL ARCHIVO workAudit.pl

#!/usr/bin/perl -U

#####
#Proyecto de Tesis Auditoria Administrativa

#Copyright (C)2002 Cynthia Gabriela Pardo Orozco FCA-UNAM

#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#####

use Pg;
use CGI qw(:standard);
use pemes::inicializar;

$JSCRIPT=<<EOF;
var nombreCgi="";
var operacion="";

```

```
function spawnWin(options) {
    var url = options[options.selectedIndex].value;

    if( nombreCgi != ""){

        document.workAudit.action = nombreCgi;

        document.workAudit.audit.value = url;

        document.workAudit.submit();

    }else{

        alert("Requiere elegir una operacion previamente");

    }
}

EOF
;

$conn = Pg::connectdb("dbname=auditoria user=pardo");

$query = new CGI;
print $query->header;

print start_html(-title=>'Seleccion de Auditoria y Operacion Audit-
Admin',
                -script=>$JSCRIPT);

$accion = $query->param(accion);
$id_normal = $query->param(dato_id);

@variables= &prueba2("workAudit");
#variables que crean el formato de la pagina que se despliega
$page_up=$variables[0];
$page_down=$variables[1];
#botones para el regreso
$locacion1=$variables[2];
$locacion2=$variables[3];
$locacion3=$variables[4];
$botones=$variables[5];
#terminan los formatos de la pagina

Pg::doQuery($conn,"select * from usuarios where id_usuario
=$id_normal", \@datosUser);

#si no llego el id o llego pero no es un usuario del tipo administrador,
se desecha
#   el id es:$datosUser[0][0]
#   el nombre$datosUser[0][1]
#   AP$datosUser[0][2]<br>
#   AM$datosUser[0][3]<br>
```

```

# LOG$datosUser[0][4]<br>
# PASS$datosUser[0][5]<br>
# TIPO$datosUser[0][6]<br>
# GRUPO$datosUser[0][7]<br>
#####
#Este es para checar el tipo de usuario
#####

if ($id_normal eq "" || $datosUser[0][6] ne "1" ){
    $contents="Los datos de validacion de usuario para ejecutar este
modulo no llegaron completos o no fueron insertados, consulte con el
administrador del sistema o <a href=valida.pl?accion=1>vuelva a
intentar.</a>\n<br>
    <br>
    ";

    print $page_up.$contents.$page_down;
}else{
#####
#Si accion viene sin ningun contenido
#####
    if($accion ne ""){
#esto no se ni para que lo puse lo mas posible es que hay que quitarlo
        print $page_up."hola";
        print $botones.$page_down;
    }else{
#no viene parametro es que apenas se va a elegir una auditoria con que
trabajar y la operacion a realizar con esa auditoria, las operaciones
pueden ser:
#Elegir Subfuncion del catalogo
#Asignar un porcentaje de importancia a cada subfuncion
#Elegir un factor del catalogo
#Asignar distribucion de puntos
#####
#Definir preguntas
#Evaluar preguntas

        Pg::doQuery($conn,"select id_empresa,fecha_inicio,id_auditoria from
auditoria where id_usuario=$id_normal",\@empresas);#trae las empresas ya
asignadas a las auditorias de un usuario

        $x = 0;
        $num_empresas = @empresas ;
        if ($num_empresas > 10){
            $num=10;
        }else{
            $num=$num_empresas;
        }
        if($num > 0){
            $contents_up = <<FIN;
            - MODULO DE OPERACIONES CON AUDITORIAS - AUDIT-ADMIN
            <p>Operaci&oacute;n a realizar<br>
            <form method=post name="workAudit">
            <table>
            <tr><td>
            <input type="hidden" name="dato_id"
value="$id_normal">
            <input type="hidden" name="audit" value="">

```

```

        <input type="radio" name="accionAuditoria"
onClick="nombreCgi='defineSub.pl'">
        1. Definir subfunciones a auditar <br>
        <input type="radio" name="accionAuditoria"
onClick="nombreCgi='defineWeight.pl'">
        2. Definir peso de cada subfuncion <br>
        <input type="radio" name="accionAuditoria"
onClick="nombreCgi='defineFactor.pl'">
        3. Definir factores a evaluar <br>
        </td>
        <td>
        <input type="radio"
name="accionAuditoria" onClick="nombreCgi='defineDistrib.pl'">
        4. Distribucion de puntos <br>
        <input type="radio" name="accionAuditoria"
onClick="nombreCgi='defineQuestion.pl'">
        5. Definir preguntas <br>
        <input type="radio" name="accionAuditoria"
onClick="nombreCgi='evalQuestion.pl'">
        6. Evaluar preguntas<br>
        </td><tr></table>

        Elija la Auditoria con que se trabajara:<br>

        <SELECT NAME="auditoria" size="$num"
onChange=spawnWin(this.options)><br>
FIN

        print $page_up.$contents_up;

        while ( $empresas[$x][0] ne "" ) {

                Pg::doQuery($conn,"select nombre_emp from empresa where
id_empresa='$empresas[$x][0]','\@empresas2);#trae las empresas ya
asignadas a las auditorias de un usuario
                #imprime las empresas existentes
                # $empresas[$x][0]=~ s/ //g;
                # $empresas[$x][1]=~ s/ //g;
                print "<OPTION value=$empresas[$x][2]>$empresas[$x][1] /
$empresas2[0][0]\n";

                $x++;
        }

        $contents_down="</SELECT><br><br>";

        print $contents_down.$botones.$page_down;
    }else {#cuando no existen auditorias creadas.
        print $page_up;
        print "No hay auditorias existentes<br>
        Requiere crear una nueva auditoria previamente";
        print $empresas[$x][0];
        print $botones.$page_down;
    }
}
}#cierre del else

exit;

```