

41132
25



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGÓN**

**“SISTEMA DE ADMINISTRACIÓN DE
RECURSOS HUMANOS”**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A N:
TERESITA DE JESÚS GARCÍA VÁZQUEZ
AARÓN EDUARDO SÁNCHEZ DELGADILLO

TESIS CON
FALLA DE ORIGEN

ASESOR: M. EN I. JUAN CARLOS ROA BEIZA

MÉXICO

SEPTIEMBRE 2003

1



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.

**TESIS
CON
FALLA DE
ORIGEN**

AGRADECIMIENTOS

Para todas las acciones que realizo, está primero la mano de Dios, por lo que quiero agradecer su infinita bondad al haberme permitido llegar a este proyecto de tesis....

A toda mi familia:

Por el gran apoyo que me han brindado en todos los aspectos, en situaciones que se han presentado buenas y malas a lo largo de este tiempo que ha transcurrido para la realización de este trabajo... no olvido los consejos oportunos de mis padres, mis hermanos, mis cuñados...a pesar de que ha transcurrido más tiempo, no me arrepiento y ha valido la pena... agradezco las motivaciones que en algún momento me hicieron llegar, así como las llamadas de atención para seguir adelante hasta el final...no tienen idea cuánto me han servido, no sólo en mi vida profesional, sino también en mi realización como persona, gracias por estar siempre conmigo y por ser mi familia...mis papás: Luis García Quezada y Victoria Vázquez Galaviz, mis hermanos: Elizabeth, Luis Martín, Leonardo, Lupita, Alfredo, Edgar, Dann...Sergio, Lulú, Hilda; mis sobrinos: Jair, Jessica, Sergio, Fernando, Mariana, Carlitos, mi tía Engracia quien me está viendo desde el cielo.

A mis amigos:

Tampoco se me olvidará la gran ayuda y la solidaridad de mis amigos, los cuales han venido a formar parte muy importante en estas circunstancias y por siempre, quiero decirles que cada uno de ustedes ha aportado grandes motivaciones y armas que sirvieron para fortalecerme más...los quiero y los aprecio de verdad...todos tienen igual relevancia así que, no importa el orden en que los mencione, ustedes lo saben, gracias a: Alejandro Rosas, Leonardo Hernández, Sergio Olguin, Angeles González, René Martínez, Graciela Quintanar, Margarita Esquivel, Marcela Miranda, Israel Cabrera; gracias Javier por tu amistad, cariño y apoyo incondicional que me has brindado desde que me conoces hasta estos momentos, no tendré con que pagártelo, que Dios te bendiga.

A mis compañeros de trabajo:

A la Contraloría Interna del IFE, no saben cómo agradezco el apoyo espontáneo que se me otorgó, además por haber creído en mí siempre y por la confianza que existió...gracias: C.P. Carlos Durán, Auditoría Técnica, al Lic. Alfredo Garciamoreno, a la Sra. Lea Castañeda, a Pily, Don Charlie, Socorro Rosas, Auditoría Financiera y demás compañeros que siempre me motivaron a seguir de frente. Gracias.

También agradezco la ayuda del PAT (Programa de Apoyo a la Titulación), el apoyo del M. en I. Juan Carlos Roa Beiza, a la Jefatura de carrera de Ingeniería en Computación y para la ENEP Aragón en general, gracias.

*Quiero manifestar que, lo que he reunido a lo largo de mi vida y hasta este momento ha sido y será UN GRAN TESORO...por lo que es mi deseo dedicar este trabajo escrito a todas estas personas que he mencionado, de todo corazón,
MUCHAS GRACIAS!!*

TERESITA DE JESÚS GARCÍA VÁZQUEZ

AGRADECIMIENTOS

Para mi Mamá y Hermana:

*Mamá, describir mi profundo amor y agradecimiento hacia ti sería imposible. Muchas gracias por ser mi madre y padre, estar siempre conmigo y enseñarme tantas e innumerables cosas a lo largo de mi vida. Sin ti no hubiera sido posible llegar hasta donde estoy. Para ti todo mi amor, aprecio y respeto que alguien podría recibir.
Te amo.*

*Vane, estoy muy agradecido de que seas mi hermana y crecer con una niña tan linda, dedicada e inteligente como tu. Sigue adelante hermana que siempre estaré contigo. Gracias por tu apoyo, compañía y buenos momentos.
Te amo.*

A ambas: Gracias por ser mi familia.

Para mi Papá:

Gracias por tu apoyo y buenos consejos. Te quiero.

Para Cynthia:

*Gracias por empezar conmigo la etapa profesional mas importante de mi vida, te agradezco todo el apoyo que me has brindado, los buenos momentos, tu comprensión y compañía pero sobre todo, tu cariño.
Te amo.*

Familiares y amigos:

Para mis Abuelos Ernestina, Felipa y Pedro(QEPD). A mis tíos Miguel, Eva, Fernando, María Elena, María, Nelson . A mis primos Stephani, Fernando, Vanessa, Renan, Nelson, Rosario y Rocío. Conchita, y todos mis familiares.

Para mis compañeros y conocidos de la carrera, la ENEP-Aragón y Ciudad Universitaria.

Para la Dirección de Sistemas. Gracias a la Lic. Laura Llyen Galicia, Alma por su gran apoyo, Doris, Josefina, Fabiola, y todas las personas involucradas en esta Dirección. A todos los instructores y compañeros de la promoción 24 del plan de

TESIS CON
FALLA DE ORIGEN

becarios.

*Para el CEPE. A mis compañeros y conocidos.
A la Ing. Yolanda Uribe Mirón por todo su apoyo y buenos consejos.*

Para mi Universidad y la ENEP- Aragón. A todos mis profesores en la carrera y revisores de tesis Ing. Gladis Fuentes, M. en C. Jesús Díaz Barriga, Ing. Rodolfo Vázquez, Ing. Cesar Francisco Germán.

Para el PAT y nuestro asesor de tesis M. en I. Juan Carlos Roa Beiza.

Para Algorítmica. Act. Javier Medina Mora, Ing. Roberto Cordiba, Adriana, Mario, Juan Manuel y Felipe.

Así como a Carlos Otero, Percy Arenas, Jossiane Rodríguez, Sergio González, Francisco Rodríguez, Sahi Camacho, Evelyn López, Humberto, Roberto Escobedo, Paola, Emilio, Ana, Tania, Mauro, Aglae, Armando, Juan Carlos, Adán, Daniel, Quique, Aurelio, Daniela, Alfredo, Ernesto, Iván, Katia, Iván Facio, Jhon y Jhan Fuentes, Miguel, Aurelio, Roberto, Enrique Betanzos, Salvador Cáceres, Sandro, Oliver, Oliver, Eduardo, Patiño, Ramón Del Villar, Vero, Chio, Mónica, Lety, Héctor y todas las personas que de una u otra forma me han acompañado en mi carrera profesional.

AARÓN EDUARDO SÁNCHEZ DELGADILLO

**TESIS CON
FALLA DE ORIGEN**

INDICE TEMÁTICO.

INTRODUCCIÓN.....	1
1.1 FILOSOFÍA, MISIÓN, VISIÓN Y OBJETIVOS DE CAPITAL HUMANO S.A....	4
1.1.1 Filosofía de la empresa.....	4
1.1.2 Misión de Capital Humano S.A.....	7
1.1.3 Visión de Capital Humano S.A.....	7
1.1.4 Objetivos de Capital Humano S.A.....	7
1.2 ATRIBUCIONES DE LA EMPRESA.....	8
1.2.1 Recursos humanos.....	9
1.2.2 Identificación de gente clave.....	9
1.2.3 Seguimiento de performance (desempeño) de programas de recursos humanos.....	9
1.2.4 Encuestas de clima organizacional.....	10
1.2.5 Sistemas de evaluación de desempeño.....	10
1.3 DIRECCIONES INVOLUCRADAS Y POLÍTICAS.....	10
1.3.1 Direcciones involucradas.....	10
1.3.2 Políticas.....	13
1.3.2.1 Políticas para los empleados.....	13
1.3.2.2 Políticas para los empleadores.....	13
1.4 PROCEDIMIENTOS ACTUALES DEL DEPARTAMENTO DE RECURSOS HUMANOS.....	14
2.1 PROBLEMÁTICA ACTUAL.....	22
2.1.1 Proceso de reclutamiento y selección y su problemática asociada.....	24
2.2 REQUERIMIENTOS DEL DEPARTAMENTO DE RECURSOS HUMANOS.....	26
2.2.1 Requerimientos generales.....	27
2.2.2 Requerimientos particulares.....	27
2.3 BÚSQUEDA Y ANÁLISIS DE LA INFORMACIÓN.....	36
2.4 IDENTIFICACIÓN DEL PROBLEMA POR ÁREAS.....	39
2.4.1 Subdirección Administrativa.....	39
2.4.1.1 Área Ejecutiva.....	39
2.4.1.2 Área de Recursos Humanos.....	40
2.4.1.3 Área de Finanzas.....	41
2.4.2 Subdirección Operativa.....	42
2.4.2.1 Área de Ventas.....	42
2.4.2.2 Área de Servicios.....	42
2.5 OPCIONES DE SOLUCIÓN Y ELECCIÓN DE LA ÓPTIMA.....	43
2.5.1 Elijiendo una base de datos.....	43

TESIS CON
 FALLA DE ORIGEN

TESIS CON
 FALLA DE ORIGEN

2.5.1.1 MySQL.....	44
2.5.1.2 PostgreSQL.....	45
2.5.1.3 Mini SQL.....	47
2.5.2 Front-End.....	49
2.5.2.1 PERL.....	49
2.5.2.2 PHP.....	50
2.5.2.3 JSP.....	51

3.1 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LAS BASES DE DATOS RELACIONALES..... 56

3.1.1 Concepto de base de datos.....	56
3.1.2 Características y definición de dato.....	57
3.1.3 Características generales de una base de datos.....	57
3.1.4 Las características particulares de una base de datos.....	58
3.1.5 Ventajas de las bases de datos.....	58
3.1.6 Desventajas de las bases de datos.....	59
3.1.7 Componentes de los sistemas de bases de datos.....	59
3.1.8 El sistema de gestión de bases de datos (SGBD o DBMS).....	62
3.1.9 Modelo relacional.....	63
3.1.11 Estructura del modelo relacional.....	64
3.1.12 Claves o Llaves.....	64
3.1.13 Proceso de normalización.....	65
3.1.14 Interrelaciones y cardinalidad.....	66
3.1.15 Relaciones uno a uno.....	66
3.1.16 Relaciones uno a muchos.....	66
3.1.17 Relaciones muchos a muchos.....	67
3.1.18 Álgebra relacional.....	67
3.1.19 Cálculo relacional.....	71

3.2 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE POSTGRESQL 7.2..... 72

3.2.1 ¿Qué es PostgreSQL?.....	72
3.2.2 Versión Open Source (Código Abierto).....	73
3.2.3 Soporte Comercial.....	73
3.2.4 Soporte de la Comunidad.....	74
3.2.5 Características de PostgreSQL.....	74
3.2.6 Catálogo del sistema PostgreSQL.....	77
3.2.7 Ventajas de PostgreSQL.....	79
3.2.8 Desventajas de PostgreSQL.....	80

3.3 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE PHP..... 80

3.3.1 Características de PHP.....	83
3.3.2 Ventajas de PHP.....	84
3.3.3 Desventajas de PHP.....	86
2.3.4 Características de PHP5.....	87

3.4 CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL SERVIDOR DE WEB APACHE VERSIÓN 1.3..... 87

3.4.1 Características de Apache.....	88
3.4.2 Ventajas de Apache.....	89
3.4.3 Desventajas de Apache.....	90
3.5 REDES Y COMUNICACIONES.....	91
3.5.1 Redes.....	91
3.5.2 Redes virtuales.....	94
3.5.3 Comunicaciones.....	96
3.5.4 La familia de protocolos TCP/IP.....	97
3.6 PLATAFORMA LINUX.....	99
3.6.1 Características de Linux.....	99
3.7 SEGURIDAD.....	102
3.7.1 Seguridad en el arranque.....	103
3.7.2 Bloqueo de la consola.....	104
3.7.3 Cuentas de usuario, grupos.....	105
3.7.4 Seguridad de las claves.....	105
3.7.5 Seguridad del root.....	107
3.7.6 El árbol de directorios.....	108
3.7.7 Permisos.....	109
3.7.8 Permisos de archivos y directorios.....	109
3.7.9 Limitar el espacio asignado a los usuarios.....	109
3.7.10 Seguridad en la red, compilando el núcleo.....	110
3.7.11 Comunicaciones seguras.....	111
3.7.12 Firewalls.....	112
3.8 METODOLOGÍAS UTILIZADAS.....	113
3.8.1 Bases de datos relacionales (back end).....	113
3.8.2 RUP - Rational Unified Process (front end).....	113
3.8.2.1 Incremento de la productividad en equipo.....	113
3.8.2.2 Seis mejores prácticas.....	114
3.8.2.3 Desarrollo iterativo de software.....	115
3.8.2.4 Características del desarrollo iterativo.....	115
3.8.2.5 Administración de requerimientos.....	116
3.8.2.6 Arquitectura basada en componentes.....	116
3.8.2.7 Modelo visual del software.....	116
3.8.2.8 Verificación de la calidad del software.....	117
3.8.2.9 Fases en RUP.....	117
4.1 APLICACIÓN DE LA METODOLOGÍA ELEGIDA.....	121
4.1.1 Diagrama de contexto.....	121
4.1.2 Diagrama de flujo de procesos.....	122
4.1.3 Diccionario de datos.....	124
4.1.4 Diagrama Entidad - Relación.....	128
4.1.5 Normalización.....	130
4.1.6 Modelo de requisitos.....	133

TESIS CON
FALLA DE ORIGEN

4.1.6.1 Diagrama de estados.....	134
4.1.6.2 Diagrama de casos de uso.....	136
4.1.6.2.1 Tablas de los casos de uso.....	138
4.1.6.2.2 Diagrama de casos de uso para el usuario.....	141
4.1.6.2.3 Diagrama de casos de uso (operador).....	142
4.1.6.2.4 Diagrama de casos de uso (administrador).....	143
4.1.6.3 Diagrama de actividad.....	144
4.1.6.3.1 Diagrama de actividad (altas, bajas, cambios en registros de catálogos).....	144
4.1.6.3.2 Diagrama de detalle de capacitaciones.....	145
4.1.6.3.3 Diagrama de actividad modificación de password.....	146
4.1.7 Modelo de análisis.....	147
4.1.7.1 Identificación de clases.....	147
4.1.7.1.1 Selección de clases.....	148
4.1.7.1.2 Diagrama estático de clases.....	152
4.1.7.1.3 Diagrama dinámico de clases.....	153
4.1.7.1.3.1 Diagramas de secuencia del Sistema de Administración de Recursos Humanos.....	153
4.1.8 Modelo de diseño.....	157
4.1.8.1 Tarjetas CRC.....	158
4.1.8.1.2 Diagramas de colaboración.....	159
4.2 DISEÑO Y CONSTRUCCIÓN DEL BACK END.....	161
4.2.1 Diseño y construcción de la base de datos (back end).....	161
4.2.2 Seguridad en la base de datos.....	169
4.2.3 Objetos de control y su relación con la base de datos.....	171
4.3. DISEÑO Y CONSTRUCCIÓN DEL FRONT END.....	173
4.4 PRUEBAS E INTEGRACIÓN DEL SISTEMA.....	184
4.4.1 Los procesos del desarrollo como fuente de pruebas.....	185
4.4.2 El plan de pruebas.....	185
4.4.3 Pruebas basadas en autómatas.....	188
4.4.3.1 Modelos de defectos asociados a autómatas.....	189
4.4.4 Modelo de casos de uso como fuente de pruebas.....	189
4.4.5 Modelo de Clases como fuente de pruebas.....	190
4.4.6 Contratos como fuente de pruebas.....	190
4.4.7 Diagramas de colaboración como fuente de pruebas.....	191
4.5 FACTIBILIDAD TÉCNICA Y OPERATIVA.....	199
4.5.1 Recursos de hardware.....	199
4.5.2 Recursos de software.....	202
4.5.3 Requisitos para arrancar el sistema.....	210
4.5.4 Recursos humanos necesarios para la operación del sistema.....	211
4.6 GENERACIÓN DE REPORTES.....	214
ANEXO 1.....	230

TESIS CON
 FALLA DE ORIGEN

MANUAL TÉCNICO	237
Respaldo de la base de datos.....	237
Recuperación del respaldo.....	238
Mantenimiento de la base de datos.....	239
Actualización de las estadísticas del planeador de consultas SQL.....	240
MANUAL DEL USUARIO	241
CONCLUSIONES	251
BIBLIOGRAFÍA	254

TESIS CON
FALLA DE ORIGEN

✓

PAGINACION

DISCONTINUA

INTRODUCCIÓN.

La existencia de nuestra especie durante miles de años ha sido marcada por hitos que la han transformado con el transcurso del tiempo siempre en aras de la evolución. Los historiadores han resuelto denominarlos "revoluciones" porque a partir de cada una de ellas se daba un salto en el proceso del desarrollo humano. Pensemos en la revolución agrícola de la prehistoria por ejemplo, o en la revolución industrial hacia el final del siglo XVIII, hasta llegar a la revolución de la información apenas en el siglo XX. En esta última, el vertiginoso avance de la tecnología de computadoras y comunicaciones ha puesto la mesa para el nacimiento de una extraordinaria forma de comunicación entre millones de personas: *Internet*. Esta red de redes mundial, ha permitido que las distancias se acorten dramáticamente y que se utilice de forma explosiva en mil y un formas distintas. Hoy, gracias a esta red, dos personas ubicadas en puntos geográficamente opuestos del planeta pueden dialogar en tiempo real a un costo relativamente pequeño; es posible cursar estudios de licenciatura o maestrías de manera remota; los servicios de noticias de todos los países están disponibles de forma inmediata y un largo etcétera. En este sentido, un importante número de empresas a nivel mundial se ha percatado de esta estratégica área de oportunidad y ha aprovechado para automatizar sus procesos y ponerlos en disponibilidad en Internet logrando con esto incrementar su productividad, hacer más eficiente su operación y, sobre todo, ahorrar millones de dólares conservando gran parte de su información organizada electrónicamente. A estas empresas hoy se les conoce como *e-business* que significa *negocio electrónico*. Asimismo, las técnicas, metodologías y herramientas de hardware y software en que se desarrollan aplicaciones para este tipo de empresas cambia de forma acelerada tratando de satisfacer las necesidades particulares de desarrolladores y clientes. Y en este orden de ideas se consideran aspectos como la confiabilidad, seguridad, robustez, extensibilidad, facilidad de utilización, etc.

En las siguientes páginas, el lector encontrará un trabajo desarrollado para convertir a la empresa Capital Humano S.A. (ficticia) en un negocio electrónico. Para ello, escogimos las herramientas de software con las mayores ventajas del mercado, las más baratas por tratarse de herramientas pertenecientes a la comunidad GNU (que se explicará posteriormente) y las más robustas. Se utilizaron metodologías consideradas como las mejores prácticas para el desarrollo de software que se tienen en la actualidad.

El presente trabajo incluye la concepción, el análisis, diseño y construcción del **Sistema de Administración de Recursos Humanos** para la empresa Capital Humano S.A.

En el primer capítulo de este proyecto mostramos las características generales de Capital Humano S.A.: Su visión, misión, filosofía, objetivos, atribuciones, políticas, estructura y procedimientos que actualmente lleva a cabo para la selección, reclutamiento y administración de personal.

En el segundo capítulo, se explican conceptos teóricos relacionados con las herramientas de desarrollo utilizadas para la construcción de dicho sistema; también se mencionan características, ventajas y desventajas, así como los aspectos de seguridad involucrados en este aplicativo.

En el tercer capítulo analizamos la problemática con relación a la forma en que se gestionan los recursos humanos actualmente. Tomamos en cuenta los requerimientos de usuario y su correspondiente mapeo a la aplicación; explicamos los medios para obtener información del personal que es aspirante a ocupar una plaza dentro de la empresa; cómo se ordena y cómo se administra cuando el elemento es ya colaborador de la organización. La problemática se estudia por áreas y, en el último capítulo, se plantea una propuesta de solución que, mediante la utilización de herramientas de software, metodologías y técnicas modernas y de comprobada eficacia de análisis y diseño, satisfaga las necesidades de información de la empresa a través de la consolidación de una aplicación vía web robusta, extensible y fácil de utilizar.

Finalmente exponemos las conclusiones de nuestro trabajo e incluimos el manual de usuario y la bibliografía.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO I. POLÍTICAS DE LA EMPRESA

*"Un viaje de mil millas comienza
siempre con un paso".*

— Lao-tse

1.1 Filosofía, misión, visión y objetivos de Capital Humano S.A.

Capital Humano S.A. ofrece una gama de servicios de **Consultoría en Recursos Humanos** orientados a satisfacer las necesidades de empresas cuya necesidad primordial es contar con personal altamente calificado en tecnologías de información.

1.1.1 Filosofía de la empresa.

La filosofía de Capital Humano está centrada en la aplicación tanto de las mejores prácticas como de nuevas tecnologías en las siguientes áreas:

- Desarrollo de recursos humanos: Individual, de equipos de trabajo y organizacional.

El desarrollo de las organizaciones guarda una relación proporcional con el de las personas que la integran. Así como las organizaciones necesitan establecer su rumbo, descubriendo y estableciendo su **Misión, Visión y Valores**, las personas deben enfocar y planificar su desarrollo en función de sus capacidades, intereses y potenciales individuales.

- Desarrollo individual, de equipos y organizacional. Detección de:
 - Competencias individuales.
 - Potencial de las personas.
 - Plan de desarrollo.
 - Plan de carrera.
 - Necesidades de capacitación.
 - Evaluación de desempeño.
- **Coaching & mentoring** (preparación personalizada), individual, de equipos y organizacional.

TESIS CON
FALLA DE ORIGEN

- Tecnología en recursos humanos.
 - **Change management** (Cambios de estilos gerenciales): Diseño estratégico.
 - Persona - Organización – Cultura: Identificación de fortalezas individuales, de equipo y organizacionales.
 - Arquitectura de alineación de las mismas.
 - Detección de estilos: individuales de aprendizaje, de comportamiento y escala de valores.
 - Mapas de estilos grupales y organizacionales.
- Capacitación.
 - Temática estratégica.
 - Desarrollo organizacional.
 - Trabajo en equipo.
 - Conducción por objetivos.
 - Juegos de conducción y evaluación en situación de crisis.
 - Evaluación individual para certificaciones técnicas.
 - Comunicación interna eficaz.
 - Creación y búsqueda de consenso en todos los niveles de la Organización, vía el conocimiento de la misión, visión y objetivos corporativos.
 - Definición de un estilo para lograr identificación, compromiso y generar sentido de pertenencia. Promover el cambio de hábitos.
 - Diseño de mensajes y elementos instrumentales con un concepto integrador y participativo.
 - Diseño y generación de canales de comunicación de tipo estratégico/informativo/formativo acorde a las necesidades/deseos de los distintos tipos de personas que conforman la organización.

- Elementos que agregan valor.
 - Pasos para llevar adelante el cambio.
 - Puntos principales para el éxito en la negociación.
 - Mejores formas para motivar a los empleados.
 - Maneras de ser incondicionalmente constructivos.
 - Puntos a considerar para sostener reuniones exitosas.
 - Puntos a considerar para lograr una exitosa delegación de funciones.
- Tecnologías de medición de desempeño.
 - Sobre la acción: Planeación, continuidad y seguimiento, actitud frente al cambio.
 - Sobre la relación: Con los subordinados, con la organización, capacidades de representación.
 - Sobre los resultados: Calidad, cantidad, conocimiento del trabajo realizado, capacidades de auto desarrollo.
- Análisis del potencial.
 - Amplitud de visión.
 - Sentido de la realidad.
 - Imaginación.
 - Capacidad de análisis.
 - Cualidad de líder efectivo.
 - Adaptabilidad social.

Las organizaciones que hoy promueven y crean las condiciones necesarias para que la gente pueda desarrollar las competencias, aptitudes y actitudes requeridas, cuentan con individuos y equipos de trabajo más efectivos y productivos en su contribución a los resultados del negocio.

TESIS CON
FALLA DE ORIGEN

1.1.2 Misión de Capital Humano S.A.

La misión de Capital Humano es participar en la consolidación de acciones efectivas orientadas a la creación de valor en el activo más importante de las empresas, es decir; su gente.

1.1.3 Visión de Capital Humano S.A.

Ser la empresa líder de consultoría en recursos humanos proporcionando este servicio a empresas del ramo de las tecnologías de información, asegurando que exista el recurso humano capaz de convertir la estrategia en realidad, creando una ventaja competitiva sustentable de la organización.

1.1.4 Objetivos de Capital Humano S.A.

Capital Humano está consciente de que la clave del éxito está en la gente. Sus objetivos son los siguientes:

- o **Aplicar la misión seriamente.** – La misión, los valores y la cultura corporativa están definidas y sirven para alinear los sistemas administrativos, estructuras, procesos, programas y políticas dentro de la empresa. Esto permite claridad entre la cultura y la conducta individual, la cual se define, se sustenta, se reconoce y se recompensa consistentemente en la organización.
- o **Atraer talento.**– Debe existir un claro enlace entre el éxito organizacional y la habilidad para atraer y retener al personal con conocimiento.
- o **Identificar acertadamente al tipo de gente que se está buscando.**– Se busca contratar gente para que haga una carrera, no para hacer sólo un trabajo. Debe dedicarse tiempo y recursos para desarrollar y aplicar modelos de competencias hechos a su medida. La idea es lograr el óptimo acoplamiento entre el perfil de la posición y la persona.
- o **Visualizar el plan de carrera como una inversión.**– Se debe tomar en cuenta que el desarrollo y evolución profesional de la gente es el "torrente

sanguíneo" del futuro de la compañía. Es indispensable recurrir a la rotación de asignaciones para ayudar a la gente a desarrollar su máximo potencial, a fin de que encuentre retos y satisfacción en el trabajo. Es importante asegurarse de que la "inversión por empleado" se encuentre en el extremo alto de los promedios de la industria.

- o **Promover al personal interno.-** Se busca cumplir este objetivo manteniendo un sistema de información actualizado en donde se localice fácilmente al personal con sus respectivos perfiles.
- o **Recompensar el desempeño.-** Como un proceso crítico se realizarán evaluaciones anuales que tendrán como objetivo medir y reconocer el potencial del empleado. Los resultados satisfactorios de un empleado se verán reflejados en el ámbito salarial.
- o **Medir la satisfacción del personal.-** Se debe contar con amplios esquemas de retroalimentación formales e informales, obteniendo reportes estadísticos de desempeño. La satisfacción del empleado se considera una medida de éxito.

1.2 Atribuciones de la empresa.

La empresa Capital Humano S.A., con varios años de experiencia en reclutamiento de personal para otras empresas, cubre una variada gama de actividades, todas enfocadas a la capacitación y desarrollo de profesionistas para el desarrollo de sistemas de información. Para ello se utilizan prácticas que garanticen servicios de calidad logrando así la satisfacción del cliente.

Los servicios que ofrece CAPITAL HUMANO S.A. han sobresalido debido al nivel avanzado de conocimientos tecnológicos, aunado a la experiencia acumulada de múltiples proyectos.

Se realizan procedimientos para encontrar a las personas adecuadas que desempeñarán cargos definidos de su empresa, y disminuir riesgos de error en la predicción de la conducta laboral de los postulantes.

TESIS CON
FALLA DE ORIGEN

1.2.1 Recursos humanos.

Capital Humano S.A., requiere personal idóneo y suficiente, para atender adecuadamente los requerimientos del mercado.

La empresa ha adoptado un procedimiento de entrenamiento permanente para su personal, mismo que conjuntamente con un entorno de trabajo apropiado reditúa su estabilidad y permanencia. Es así como se ha podido mantener un bajo nivel de rotación de personal en un área que se caracteriza por su alta movilidad.

La empresa cuenta con una dotación de recursos humanos integrados por profesionales capacitados para cubrir las necesidades que se requieran.

Por medio de este sistema facilitaremos el control centralizado y la operación descentralizada, lo que permitirá a los líderes del área de recursos humanos concentrarse en las responsabilidades estratégicas de planificación y de programas de desarrollo de personal, mientras las actividades operativas son asumidas y optimizadas por la propia herramienta (sistemas).

1.2.2 Identificación de gente clave.

Esta herramienta permitirá identificar a las personas "clave" para el desarrollo del plan de carrera dentro de la empresa, gestionando en dado caso la movilidad y promoción de los profesionales que trabajan en ella.

1.2.3 Seguimiento de performance (desempeño) de programas de recursos humanos.

Esta tecnología de trabajo permite implementar acciones correctivas que generen adhesión y un compromiso de largo plazo en los colaboradores.

1.2.4 Encuestas de clima organizacional.

Permiten conocer en forma objetiva el grado de satisfacción de la gente en su quehacer cotidiano y valorar como se alinea la estrategia de la empresa con su modelo cultural. Esta metodología está diseñada en función de las necesidades del cliente.

1.2.5 Sistemas de evaluación de desempeño.

Miden el compromiso y la eficiencia de los integrantes de la organización. Proporciona a la empresa un equipo de ventas y captadores profesionales para cursos de capacitación con franquicia tributaria orientados a Pymes¹.

Se caracterizan los cargos de la empresa del cliente, considerando sus actividades o atribuciones, requisitos, responsabilidades y condiciones de trabajo.

La persona podrá preparar una currícula eficaz y enfrentar efectivamente una entrevista de trabajo.

1.3 Direcciones involucradas y Políticas.

1.3.1 Direcciones involucradas.

Para comprender la distribución laboral de la empresa Capital Humano S.A. en la figura 1.3.1.1 se presenta su organigrama.

TESIS CON
FALLA DE ORIGEN

¹ Pequeñas y medianas empresas.
http://asexma.cl/contenido/fomento/consultoria_pymes/388html

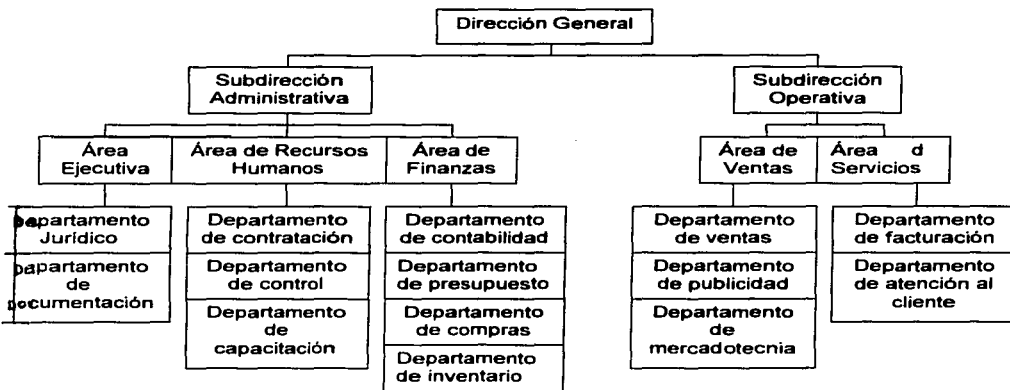


Fig. 1.3.1.1 Organigrama de la empresa Capital Humano S.A.

- **Dirección General.** El director general de la empresa es el encargado de coordinar el funcionamiento de la empresa, evaluar el funcionamiento de las subdirecciones y de tomar las decisiones necesarias para el crecimiento integral de la misma.
- **Subdirección Administrativa.** Maneja la administración de la empresa, el correcto ejercicio de los recursos tanto materiales como técnicos y humanos.
- **Subdirección Operativa.** Lleva el control de la operación del negocio de la empresa y la obtención de los recursos materiales, así como el seguimiento del proceso de negocio.
- **Área Ejecutiva.** Concerta citas con sus posibles clientes y seguimiento a las mismas, así como los asuntos legales.

- Área de Recursos Humanos. Gestiona los Recursos Humanos.
- Área de Finanzas. Administra los recursos materiales (económicos).
- Área de Ventas. Verifica los procesos de ventas de servicios de recursos humanos (giro de la empresa).
- Área de Servicios. Regula la relación cliente-servicio.
- Departamento Jurídico. Se encarga de los asuntos legales.
- Departamento de Documentación. Gestiona la documentación de la empresa.
- Departamento de Control de Recursos Humanos. En este departamento se lleva un control del trabajo de los empleados de la empresa, así como las necesidades de nuevo personal.
- Departamento de Contratación de Recursos Humanos. Se encarga de la evaluación de candidatos y de su contratación.
- Departamento de Capacitación de Recursos Humanos. Se diseñan perfiles para el desarrollo de plan de carrera.
- Departamento de Contabilidad. Controla los ingresos y egresos de la empresa, sin tener injerencia en la distribución interna.
- Departamento de Presupuesto. Administra el ejercicio de los recursos materiales de la empresa.
- Departamento de Compras. Evalúa las necesidades de la empresa y adquiere los insumos necesarios para la operación completa de la empresa.
- Departamento de Inventario. Contabiliza los bienes inmuebles.
- Departamento de Ventas. Concentra la fuerza de ventas de la empresa. Genera contratos con los clientes.
- Departamento de Publicidad. Promociona a la empresa de acuerdo a las necesidades del mercado.
- Departamento de Mercadotecnia. Estudia el mercado e informa sobre sus tendencias.
- Departamento de Facturación. Expide las facturas y cobros a los clientes.
- Departamento de Atención a Clientes. Vigila la entera satisfacción del cliente.

TESIS CON
FALLA DE ORIGEN

1.3.2 Políticas.

Las políticas de la empresa son un conjunto de normas que funcionan como un reglamento con el fin de establecer un comportamiento uniforme dentro de la misma empresa, así como definir criterios de contratación de personal.

1.3.2.1 Políticas para los empleados.

- Los empleados observarán durante el tiempo que estén dentro de las instalaciones del cliente las normas bajo las cuales se rigen.
- Los empleados respetarán durante el proyecto el horario establecido de trabajo.
- Si el proyecto requiere, el empleado deberá laborar en un horario extraordinario.
- Es responsabilidad del empleado generar un reporte de actividades mensual firmado por su jefe inmediato y por el encargado del proyecto.
- En caso de que el empleado sea requerido para realizar actividades fuera del lugar de residencia, se le asignará un presupuesto para viáticos que no excederá la suma de sus gastos de transportación, hospedaje y alimentación.
- El empleado tiene derecho a recibir capacitación dentro de las instalaciones del cliente de acuerdo a los lineamientos establecidos por el departamento de capacitación.

1.3.2.2 Políticas para los empleadores.

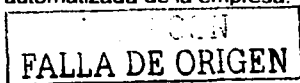
- Para llevar a cabo la selección de candidatos a un puesto dentro de la empresa, el empleador deberá publicar la vacante al departamento de recursos humanos con el fin de ubicar en primera instancia al personal que cumpla con el perfil de la vacante y que se encuentre laborando en la empresa.

- El primer paso para contratar a un candidato será aplicar una entrevista en la que se identifiquen los puntos básicos manejados por el departamento de contratación de recursos humanos.
- Para contratar a un candidato será necesario aplicar un examen de conocimientos técnicos y un psicométrico. Un candidato que no alcance la calificación mínima establecida por el departamento de recursos humanos no podrá ser considerado para contratación.
- Los candidatos a contratar no podrán ser menores de 18 años ni mayores de 40 años.
- Los candidatos a contratar deberán contar con el 100% de créditos a nivel licenciatura y promedio mínimo de 8.
- Los candidatos que no presenten cédula profesional deberán ser considerados como pasantes y no aptos para puestos de líder de proyecto y superiores.
- Es requisito indispensable para los candidatos varones que presenten la cartilla del Servicio Militar Nacional.
- Es requisito indispensable para todos los candidatos que presenten pasaporte y visa vigentes.
- Es requisito indispensable para todos los candidatos que posean un 80% de dominio del idioma inglés a nivel técnico.

La subcontratación de personal con empresas externas se llevará a cabo solamente para proyectos cuya duración sea menor de 3 meses y no exista en la empresa el personal que cubra el perfil requerido.

1.4 Procedimientos actuales del Departamento de Recursos Humanos.

Actualmente la operación del Área de Recursos Humanos es la menos automatizada de la empresa.



El Departamento de Recursos Humanos cuenta con procedimientos, que en su mayoría manuales, gestionan la información de la planta laboral. En las empresas éstos no son viables ya que se concentran en archivos de volúmenes considerables, generando una mal funcionamiento en este departamento.

La empresa recibe un promedio mensual de 1500 notas relacionadas con tramites administrativos, lo que hace que la operación en recursos humanos se vea obstaculizada, ya que el procedimiento utilizado para clasificar este material es como se mencionaba, manual.

Por otro lado el departamento de Recursos Humanos responde con gran lentitud a las peticiones de la empresa sobre recursos con perfiles determinados.

Respecto al proceso de contratación se realizan los siguientes pasos:

Publicación de la oferta de trabajo, recepción de currícula, en el recibidor de la empresa, la currícula es llevada al departamento de recursos humanos donde una persona se encarga de revisarlo y si el candidato tiene el perfil requerido entonces se procede a concertar una cita para hacerle una entrevista; si el candidato no tiene el perfil para el puesto solicitado pero tiene un perfil que puede ser útil en un futuro se procede a archivarlo. Propiciando con esto, que se vayan acumulando en los expedientes, cientos o miles de archivos de documentos que, con el paso del tiempo, provoca que se pierda la noción de los mismos. Cuando éstos se han almacenado por un tiempo mayor a un año se pierde la versatilidad y la seguridad de que lo contenido en el currículo vitae sea información reciente. De esta forma el proceso se lleva a cabo manualmente traduciendo la decisión para una contratación de una persona en algo demasiado lento.

Una situación semejante ocurre en las evaluaciones y recomendaciones de ascensos y/o entrega de incentivos al personal que labora en la empresa. Al tener los datos de las personas que laboran en grandes archivos, se ocasionan grandes retrasos de tiempo. De tal forma que una persona que se ha ganado un ascenso no le es otorgado en el tiempo razonable para tal efecto, generando una baja de productividad en el personal que labora en la empresa.

Actualmente si se necesita conocer información de una persona que labora en la empresa, se tiene que buscar a los archivos generando un retraso en el resultado. Otro inconveniente se presenta al tener nuestra información concentrada en un lugar porque esta expuesta a siniestros naturales como incendios, terremotos etc.

El exceso de papeleo en los procedimientos como envío, recepción y almacenamiento de solicitudes de empleo, cartas de recomendación, etc., provoca que sea imposible su organización a largo plazo.

Los informes y reportes tardan mucho tiempo en desarrollarse porque la información se encuentra dispersa por el exceso de papeleo.

Al tener la información archivada no se puede obtener acceso a ella de manera remota, de tal forma que cuando un alto ejecutivo necesita información para poder tomar una decisión, ésta suele tardar demasiado tiempo ya que se tiene que buscar con anticipación.

Anualmente el departamento de recursos humanos debe entregar un informe detallado con respecto a sus actividades correspondientes tales como logros,

TESIS CON
FALLA DE ORIGEN

objetivos, metas etc., con estadísticas detalladas de las actividades realizadas en el periodo. La información necesaria para elaborar las estadística debe ser buscada en los archivos.

También es de fundamental importancia la consolidación de información respecto al personal administrativo de las empresa. Con el objetivo de facilitar un análisis y el diagnóstico de varios aspectos que caracterizan el quehacer de la empresa proporcionando elementos necesarios para una buena planificación.

El valor de una empresa lo determina su personal, por eso es importante elaborar un inventario de sus recursos humanos. Este inventario es muy útil para la administración del personal, su planeación y estrategia. El inventario de recursos humanos abarca una serie de registros históricos de la suma de aptitudes, experiencias y cualidades de los empleados que permite generar un análisis y una evaluación del nivel competitivo en un momento dado, cabe aclarar que este inventario es desarrollado por el personal de recursos humanos manualmente donde una persona revisa documento por documento para obtener el informe.

Actualmente se desarrolla un registro del análisis de las tareas, en este paso se debe desarrollar una lista con cuatro columnas. La primera debe tener las tareas incluyendo lo que se va a realizar, en la segunda columna van las normas de desempeño como cantidad, calidad, etc.; en la tercera columna se describen las aptitudes específicas y en la última columna se describen las aptitudes requeridas. El personal de recursos humanos desarrolla un hoja de instrucciones donde prepara un programa de capacitación para el puesto. Podemos decir que para llevar a cabo una excelente capacitación se debe hacer un análisis minucioso del personal de la empresa de acuerdo a las habilidades, conocimientos, actitudes y aptitudes que el personal requiere para desempeñar su puesto. Los principales medios para formar profesionalmente a los recursos humanos son cursos,

eventos, conferencias, lecturas, instrucciones programadas, software educativo, becas y otras formas de educación a distancia.

La capacitación se debe dar a medida que cambian las exigencias del puesto, se tienen que modificar y actualizar las habilidades. El departamento de recursos humanos realiza un análisis sobre niveles salariales, prestaciones, prácticas de los supervisores, asistencia en la planeación de la carrera profesional y la retroalimentación que los empleados reciben de su desempeño. También el departamento de recursos humanos compara la división, sección o departamento con otra entidad análoga, a fin de identificar áreas de desempeño insuficiente. Este enfoque se emplea para la evaluación del ausentismo, la tasa de rotación y los niveles salariales. También cuando se está poniendo a prueba un nuevo sistema, el departamento de recursos humanos examina los documentos sobre contratación, compensación, disciplina y evaluación del desempeño. El objetivo es garantizar la existencia de iguales normas y procedimientos para toda la organización, así como el cumplimiento de las disposiciones legales.

El departamento de recursos humanos realiza una gran cantidad de actividades técnicas para determinar la forma en que se llevan a cabo las actividades del personal. Para que esta información resulte útil, se compila en un informe global. El informe de auditoría constituye una descripción general de las actividades de personal e incluye tanto recomendaciones como el reconocimiento formal de las prácticas que están logrando su objetivo.

Dicho informe contiene varias partes: una se dirige a los gerentes de línea, otra a los gerentes que tienen a su cargo funciones específicas del departamento de personal y la parte final se dirige al gerente general del departamento de personal. En el caso de los gerentes de línea, el informe resume los objetivos de personal que se han postulado, así como las responsabilidades y los deberes de cada gerente. Entre las responsabilidades y deberes de los gerentes de línea se cuentan la responsabilidad de entrevistar a los solicitantes, capacitar a sus

empleados, evaluar el desempeño, motivar a su equipo de trabajo y satisfacer sus necesidades personales.

El informe identifica también los problemas de personal. Se detectan las desviaciones respecto a las políticas vigentes y/o disposiciones legales. Las prácticas insuficientes o erróneas se señalan en el informe, junto con recomendaciones para solucionar los problemas.

El informe puede contener también información sobre las actitudes prevalecientes entre los gerentes de línea respecto a los programas de personal, en ocasiones con información enriquecida mediante estadísticas externas.

El informe dirigido al gerente de personal contiene toda la información dada a los gerentes de línea y a los gerentes de departamento de personal. Además de las actitudes entre gerentes de línea y empleados respecto al desempeño del departamento de personal. Los objetivos del departamento y sus planes para alcanzarlos. Los problemas de recursos humanos y sus implicaciones. Recomendaciones respecto a los cambios necesarios y la prioridad de cada uno.

Mediante la información que contiene el informe de auditoría, el gerente de personal puede lograr una perspectiva amplia e integral de la condición de las funciones de personal de su empresa.

La evaluación permite establecer una estrategia a futuro y sirve como punto de referencia para futuras evaluaciones. En términos generales, el objetivo del sistema será servir de apoyo a la gestión de los recursos humanos, tanto a nivel gerencial como operativo, brindando también información que podrá utilizarse en procesos de planificación. El sistema estará organizado en módulos cooperantes, cuyos objetivos específicos pueden enunciarse como:

- Facilitar el registro, la actualización y la consulta del legajo del personal.
- Generar información con una perspectiva global, incluyendo resúmenes de cargos por sectores, por fuente de financiamiento, por tipo de puestos, entre otros.
- Facilitar y apoyar la realización de los controles administrativos tendientes a fortalecer la gestión, emitir las planillas, recibos, cheques de liquidaciones y otras planillas y documentos, de acuerdo al registro de asistencia del personal y a las normas administrativas aplicables.
- Apoyar los procesos de auto-evaluación institucional a través de la emisión de reportes estadísticos y otros informes

El sistema facilita la gestión de todos los datos relacionados a la disponibilidad de los recursos para la contratación del personal, definición de categorías, puestos, áreas de trabajos, horarios de trabajo, asistencias del personal, contratación y movimientos del personal, tipos de liquidaciones de sueldos, datos necesarios para realizar los controles y verificaciones de todos los datos relacionados al personal que servirán para la liquidación de salarios del personal y posterior emisión de las planillas, recibos, cheques de sueldos.

**TESIS CON
FALLA DE ORIGEN**

CAPÍTULO II. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN

*"Rien ne va de soi. Rien n'est donné.
Tout est construit (Nada es evidente.
Nada está dado. Todo es construido)".*

— Gastón Bachelard

2.1 Problemática actual.

Para poder crear un sistema de administración de recursos humanos nos enfrentamos con la necesidad de comprender los procedimientos que el departamento de recursos humanos de la empresa lleva a cabo tanto para el reclutamiento de candidatos como para el seguimiento de la plantilla actual de trabajadores. Para ello es necesario establecer sesiones de preguntas y respuestas con la gente que actualmente lleva a cabo dichos procesos para que se generen los esquemas de procedimientos que servirán para el análisis del sistema. En general, la gente a la cual es necesario entrevistar no está disponible, lo cual retrasa la recopilación de información necesaria para el sistema.

El control de los recursos humanos en la empresa comienza con la necesidad de asignar personal a una determinada labor. Una vez que se hubo definido la actividad que el futuro empleado desempeñará, se definen los criterios de selección, como conocimientos y habilidades necesarias, tiempo de experiencia previa y nivel de escolaridad requerido entre otros. Esto permite establecer un perfil con el cual se podrá ubicar al recurso humano que mejor desempeñará las actividades que se requieren.

El siguiente paso consiste en la ubicación de las personas para lo cual, en primera instancia, se realiza una búsqueda dentro del personal que labora actualmente en la empresa y en caso de que se cumplan el perfil y los demás requisitos además de que se encuentre en posibilidad de ser asignados a nuevas actividades, el proceso de búsqueda es finalizado. Generalmente, esta actividad se realiza preguntando a los superiores tanto el nivel como la disponibilidad del personal que se encuentra o encontró bajo su supervisión. En caso de que no se ubique, se hace una revisión a la documentación (currícula) y localización del empleado para la verificación de su disponibilidad (inclusive, existe la posibilidad de que ya no labore en la empresa).

TESIS CON
FALLA DE ORIGEN

Si dentro de la empresa no es posible ubicar o disponer del personal requerido para las nuevas actividades, es necesario realizar un proceso de reclutamiento externo publicando la vacante en periódicos, internet o algún otro medio de comunicación masiva. Con esta acción, lo más probable es que se presenten varios candidatos a los que hay que evaluar tanto técnica como psicológicamente para ubicar al personal que mejor se aproxime al perfil que se requiere.

Para determinar las habilidades técnicas del candidato se realizan exámenes y entrevistas con el personal calificado, el cual emitirá una evaluación para que la gente encargada del reclutamiento proceda a contratar o rechace a cada candidato.

La primera referencia que se tiene de un candidato es a través de un currículum vitae o algún tipo de presentación que muestre la experiencia previa. El candidato generalmente envía su presentación y espera hasta que se le asigne una entrevista.

En general, un currículum vitae es un inventario de calificación que contiene información sobre elementos como el historial de desempeño del empleado. Se pueden describir como requisitos sistemáticos, ya sean manuales o computarizados, que enlistan la educación, carrera o interés de desarrollo, idiomas, habilidades especiales, etc. de los trabajadores, para utilizarse en la proyección de los candidatos internos para promoción.

Una vez que se hubo llevado a cabo la selección de los candidatos que mejor se aproximan al perfil requerido, es necesario presentarlos con el cliente o usuario que solicitó al personal, para que se lleve a cabo una última evaluación, con la que se definirá quién será la persona que obtendrá el puesto para el cual fue solicitado.

TESIS CON
FALLA DE ORIGEN

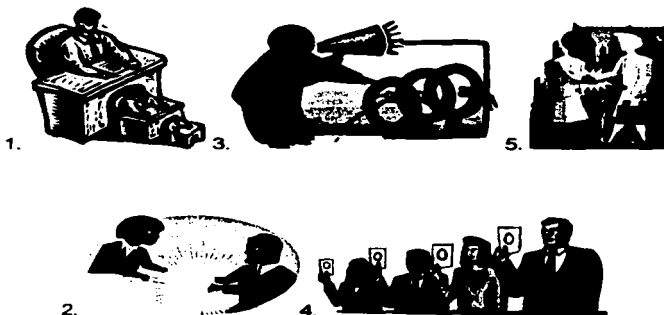
2.1.1 Proceso de reclutamiento y selección y su problemática asociada.

Figura 2.1.1.1 Descripción del proceso de reclutamiento

En la figura 2.1.1.1 se observa el proceso de reclutamiento de personal. 1. Definición del perfil y solicitud a recursos humanos. 2. Consulta manual de la currícula existente. 3. Publicación de la oferta de trabajo. 4. Evaluación de candidatos de acuerdo al perfil. 5. Presentación al cliente de candidatos calificados. A continuación se especifican los problemas que existen en cada etapa de dicho reclutamiento:

- Definición del perfil y solicitud a recursos humanos. Para poder llegar a la definición del perfil es necesario contar con el apoyo del personal con la mayor experiencia que se encuentre involucrada en el proyecto. Si no es posible contar con personal suficientemente capacitado para este fin, es necesario introducir a alguien en el proyecto para que pueda identificar y

jerarquizar las necesidades. En muchas ocasiones es difícil definir claramente un objetivo, lo que dificulta en muchas formas la definición de un perfil. Además de tener en cuenta el aspecto técnico, es muy importante saber definir cómo el recurso que cumpla con el perfil deberá controlar la presión, manejar a otros recursos y/o canalizar recursos económicos o materiales de la empresa.

- Consulta manual de la currícula existente. Una búsqueda manual puede excluir por cansancio o descuido al candidato idóneo, además de que es posible tener más de una copia de algún currículum con variaciones en la información debido a actualizaciones, lo que puede traducirse en información errónea o incompleta. Otro problema reside en que es difícil acceder físicamente al archivo y personal con poco conocimiento del perfil puede omitir a algún candidato debido a ignorancia en un tema específico, además de que el proceso se ralentiza pues se realiza una mayor cantidad de pasos dentro del proceso. El hecho de que un archivo se encuentre exclusivamente en forma física en un lugar evita que el currículum del recurso se encuentre publicado en otras localidades, lo que impide que el recurso se pueda ubicar en otras localidades, o que un gerente que se encuentre en otra localidad lo pueda seleccionar para trabajar de forma remota.
- Publicación de la oferta de trabajo. Para una persona con pocos conocimientos acerca del perfil, es complejo detallar la necesidad y características solicitadas, lo que puede producir una sobre demanda de personal que no cumpla satisfactoriamente. Esto retrasa el proceso pues es necesario evaluar a una mayor cantidad de candidatos. Generalmente los candidatos que se presentan a solicitar el puesto no cumplen cabalmente con el perfil, pero eso no lo hacen de conocimiento del entrevistador con el fin de no perder la oportunidad, lo que provoca que los exámenes de selección tengan que ser exhaustivos para definir el perfil, y esto se traduce en mucho tiempo por cada candidato.

- Evaluación de candidatos de acuerdo al perfil. Debido a la ambigüedad en los términos técnicos y nombres de las habilidades, puede haber discrepancias entre el perfil y la currícula de los candidatos, lo que dificulta el trabajo al departamento de recursos humanos para la primera selección. Es por ello que es necesario aplicar algún tipo de mecanismo de evaluación técnica con personas calificadas. Generalmente este proceso se lleva a cabo en una segunda entrevista, lo que retrasa la selección dependiendo del número de candidatos y de la cercanía entre la gente de recursos humanos y la del área técnica.
- Presentación al cliente de candidatos calificados. Una vez que se eligieron los candidatos que mejor se apegan al perfil solicitado, se concerta una entrevista con el cliente para que lleve a cabo una última evaluación técnica y tome la decisión definitiva. El cliente puede aceptar o no al candidato, dependiendo de su apego al perfil, de su experiencia y de su conocimiento de lo que hará dentro del proyecto. En general el cliente se atiene a aquello que la empresa le brinde, pero está en todo el derecho de pedir un cambio de recurso en cualquier momento.

2.2 Requerimientos del Departamento de Recursos Humanos.

Capital Humano S.A. desea contar con una herramienta amigable y fácil de usar que, además, sea accesible desde Internet para poder resolver la problemática detectada y, de esta manera, hacer más eficiente la operación del área de recursos humanos, perteneciente a la subdirección administrativa.

TESIS CON
FALLA DE ORIGEN

2.2.1 Requerimientos generales.

La empresa Capital Humano SA, a través de las subdirecciones administrativa y operativa, respectivamente, requiere que la nueva herramienta les permita:

- Hacer más sencilla la forma en que se gestiona la información de la planta laboral de la empresa,
- que la operación de los departamentos beneficiados con esta herramienta, en general, sea más eficiente y productiva,
- que los tiempos de respuesta exigidos por la dirección general y subdirecciones para la toma de decisiones sean más cortos,
- contar con un fácil y ágil acceso desde internet,
- mantener la integridad de la información mediante mecanismos de seguridad y mediante un estricto control de niveles de acceso,
- navegar a través de la aplicación de manera fluida y sin complicaciones,
- establecer un mecanismo de mensajes entre usuarios,
- mantener toda la información respaldada en una base de datos.

2.2.2 Requerimientos particulares.

Para poder comprender mejor las necesidades de Capital Humano SA, se listan a continuación los requerimientos particulares que se desprenden de los requerimientos descritos en el punto 2.2.1 y se especifica el departamento, área y/o subdirección a la que pertenecen:

Los departamentos que pertenecen al área de recursos humanos (departamento de contratación, de control y capacitación) tienen los siguientes requerimientos particulares:

- **Hacer más sencilla la forma en que se gestiona la información de la planta laboral de la empresa.**
 - La aplicación debe ser capaz de dar de alta y de baja, de manera intuitiva a través de un **browser** (intérprete de HTML): La

información que se obtenga tanto de los candidatos como de la actual plantilla laboral, la información de cursos de capacitación ofrecidos por la empresa y/o por empresas proveedoras externas, la información de proyectos existentes, boletines informativos, las preguntas y respuestas de entrevistas, etc.

- La información del candidato podrá ser alimentada por el candidato mismo, para que cuando llegue el momento de la entrevista con el personal de RH, ya se cuente con un informe-análisis preliminar de los datos generales del candidato.
- La información de cualquier empleado podrá ser modificada por el empleado mismo para que ésta siempre esté actualizada.
- Deber haber un módulo que permita dar mantenimiento a los catálogos y a las reglas de negocio.
- Los módulos de la aplicación deben estar perfectamente separados por las reglas de seguridad interna y deben asignarse niveles de acceso entre los usuarios.
- Las operaciones que realice el sistema deben ser precisas y los tiempos de respuesta aceptables, tomando en cuenta que muchos usuarios se conectarán utilizando un módem.
- El sistema debe ser paramétrico y flexible, debe permitir crecimiento y adaptarse a los cambios que sufra la empresa y/o el departamento de RH.
- Debe estar montado en una plataforma estable y debe contar con el soporte técnico adecuado durante su desempeño.

Los siguientes requerimientos de la subdirección administrativa, también se desprenden de necesidades de los departamentos que pertenecen al área de Recursos humanos y a los departamentos del área de finanzas (departamentos de compras y presupuesto), además, incluyen a los departamentos de ventas y facturación adscritos al área de ventas y servicios respectivamente, quienes reportan directamente a la subdirección operativa:

TESIS CON
FALLA DE ORIGEN

- **Que la operación de los departamentos beneficiados con esta herramienta, en general, sea más eficiente y productiva.**
 - El acceso a la aplicación debe ser controlado por un nombre de usuario y contraseña.
 - El nombre del usuario y su contraseña deben ser únicos. No puede haber dos usuarios del sistema compartiendo cualquiera de estos dos valores. Siempre que se genere un nombre de usuario o contraseña nuevos, deberá validarse que no existan en la base de datos. De existir tales valores, se solicitará al usuario que defina un nombre de usuario o contraseña diferentes.
 - Los mensajes de error al no poder ingresar al sistema no deben especificar en qué se equivocó el usuario, deben ser genéricos. Ej. "Datos erróneos, verifique su nombre de usuario y/o contraseña e intente nuevamente".
 - La contraseña debe podrá modificarse únicamente por el usuario dueño de la misma. En caso de olvido o pérdida, se deberá notificar al administrador para que éste asigne una nueva contraseña temporal, que deberá ser modificada por el usuario en cuestión a la brevedad posible.
 - La aplicación debe mostrar en su página principal, dos opciones: La primera, para que los usuarios existentes puedan ingresar a su cuenta, y los usuarios que no estén registrados en el sistema, puedan hacerlo en línea utilizando la sección "Registro de usuarios nuevos".
 - Una vez que el usuario ha sido registrado podrá ingresar al sistema.
 - La posición global de la aplicación debe mostrar el contenido de la misma de acuerdo al nivel del usuario y, además, debe ser capaz de reconocer al usuario que se ha conectado.

- Los menús, ligas, opciones y operaciones permitidas al usuario deben ser fáciles de acceder, concretas y no deben permitir que el usuario invierta más de 10 minutos llenando formularios.
- En la posición global de cada usuario, se deberán visualizar boletines informativos que cambiarán dinámicamente según considere el usuario responsable de los mismos. Dichos boletines contendrán asuntos relativos a la plantilla laboral en general y funcionarán como un canal de comunicación entre la alta dirección y subdirecciones de la empresa y todos los colaboradores. El usuario **operador** será el responsable del mantenimiento de los boletines informativos. El será el único usuario que podrá dar de alta, modificar o eliminar boletines. Se deberá contar con un mecanismo de respaldo de boletines para que no sean completamente eliminados. Debe existir la posibilidad de reciclar boletines cuando se trate de eventos o mensajes periódicos.
- El contenido de la aplicación no debe estar saturada de imágenes ni de elementos que hagan lento el tráfico de datos entre cliente y servidor.
- Se debe incluir un módulo de búsqueda de recursos, dentro del cual habrá una búsqueda simple y una avanzada.
 - La búsqueda simple debe contener dos filtros: Habilidad y disponibilidad; de manera que el resultado de la búsqueda sea rápido y permita encontrar recursos que poseen la habilidad requerida, así como su disponibilidad para asignarse a un determinado proyecto.
 - La búsqueda avanzada debe contener un número de filtros mayor para permitir acotar el resultado y hacerlo más preciso. Dentro de estos filtros deben figurar además de los filtros de la búsqueda simple: Edad, sexo, nacionalidad, documentación disponible, capacitación, estatus, perfil, rol, salario, tarifa, ubicación, etc.

- o Es deseable que algunas operaciones que sean realizadas por usuarios y administrador (alta, baja o cambios de información), surtan efecto hasta que sean aprobadas por un usuario con mayor autoridad (Operador).
 - En este rubro, el **operador** verá en el contenido de su posición global, una lista de operaciones que aparecerán como pendientes y, al escoger una en particular, podrá visualizar el detalle de cada caso para, de esta forma, tomar una decisión. Las opciones de decisión serán: Autorización, Rechazo o emisión de una Recomendación con comentarios.
 - El **usuario** que originalmente solicitó algún tipo de operación, como pueden ser modificaciones en sus datos privados, por ejemplo, recibirá la notificación de la decisión del **operador** en su propia posición global y la información habrá sido afectada siempre y cuando la decisión haya sido aprobada por el **operador**. Si este último toma la decisión de rechazar el cambio, simplemente no se afectarán los datos.
 - El **administrador** que, de la misma manera, intente hacer alguna modificación a catálogos o tablas que contengan reglas de negocio, se verá en la misma situación que el punto anterior, es decir, el **operador** será quién en última instancia decida si los datos se afectan o no en la base de datos. Un ejemplo para este caso puede ser la baja de un curso de capacitación en el catálogo de cursos de capacitación que ofrece la empresa. El **administrador** puede considerar que por haber concluido dicho curso, es necesario eliminarlo. Cuando intenta eliminarlo mediante la aplicación, ésta responderá que la operación requiere autorización del **operador**, pero que la petición ha sido enviada. Una vez que el **operador** ingrese al sistema, verá en su posición global, una lista de operaciones por autorizar. Asumiendo que toma

en primer lugar el caso de eliminación del curso del catálogo correspondiente y suponiendo que está de acuerdo, seleccionará la opción de autorizar, lo que implica que hasta ese momento se hará efectiva la operación de eliminación del registro en la base de datos.

La Dirección General, quien es la encargada de aprobar la adquisición de los servicios para la construcción de esta aplicación, reunió los siguientes requerimientos particulares:

- **Que los tiempos de respuesta exigidos por la dirección general y subdirecciones para la toma de decisiones sean más cortos.**
 - El acceso de usuarios pertenecientes a las subdirecciones y/o a la Dirección General, deben tener un acceso absoluto al sistema, aunque de sólo lectura poniendo especial énfasis en las secciones del sistema que impliquen inteligencia de negocio (reportes y búsquedas).
 - La aplicación debe contar con un módulo de reportes ejecutivos.
 - El módulo de reportes debe ser flexible y preciso. Debe incluir los filtros necesarios que permitan obtener un mayor grado de precisión en el tipo de información que se busca reportar.
 - El módulo no debe permitir ambigüedades y debe desplegar resultados en un tiempo razonable.

Todos los departamentos a las órdenes de las dos subdirecciones y, a su vez la dirección general de la empresa, coincidieron al definir los siguientes requerimientos debido a que la necesidad de conectarse al sistema dentro o fuera de la empresa es algo que todos comparten, además, permitirá que las sucursales geográficamente distribuidas fuera del corporativo puedan acceder al sistema centralizado:

TESIS CON
FALLA DE ORIGEN

- **Contar con un fácil y ágil acceso desde Internet.**
 - La aplicación debe estar disponible en Internet en una dirección fácil de recordar.
 - El acceso debe ser ágil, las páginas no deben tardar demasiado en cargar.
 - Se debe permitir el acceso al sistema a toda aquella persona que pertenezca a la empresa y a aquellas que sean candidatas toda vez que se registren en línea para poder acceder al sistema.

El área ejecutiva, a través del departamento jurídico, ha manifestado la necesidad de mantener la privacidad de la información. El sistema, a pesar de no estar considerado como una fuente de información de alto riesgo para la empresa, ciertamente debe respetar algunos criterios de seguridad para evitar posibles ataques cibernéticos, accesos no autorizados, duplicidad de funciones y/o manejos irregulares de información de empleados en general. Partiendo de las premisas anteriores, éstos son los requerimientos particulares:

- **Mantener la integridad de la información mediante mecanismos de seguridad y mediante un estricto control de niveles de acceso.**
 - Es imperativo que el ingreso a la aplicación esté controlado por un nombre de usuario y contraseña.
 - Si el usuario fracasa en su intento por ingresar a la aplicación más de tres veces, es decir, proporciona una combinación de nombre de usuario y/o contraseña inválidos, su cuenta se deberá desactivar, debiendo notificar esta situación al administrador para que dicha cuenta pueda ser reactivada. El valor de intentos fallidos debe ser paramétrico.
 - La página debe contar con un certificado digital instalado, es decir, el sitio debe ser seguro y debe implementar HTTPS (protocolo de transferencia de hipertexto seguro) mediante la encriptación de la información a 128 bits (SSL 128bits). La llave deberá pertenecer a

una empresa reconocida en el ramo de la expedición de certificados digitales como **Verisign**² por ejemplo.

- Se deben establecer niveles de acceso: **Administrador, operador y usuario**.
 - Los niveles son jerárquicos comenzando con el **Administrador** como el usuario con autoridad de acceso total a catálogos, reglas de negocio, creación de usuarios, reactivación de cuentas, etc. Continuando con el **operador** quien tendrá una autoridad intermedia y acceso de sólo lectura a información de usuarios, catálogos, reglas de negocio, boletines, etc. El usuario catalogado como **operador** es quien tomará las decisiones de negocio durante la operación del sistema. Cualquier operación cuyo resultado sea una escritura o una eliminación de información de la base de datos, tendrá que pasar por el **operador**, quien tendrá la capacidad de tomar decisiones al respecto. Finalmente los **usuarios** tendrán la autoridad más baja del sistema y solo tendrán acceso a información propia, pudiendo leer sus datos pero, como se mencionó anteriormente, cualquier intento de escritura para modificar o eliminar datos propios tendrá que ser avalada por el **operador**.
- Debe existir un parámetro de inactividad que, cuando se cumpla, invalide la sesión, enviando al usuario a la pantalla de *Login*.

El departamento de compras, adscrito al área de finanzas de la subdirección administrativa, quien originalmente fue la interesada en la adquisición de este sistema, ha comunicado como requerimiento principal que la aplicación debe ser una herramienta útil que facilite el trabajo de quien la utilice. Aquí se listan las particularidades de dicho requerimiento:

² <http://www.verisign.com/>

- **Navegar a través de la aplicación de manera fluida y sin complicaciones.**
 - El inicio de la aplicación debe mostrar una pantalla de login y/o registro de usuarios nuevos.
 - Una vez que el usuario ha ingresado a la aplicación, se le debe dar la bienvenida y tendrá un menú en el extremo izquierdo, en el derecho, se deberán visualizar los boletines activos. Las opciones del menú estarán en función de su nivel de acceso al sistema. Cada opción del menú, llevará al usuario a una página que puede contener formularios editables, no editables o simplemente texto. Cada pantalla accedida tendrá un icono o una liga a la posición global (Home) o un botón de regresar que lo debe conducir a la pantalla inmediata anterior.
 - Cuando el usuario desee salir del sistema lo deberá hacer presionando la liga **Salir** del sistema con el objeto de que su sesión termine correctamente.

La dirección general, en conjunto con las subdirecciones administrativa y operativa, desean poder estar en contacto con todo el personal a través del mecanismo que a continuación se detalla en el requerimiento:

- **Establecer un mecanismo de mensajes entre usuarios.**
 - El sistema debe contener un mecanismo personalizado de envío de mensajes entre usuarios. Esto permitirá el envío de avisos urgentes dirigidos a una o más personas que existen dentro del sistema.
 - Las operaciones de escritura que sean solicitadas por los usuarios y/o el administrador podrán llevar un mensaje dirigido hacia el operador. Asimismo, el operador podrá tomar su decisión y tendrá la oportunidad de añadir un comentario al mensaje que tenga relación con la decisión que ha tomado.

El área de recursos humanos, a través de su departamento de control está preocupada por una eventualidad que pueda propiciar pérdidas de información. Asimismo, desean mantener resguardado el histórico de toda la información que se vaya acumulando durante el tiempo de operación del sistema:

- o **Mantener toda la información respaldada en una base de datos.**
 - o La información debe residir en una base de datos que permita conservar la información íntegra y libre de inconsistencias.
 - o Debe considerarse que las operaciones en la base de datos sean optimizadas y deberán ser minuciosamente revisadas y validadas por un administrador de bases de datos.
 - o Se deben establecer mecanismos de respaldos periódicos de la base de datos, así como mecanismos de depuración que impidan una saturación de la misma.
 - o Los respaldos deben ser guardados en cualquier medio magnético, siempre y cuando resida en equipos distintos al que aloja la base de datos principal.

2.3 Búsqueda y análisis de la información.

Actualmente el área de Recursos Humanos tiene varios procedimientos para gestionar la información de la planta de su personal éstos, en su mayoría son manuales, de tal forma que no son viables en una organización relativamente grande. Es de fundamental importancia la consolidación de información respecto al personal administrativo de las empresas con el objetivo de facilitar el análisis y el diagnóstico de varios aspectos que caracterizan las funciones de éstas proporcionando elementos necesarios para una buena planificación.

Por lo anterior, la empresa Capital Humano S.A. lleva a cabo el siguiente procedimiento para obtener información del personal que labora internamente y el personal capacitado para otras empresas: publicación de la oferta de trabajo,

TESIS CON
FALLA DE ORIGEN

reclutamiento, capacitación, verificación de personalidad, es importante elaborar un inventario de sus recursos humanos.

Cabe señalar que de este procedimiento se seleccionaron datos relevantes que se consideraron para el sistema actual y se determinaron de acuerdo a las políticas de la empresa. Dichos datos son: personales, generales, evaluaciones, áreas designadas, perfiles, fecha de inicio, cursos recibidos, experiencia laboral, puestos desempeñados.

Por lo tanto, se dio paso a la generación de nuevos formatos de búsqueda y análisis de información, todos éstos se manejan por un número de folio, que a continuación se describen y que podrán ser consultados en el Anexo1:

- **Formato 2.3.1 Solicitud de empleo.** Este documento es llenado por el aspirante escribiendo todos sus datos, tanto personales como de estudios y experiencia laboral, se compone de dos hojas. En este formato se solicitan los datos generales de la persona a ingresar, con el fin de obtener información actual y consistente.
- **Formato 2.3.2 Evaluaciones.** Este documento se tiene el registro de datos, que sirve para que los jefes tengan los recursos suficientes para determinar cuál es la situación del personal y si tiene posibilidades de poder cambiar de puesto o empresa cliente, es requisitado por el personal encargado de recursos humanos (se compone de dos hojas). Este formato se llena de acuerdo a las entrevistas realizadas, además se le va explicando a la persona que ingresa, el sueldo que va a recibir y todas sus características.
- **Formato 2.3.3 Expediente.** En este documento se tiene el control del registro de datos del personal, para archivar. El expediente es manejado también por el personal de recursos humanos.

- **Formato 2.3.4 Hoja final del proceso de selección.** En esta hoja se complementan los datos del personal con el fin de tener más consistencia en el manejo de información.

Cabe señalar que la descripción de los formatos anteriores, corresponde al orden cronológico que se sigue para el procedimiento utilizado para selección de personal.

Una vez obtenida la información de los formatos, el personal de recursos humanos los turna al área correspondiente.

Se pueden apreciar dichos formatos en el Anexo 1.

TESIS CON
FALLA DE ORIGEN

2.4 Identificación del problema por áreas.

La figura 2.4.1 representa el organigrama de la empresa Capital Humano S.A., de la cual se detallan los problemas por área.

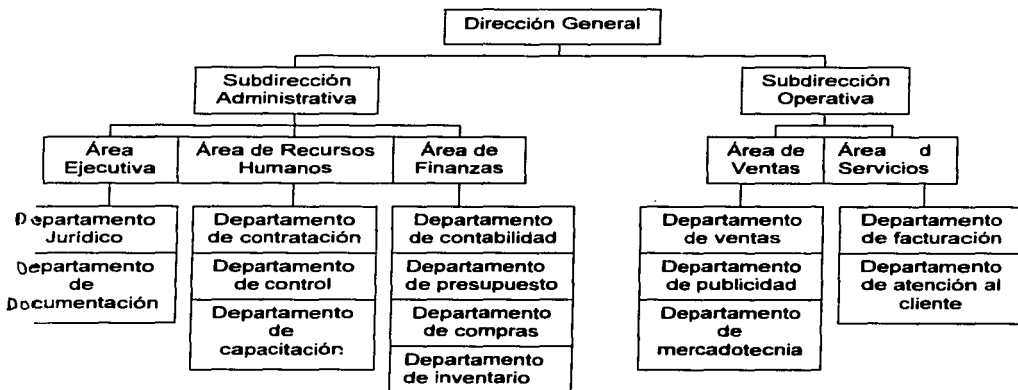


Fig. 2.4.1 Organigrama de la empresa Capital Humano S.A.

2.4.1 Subdirección Administrativa.

2.4.1.1 Área Ejecutiva.

El Área ejecutiva consta de dos departamentos, el Departamento Jurídico y el Departamento de Documentación. En el primero se llevan a cabo todos los procedimientos legales de la empresa tales como elaboración y/o actualización de contratos, recisiones de contrato, autorizaciones de permisos, permisos

especiales, etc. Por ejemplo si un empleado acumula las faltas pertinentes para su despido, se notifica a este departamento y el abogado de la empresa consulta el historial del empleado en la empresa para dictaminar si es necesaria la rescisión de contrato.

El Departamento de Documentación lleva a cabo las actividades referentes al archivo de información y documentos en general sobre los procesos a los que se presenta la empresa.

La problemática del Área Ejecutiva radica en el acceso rápido a la información referente a sus departamentos y la vinculación con los demás. Para llevar a cabo las funciones de manera optimizada y eficaz.

2.4.1.2 Área de Recursos Humanos.

El Área de Recursos Humanos consta de tres departamentos, el Departamento de Contratación, el Departamento de Control y el Departamento de Capacitación.

El Departamento de Contratación realiza los procesos pertinentes para la contratación de personal adecuado a través de exámenes psicométricos, de conocimientos, médicos, etc.

El Departamento de Control se encarga de regular que la calidad de trabajo realizado por el personal sea la adecuada o pertinente para la buena imagen de la empresa y en general de supervisar todo el trabajo del personal de la misma.

En el Departamento de Capacitación se diseñan perfiles de carrera en los que se establecen cursos que los empleados pueden tomar con el fin de seguir creciendo profesionalmente dentro de la empresa.

TESIS CON
FALLA DE ORIGEN

La principal problemática de esta área radica en el proceso de contratación de un recurso humano ya que puede tardar muchos días, porque se acumula una cantidad enorme de documentación de candidatos. En cuanto al área de capacitación, el personal tiene que acudir directamente al departamento correspondiente a solicitar y consultar los posibles cursos de capacitación. Dentro del departamento de control es necesario contar con una herramienta que permita cuantificar correctamente el desempeño de los empleados.

2.4.1.3 Área de Finanzas.

El Área de Finanzas consta de cuatro departamentos los cuales son: Departamento de Contabilidad, el Departamento de Presupuestos, el Departamento de Compras y el Departamento de Inventarios.

El Departamento de Contabilidad realiza todos los procesos contables de la empresa tales como la nómina de los empleados, control de activos y pasivos, así como el control de ingresos y egresos de la empresa sin tener injerencia en la distribución interna.

El Departamento de Presupuesto se encarga de llevar a cabo el ejercicio de los recursos materiales, así como la asignación y distribución de los mismos en la empresa.

El Departamento de Compras realiza las evaluaciones de las necesidades de la empresa para recursos materiales, proveedores y se adquieren los insumos necesario para la correcta operación.

El Departamento de Inventario lleva el control de todos los pasivos de la empresa. Esta área es la responsable de verificar los consumos en la empresa, creación de materiales, precios de los mismos y niveles de inventario. El problema radica en la gran dependencia entre departamentos, ya que es necesario poder

descentralizar la información de cada uno para que se pueda homogeneizar la información de inventarios, ingresos y egresos, etc.

2.4.2 Subdirección Operativa.

2.4.2.1 Área de Ventas.

Esta área consta de dos departamentos, el Departamento de Ventas y el Departamento de Mercadotecnia.

En el Departamento de Ventas se concentra la fuerza de ventas de la empresa. Son los encargados de generar contratos con los clientes.

El Departamento de Mercadotecnia se encarga de estudiar el mercado, analizando las tendencias y posibles soluciones.

El Departamento de Publicidad se encarga de promocionar a la empresa de acuerdo a las necesidades del mercado, así como de llevar una planeación estratégica.

El problema del Área de Ventas es contar con la misma información para poder llevar a cabo las actividades de cada departamento de manera optimizada y eficaz.

2.4.2.2 Área de Servicios.

El Área de Servicios consta de 2 departamentos, el Departamento de Facturación y el Departamento de Atención al Cliente.

El Departamento de Facturación se encarga de generar los procesos de facturación y los cobros a los clientes.

TESIS CON
FALLA DE ORIGEN

El Departamento de Atención a Clientes lleva un seguimiento del servicio con el fin de mantener una buena relación con los clientes y los empleados.

La problemática en esta área es brindar un servicio de calidad al cliente para su entera satisfacción, así como contar con la información íntegra para los procesos de facturación provenientes de las demás áreas.

2.5 Opciones de solución y elección de la óptima.

Para el desarrollo de un sistema se deben tener en cuenta dos factores importantes, un manejador de base de datos (DBMS) que administre la información y; un lenguaje de programación con el cual se realice la interfaz gráfica para que los usuarios puedan interactuar con la aplicación.

2.5.1 Eligiendo una base de datos.

La elección de una base de datos ha sido, por mucho tiempo, un punto de discusión necesario dentro de los departamentos de sistemas, debido a la importancia que implica para la organización. Para aplicaciones de misión crítica esta discusión es aún mucho más importante, y los factores que inciden dentro de la decisión de adoptar una u otra plataforma son muy variados y complejos, generalmente la inversión a realizar en una solución comercial para base de datos es muy alta, y a veces injustificada. Es importante considerar que la inversión inicial en la compra de una plataforma de base de datos no necesariamente será la única porque generalmente existen costos extras para capacitación, mantenimiento y administración, entre otros.

TESIS CON
FALLA DE ORIGEN

2.5.1.1 MySQL.

Un nuevo contendiente dentro del mercado de base de datos es MySQL, se trata de una base de datos Open Source (de Código Abierto) y, por lo tanto, sin costo. MySQL tiene un rendimiento excelente en estabilidad, facilidad de afinación y conectividad. Tiene como lenguaje de base de datos al Lenguaje de Consulta Estructurado (SQL) que es un lenguaje estandarizado para la administración de los datos. Considerando su facilidad de uso, MySQL se posiciona como la base de datos que tiene el mejor rendimiento para uso en aplicaciones. Es considerado como un veloz manejador de base de datos, dependiendo del tamaño de la base de datos y del número de operaciones que se realicen. Puede funcionar en diferentes sistemas operativos como Linux, Microsoft Windows, HP-UX, Open-BSD, Sun Solaris, SunOS4.

Sin embargo, esta base de datos todavía no nos ofrece todas las funcionalidades que nos ofrecen otros manejadores de bases de datos, podemos enumerar: clustering, alta disponibilidad, procesamiento OLAP, etc. Su principal objetivo de diseño fue la velocidad. Se sacrificaron algunas características esenciales en sistemas más "serios" con este fin. Otra característica importante es que consume muy pocos recursos, tanto de CPU como de memoria. Licencia GPL a partir de la versión 3.23.19.

2.5.1.1.1 Ventajas: Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás. Mejores utilidades de administración (backup, recuperación de errores, etc). No suele perder información ni corromper los datos en caso de se traben un proceso. Mejor integración con PHP. No hay límites en el tamaño de los registros. Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.

2.5.1.1.2 Inconvenientes: No soporta transacciones, *roll-backs* ni *subselects*. No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

2.5.1.2 PostgreSQL.

PostgreSQL ofrece muchas ventajas respecto a otros sistemas de bases de datos: Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede ser demandado por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software. Esto tiene varias ventajas adicionales:

- Modelos de negocios más rentables con instalaciones a gran escala.
- No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.

PostgreSQL intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase. Licencia BSD.

2.5.1.2.1 Ventajas: Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM. Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial). Tiene mejor soporte para triggers y procedimientos en el servidor. Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.

PostgreSQL tiene mejor soporte que los proveedores comerciales además es Multiplataforma, PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas. PostgreSQL está diseñado para ambientes de alto volumen, usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes

volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología por las mismas razones. Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora).

Una lista breve de características técnicas que PostgreSQL ofrece:

- Cumple con ANSI SQL
- Integridad referencial
- Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de réplica
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby
- Reglas
- Vistas
- Triggers
- Unicode
- Secuencias
- Herencia
- Outer Joins
- Sub-selects
- Una API abierta
- Procedimientos almacenados
- Soporte nativo SSL
- Lenguajes procedurales
- Índices parciales y funcionales
- Autenticación Kerberos nativa
- Soporte para consultas con UNION, UNION ALL y EXCEPT
- Extensiones para SHA1, MD5, XML y otras funcionalidades
- Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL

- Sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario y rápido desarrollo de nuevos tipos
- Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL

2.5.1.3 Mini SQL.

Mini SQL es un motor de base de datos de peso ligero disponible vía Internet. Aunque no es gratuito, se permite su uso para instituciones educativas sin ningún costo. Por la gran velocidad con que realiza sus operaciones, sus escasos requerimientos y la simplicidad de su implementación es ideal como soporte de base de datos para aplicaciones WEB. Otra de las características interesantes es que está disponible en versiones para UNIX y Win 32, por lo que permite desarrollar aplicaciones usando el entorno Win 32 y luego pasarlas a UNIX para su operación, si se desea.

2.5.1.3.1 Ventajas: Puede manejar bases de datos de un tamaño considerable con buenos tiempos de respuesta. Como buen sistema de base de datos relacional, puede manejar diferentes tablas dentro de una misma base y crear relaciones entre ellas. No es necesario realizar un desglose de los campos enviados desde el formulario para obtener el valor de las variables. Con w3-mysql se puede utilizar una variable con el nombre asignado en el atributo **NAME** del campo de formulario y esta tendrá el valor con el que fue enviado.

2.5.1.3.2 Desventajas: No basta con conocer los comandos de *LITE*³ para realizar las consultas, se debe además tener conocimientos básicos de la diferencia de los sistemas Isis. Mini-SQL requiere estar ejecutándose como un proceso permanente dentro del servidor para efectuar consultas a través del Web.

Características	MySQL	PostgreSQL	mSQL
Popularidad	Bueno	Bueno	Regular

³ Pequeño SQL del cual Mini SQL es propietario y que sirve para crear las consultas.

Facilidad de Aprendizaje	Bueno	Bueno	Regular
Interfaz amigable	Regular	Regular	Regular
Manejo de grandes Bases de datos	Regular	Bueno	regular
Independencia de la arquitectura	Bueno	Bueno	Regular
Intercambio de información con otras aplicaciones	Bueno	Bueno	Regular

Tabla 2.5.1 Tabla de comparación

El producto seleccionado fue PostgreSQL debido a que el sistema manejará una base de datos de gran tamaño. PostgreSQL es un manejador de base de datos relacional y es de tipo libre esto hace que el costo sea mínimo. Aunque la interfaz gráfica no es sencilla, actualmente existen varias herramientas que lo hacen más amigable como PGAcces. PostgreSQL también ofrece la posibilidad de emigrar con facilidad, a otro sistema manejador por la sencillez de su diseño de tablas, relaciones y la seguridad en la integridad de los datos.

TESIS CON
FALLA DE ORIGEN

2.5.2 Front-End.

El Front-End se refiere a la interfaz gráfica que permite al usuario interactuar con el sistema de forma amigable. Existen varios lenguajes de programación vía web que logran este objetivo. Las opciones que se encuentran más atractivas por sus características son PHP, JSP y PERL.

2.5.2.1 PERL.

PERL significa Practical Extraction and Report Language, es decir; lenguaje práctico de extracción y de informes. Es un lenguaje con el objetivo principal de simplificar tareas de administración de unos sistemas UNIX. Es un lenguaje interpretado que tiene varias utilidades. Pero está principalmente orientado a la búsqueda, extracción y formateo de archivos tipo texto. También es muy usado para manejo y gestión de procesos (estado de procesos, conteo y extracción de parámetros característicos).

Es una combinación de las características de los lenguajes más usados por los programadores de sistemas, como son los shell del sistema operativo UNIX, los de utilidad (que incluye un lenguaje interpretado propio) awk para formateo y tratamiento de texto e incluso de características de pascal, aunque su potencia se basa en la similitud con las mejores características de lenguaje estructurado C. En general cualquier utilidad que se necesite realizar en sh, awk o sed, se puede implementar de una manera más potente y sencilla mediante el lenguaje PERL.

Algunas ventajas del uso del lenguaje PERL son las siguientes:

- Construcción de pequeños programas que pueden ser usados como filtros para obtener información de ficheros, realizar búsquedas
- Se puede utilizar en varios entornos, como puede ser Windows 95, OS/2....., sin realizar cambios de código, siendo únicamente necesario la introducción del intérprete PERL correspondiente a cada sistema operativo.

- También es uno de los lenguajes más utilizados en la programación de CGI scripts, que son guiones o scripts (Common Gateway Interface), para intercambio de información entre aplicaciones externas y servicios de información. Como ejemplo de ello tenemos los programas de búsqueda usados por el browser Netscape.
- El mantenimiento y depuración de un programa en PERL es mucho más sencillo que la de cualquier programa en C.

Es un lenguaje que hereda estructuras principales de los intérpretes de comandos de UNIX, especialmente el csh y de otras utilidades estándar, como awk y sed. En realidad, puede hacer todo lo que hacen cualquiera de ellos y todos ellos juntos, y la mayoría de las veces de forma más simple, comprensible y fácil de depurar. PERL es un lenguaje interpretado, aunque en realidad, el intérprete de PERL, como todos los intérpretes modernos, compila los programas antes de ejecutarlos. Por eso se habla de scripts y no de programas, concepto referido principalmente a programas compilados al lenguaje de máquina nativo del computador y sistema operativo en el que se ejecuta.

2.5.2.2 PHP.

Es un potente lenguaje script del lado del servidor, se utiliza principalmente para generar páginas dinámicas.

2.5.2.2.1 Características de PHP.

Multiplataforma: PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.

Sencillez: La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C incluimos al Java y Javascript, de hecho mucha de la

funcionalidad del PHP se la debe al C en funciones como `fread()` o `srllen()`, así que muchos programadores se sentirán como en casa.

Rapidez. PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

Base de Datos: Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre se podrá disponer de ODBC para situaciones que lo requieran.

Variedad: Se cuenta con varios módulos para cuando un programador PHP necesite una interfaz para una biblioteca en particular, fácilmente podrá crear una API para ésta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.

2.5.2.3 JSP.

Java Server Page es un lenguaje de script del servidor basado en Java que permite mezclar HTML estático con páginas HTML generadas dinámicamente. Se crea la página HTML estática y se le añaden tags especialmente para hacerlas dinámicas. JSP funciona en diversos servidores web. JSP se compila en un servlet de Java. El servlet es posteriormente compilado en bytecodes de Java. JSP es esencialmente independiente de la plataforma y del servidor. JSP es más fácilmente extensible, los componentes JSP son reutilizables de distintas plataformas. JSP se beneficia del modelo de seguridad de Java. JSP es una extensión de la tecnología Java Servlets. Mientras que estos últimos tienen que mantener plantillas de código HTML dentro del programa, JSP contiene estas plantillas dentro de las propias páginas. La tecnología JSP está ideada para facilitar la creación de aplicaciones web. Separación entre generación de contenido y presentación. Mediante etiquetas HTML o XML se da formato a la página, y mediante etiquetas JSP, código script o JavaBeans se añade el

componente dinámico. De esta forma, encapsulando la lógica, se puede modificar la presentación de la página JSP sin afectar a la generación del contenido.

Reutilización de componentes de software: La descomposición en componentes que caracteriza a esta tecnología facilita el que objetos de una aplicación vuelvan a utilizarse en dominios diferentes. Por ejemplo, objetos como el analizador léxico o el extractor de raíces definidos para el glosario, pueden ser reutilizados para desarrollar otro tipo de aplicaciones en las que se necesite integrar técnicas de RI.

Utilización de etiquetas XML para simplificar el desarrollo de las páginas: Así se ocultan detalles de implementación y se facilita la construcción de páginas que integren código JSP, incluso a desarrolladores web poco acostumbrados a trabajar con lenguajes script. A través del mecanismo de extensibilidad de las librerías de etiquetas, estas pueden personalizarse para las necesidades de cada sistema.

Independencia de la plataforma tanto cliente como servidor: Por un lado, la utilización de código Java garantiza la portabilidad de la aplicación para su ejecución en cualquier servidor que contenga una máquina virtual Java. Esta es una ventaja substancial frente a otras tecnologías similares. En el caso del cliente, al recibir sólo páginas HTML hace que sea compatible con cualquier navegador.

Funcionamiento: Cuando se produce una solicitud de una página JSP por parte de un navegador, se comienza comprobando si se trata de la primera solicitud de dicha página. En ese caso se compila a un servlet, que es ejecutado y cuya salida es devuelta al usuario que realizó la petición. Las siguientes solicitudes son más eficientes al no necesitar que dicha página sea de nuevo compilada, ya que sólo se invoca al servlet que se generó en la petición inicial. En este sentido esta tecnología es más ventajosa que otras como ASP, que requieren cada vez una nueva compilación. El proceso de compilación de las páginas consiste en analizar

su contenido buscando etiquetas JSP y traduciendo éstas a código Java equivalente. El contenido estático de las páginas (código html) es traducido a cadenas de caracteres en lenguaje Java. Las etiquetas de componentes JavaBeans son traducidas a su correspondiente .application: es accesible durante el tiempo que dure la aplicación en la que se usa.

En lo referente a sus características, JSP crea páginas web dinámicamente desde páginas HTML marcadas con unas etiquetas especiales. Soportan la inclusión de código directamente en la página, así como el uso de componentes encapsulados: componentes COM en ASP y JavaBeans en JSP. La mayor diferencia entre las dos tecnologías es la compatibilidad de plataforma. JSP está disponible para cualquier servidor web y cualquier plataforma con soporte para Java Servlets. Además, el motor JSP está escrito completamente en Java. Otra diferencia es que las páginas JSP se compilan la primera vez que son solicitadas mientras que las siguientes solicitudes se cargan más rápidamente debido a que quedan residentes en la memoria del servidor.

Características	PHP	JSP	PERL
Servidor Web	Apache, IIS, etc	Apache, Tomcat, Resin, SUN One JRun	Apache, IIS
Plataforma	UNIX, Windows	UNIX, WINDOWS	UNIX, WINDOWS
Lenguaje Script	Si	Si	Si
Seguridad contra caídas del sistema	Si	Si	Si
Compatibilidad con Bases de datos	Si	Si	No
Soporte de Herramientas	Si	Si	Si
Fácil aprendizaje	Si	Si	No
Gratuito	Si	Si	Si

Tabla 2.5.2 Componente del Front-End

Conclusión:

PHP y PostgreSQL son dos de las mejores tecnologías que existen actualmente. Su velocidad y robustez los caracteriza. PHP es rápido, compacto, simple y está diseñado específicamente para pre-procesamiento HTML y además es fácil hacer una interfaz con PostgreSQL (y otros servidores de bases de datos). PostgreSQL es un manejador de base de datos robusto, diseñado para ambientes de alto volumen, multiplataforma y además no hay costo asociado a la licencia del software.

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN

CAPÍTULO III TEORÍA BÁSICA

*"Dadme un punto de apoyo y
moveré al mundo".
— Arquímedes*

3.1 Características, ventajas y desventajas de las bases de datos relacionales

3.1.1 Concepto de base de datos.

La idea de base de datos surge como una necesidad de mantener datos relacionados. Las definiciones de base de datos son numerosas, pero todas coinciden en que es un conjunto de datos almacenados con la capacidad de acceder directamente a ellos, los cuales están interrelacionados y estructurados de acuerdo a un modelo capaz de recoger el máximo contenido semántico.

Definición 1:

"Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una o más aplicaciones de la mejor forma posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados".⁴

Definición 2:

"Colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios".⁵

TESIS CON
FALLA DE ORIGEN

⁴ Martin, 1975.

⁵ Deen, 1985.

3.1.2 Características y definición de dato.

Un dato es la mínima unidad de información y la información es un conjunto de datos con la característica de que han sido procesados. Sus características son:

- No efímero, es decir; permanece con el paso del tiempo.
- Estructurado, para que facilite el compartirlos por aquellos que lo necesiten.
- Operacional Transaccional (OLTP), manipularlos aplicando operadores para obtener resultados.
- Tiene sentido semántico.
- Íntegro, en el sentido que refleja una realidad existente.

3.1.3 Características generales de una base de datos.

- Considera interrelaciones entre datos y restricciones semánticas que deben estar presentes en una base de datos. Una base de datos no sólo debe almacenar entidades y atributos, sino que también debe almacenar interrelaciones entre datos.
- La redundancia de datos debe ser controlada, de forma que no existan duplicidades perjudiciales ni innecesarias. En las bases de datos no está permitida la redundancia lógica, pero sí se admite cierta redundancia física por motivos de eficiencia.
- Las bases de datos pretenden servir a toda la organización, es decir; a múltiples usuarios y a diferentes aplicaciones.
- La independencia lógica y física de los datos.
- La definición y descripción del conjunto de datos contenido en la base debe ser única e integrada con los mismos datos.
- La actualización y recuperación de las bases de datos debe realizarse mediante procesos bien determinados, de modo que se mantenga la integridad, seguridad y confidencialidad de la base.

3.1.4 Las características particulares de una base de datos.

El objetivo de disminuir la redundancia de un conjunto de datos determina dos características fundamentales que poseerá cualquier Base de Datos:

Integrada: se entiende que una base de datos puede considerarse como una unificación de varios archivos de datos independientes, donde se elimina parcial o totalmente cualquier redundancia entre los mismos.

Compartida: Se entiende que partes individuales de la Base de Datos pueden compartirse entre varios usuarios distintos, en el sentido que cada uno de ellos puede tener acceso a la misma parte de la Base de Datos y utilizarla con propósitos diferentes, es decir; la integración.

3.1.5 Ventajas de las bases de datos.

La tabla 3.1.5.1 describe las ventajas de las bases de datos:

Con respecto a	Ventajas
Los datos	<ul style="list-style-type: none"> • Independencia de éstos respecto de los tratamientos y viceversa • Mejor disponibilidad de los mismos • Mayor eficiencia en la consulta, codificación y entrada
Los resultados	<ul style="list-style-type: none"> • Mayor coherencia • Mayor valor informativo • Mejor y documentación más normalizada de la información
Los usuarios	<ul style="list-style-type: none"> • Acceso más rápido y sencillo de los usuarios finales • Más facilidades para compartir los datos por el conjunto de los usuarios • Mayor flexibilidad para atender a demandas cambiantes.

Tabla 3.1.5.1 Ventajas de las bases de datos.

**TESIS CON
FALLA DE ORIGEN**

3.1.6 Desventajas de las bases de datos.

La tabla 3.1.6.1 describe las desventajas de las bases de datos:

Con respecto a	Desventajas
La implantación	<ul style="list-style-type: none"> • Costosa en equipos (lógico y físico) • Ausencia de estándares • Larga y difícil puesta en marcha • Rentabilidad a mediano plazo
Los usuarios	<ul style="list-style-type: none"> • Personal especializado • Desfase entre teoría y práctica

Tabla 3.1.6.1 Desventajas de las bases de datos.

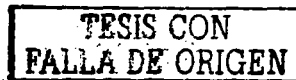
La instalación costosa se refiere a equipos, nuevas instalaciones o actualizaciones, sistemas operativos, compiladores, SGBD (Sistema de Gestión de Bases de Datos) comerciales, computadoras más potentes en rendimiento, procesamiento, etc.

Personal especializado: es clave para la administración de la base de datos por que se requiere de conocimientos específicos.

3.1.7 Componentes de los sistemas de bases de datos.

Los componentes de un sistema de bases de datos son:

- La base de datos, en donde tenemos nuestra colección de datos interrelacionados entre si.
- El Sistema de Gestión de Bases de Datos (SGBD, DBMS) o motor, tal como Oracle, PostgreSQL, Sybase, etc. Éste nos permite administrar nuestra base de datos.



- Programas de aplicación, los cuales interactúan con nuestra base de datos.
- Usuarios. Los cuales pueden ser finales, administradores de bases de datos (DBA), programadores, etc.
- Computadoras, nuestras herramientas o dispositivos físicos de almacenamiento.
- Programas de aplicación específicos como generadores de informes, de interfaces, herramientas de desarrollo, de administración, etc.

En la Figura 3.1.7.1 se detalla un esquema general de la arquitectura de una base de datos, las relaciones entre ellos y la base de datos lógica.

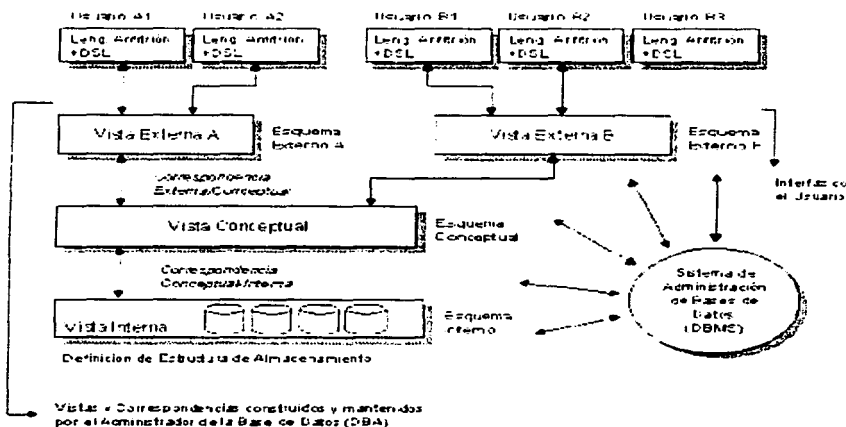


Figura 3.1.7.1 Esquema general de la arquitectura de una base de datos

TESIS CON
FALLA DE ORIGEN

Descripción de la figura 3.1.7.1

Una vista externa es una visión particular de un usuario o un grupo de usuarios de la base de datos. El esquema externo representa una forma de definición o formalización de esta vista externa.

La vista conceptual pretende ser la representación total y abstracta de los datos que componen la base; la formalización de ésta se logra mediante el esquema conceptual.

La vista interna es de un nivel muy bajo y corresponde al almacenamiento físico de los datos de la base como los tipos de registros almacenados.

Las correspondencias se pueden definir como una asociación de distintas representaciones para un mismo dato.

Un DSL es un sub-lenguaje de datos, es una combinación de dos lenguajes: un lenguaje de definición de datos (DDL) y un lenguaje de manipulación de datos (DML). Este lenguaje representa un intermediario entre el sistema de base de datos y algún lenguaje como C, FORTRAN etc. El DSL provee herramientas a los lenguajes tradicionales para que se integren al sistema de base de datos. Existen distintos tipos de DSL para un mismo sistema.

DBMS (Sistema de Administración de Bases de Datos) es el software que maneja todos los accesos a la base de datos

La interfaz con el usuario (Front-End) es el límite de acceso que tiene un usuario común a la base, todo lo que está bajo este límite es transparente y desconocido para él (Back-End).

El administrador de bases de datos (DBA) corresponde a la persona o grupo de personas encargadas del control general de la base de datos. Sus funciones son:

- Decidir el contenido de la base de datos: comprende la identificación de entidades de interés para la organización y los datos a registrar de estas entidades. Luego se define el contenido de la base de datos generando un modelo conceptual.
- Decidir la estructura de almacenamiento y la estrategia de acceso: esto es decidir como deben representarse los datos en forma interna y hacer la correspondencia entre estos y el modelo conceptual ya definido.
- Vincularse con los usuarios: comprende toda una labor de prestación de servicios que busca garantizar la existencia, en la base, de los datos necesarios y la formalización de los distintos esquemas externos.
- Definir los controles de autorización y procedimientos de validación: involucra la definición de restricciones de seguridad y protección para la conservación de la integridad de los datos.
- Definir una estrategia de respaldo y recuperación: esto corresponde a un esquema de seguridad más amplio que lo anterior y, básicamente, su objetivo es la operación exitosa del sistema.
- Controles de desempeño y responder a los cambios de requerimientos: con el fin de lograr un desempeño aceptable, según expectativas, del Sistema mediante mecanismos de control.

3.1.8 El sistema de gestión de bases de datos (SGBD o DBMS).

Un sistema de gestión de bases de datos consiste de una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. La colección de datos es la base de datos, y es la que contiene información por ejemplo acerca de una empresa determinada.

El objetivo principal de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información en la base de datos.

TESIS CON
FALLA DE ORIGEN

3.1.9 Modelo relacional.

La razón por la cual el *Modelo Relacional* y los *Administradores de Bases de Datos Relacionales* (ABDR; en inglés RDBMS, o *Relational Database Management Systems*) se han hecho muy populares y usados en los últimos veinte años; es que el *Modelo Relacional* propone una solución unificada, de alto nivel y con un sólido fundamento teórico al problema del almacenamiento, recuperación y mantenimiento de datos.

3.1.10 Objetivos del modelo relacional.

El trabajo publicado por Codd presentaba un nuevo modelo de datos que perseguía una serie de objetivos, que se resumen en los siguientes:

- Independencia física. El modo en el que se almacenan los datos no influye en su manipulación lógica y por tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico.
- Independencia lógica. El añadir, eliminar o modificar objetos de la base de datos no repercute en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).
- Flexibilidad. En el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera.
- Uniformidad. Las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
- Sencillez. Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil de comprender y de utilizar por parte del usuario final.

El modelo relacional se divide en 3 partes: estructura, integridad y manipulación de los datos.

3.1.11 Estructura del modelo relacional.

La *relación* es el elemento básico del modelo relacional y se representa por una *tabla*. En la tabla 3.1.10.1 se detallan los términos.

Relación	Tabla
Tupla	Fila
Atributo	Columna
Número de tuplas	Cardinalidad
Número de atributos	Grado
Dominio	Colección de valores, de los cuales uno o mas atributos obtienen sus valores reales
Llave primaria	Identificador único para la tabla, es decir, una columna o combinación de columnas con la propiedad de que nunca existen 2 filas de la tabla con el mismo valor en esa columna o combinación de columnas

Tabla 3.1.11.1

Es importante señalar que la tabla es plana en el sentido de que el cruce de una fila y una columna sólo puede dar un valor, es decir; no se admiten atributos multivaluados.

3.1.12 Claves o Llaves.

Una *clave* o *llave candidata* de una relación es un conjunto no vacío de atributos que identifican unívoca y mínimamente cada tupla. Toda relación siempre tendrá una llave candidata.

TESIS CON
FALLA DE ORIGEN

Clave o llave primaria: es aquella llave candidata que el usuario elegirá, por consideraciones ajenas al modelo relacional, para identificar las tuplas de la relación. El modelo relacional no incluye este concepto de elegir una llave como primaria, cuando hay varias candidatas.

Clave o llave alternativas: Son aquellas llaves candidatas que no han sido escogidas como llaves primarias.

Clave o llave ajena o foránea: de una relación R2 es un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la llave primaria de una relación R1 (R1 y R2 no son necesariamente distintas). Notar que la llave ajena y la correspondiente llave primaria han de estar definidas sobre los mismos dominios.

3.1.13 Proceso de normalización.

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

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

Primera forma normal (1NF).

Se dice que una tabla se encuentra en primera forma normal (1NF) si y solo si cada uno de los campos contiene un único valor o valor atómico para un registro determinado.

Segunda forma normal (2NF).

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

Tercera forma normal (3NF).

Se dice que una tabla está en tercera forma normal si y sólo si los campos de la tabla dependen únicamente de la llave, dicho en otras palabras los campos de las tablas no dependen unos de otros.

3.1.14 Interrelaciones y cardinalidad.

La Cardinalidad son relaciones que existen entre varias tablas de la base de datos (Clientes y Pedidos, por ejemplo). Existen tres formas de interrelaciones dependiendo de la cardinalidad con la que se combinan los elementos de ambas tablas.

3.1.15 Relaciones uno a uno.

Una interrelación es de uno a uno entre la tabla A y la tabla B cuando a cada elemento de la tabla A se le asigna un único elemento de la tabla B y para cada elemento de la tabla B contiene un único elemento en la tabla A.

3.1.16 Relaciones uno a muchos.

Una interrelación es de uno a varios entre las tablas A y B cuando un elemento de la tabla A posee varios elementos relacionados en la tabla B y cuando un elemento de la tabla B posee un único elemento relacionado en la tabla A.

TESIS CON
FALLA DE ORIGEN

3.1.17 Relaciones muchos a muchos.

Una interrelación es de varios a varios entre las tablas A y B cuando un elemento de la tabla A posee varios elementos relacionados en la tabla B y cuando un elemento de la tabla B posee varios elementos relacionados en la tabla A.

3.1.18 Álgebra relacional.

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

1. Unión
2. Intersección
3. Diferencia
4. Producto
5. Selección
6. Proyección
7. Reunión
8. División
9. Asignación

Las cuatro primeras se basan en la teoría de conjunto en las matemáticas; las cuatro siguientes son operaciones propias del álgebra relacional y la última es la operación estándar de otorgar un valor a un elemento.

TESIS CON
FALLA DE ORIGEN

Unión.

La operación de unión permite combinar datos de varias relaciones. Para poder realizar esta operación es necesario e imprescindible que las tablas a unir tengan las mismas estructuras y que sus campos sean iguales.

Intersección.

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

Diferencia.

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

Producto.

La operación producto consiste en la realización de un producto cartesiano entre dos tablas dando como resultado todas las posibles combinaciones entre los registros de la primera y los registros de la segunda. El producto se muestra en la figura 3.1.18.1

Tabla A

X	Y
10	22
11	25

Tabla B

W	Z
33	54
37	98
42	100

TESIS CON
FALLA DE ORIGEN

El producto de $A * B$ daría como resultado la siguiente tabla:

Tabla A * Tabla B

10	22	33	54
10	22	37	98
10	22	42	100
11	25	33	54
11	25	37	98
11	25	42	100

Figura 3.1.18.1 Ejemplo de la operación Producto.

Selección.

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

Proyección.

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

Reunión.

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

División.

La operación división es la operación contraria al producto. La operación se muestra en el siguiente ejemplo.

Lo primero es extraer en una tabla todos los códigos de todos los productos, a esta tabla la denominamos A.

Tabla A
Código Producto

1035
2241
2249
5818

Figura 3.1.18.2

En una segunda tabla extraemos, de la tabla de ventas por ejemplo, el código del producto y el código comercial que lo ha vendido, lo hacemos con una proyección y evitamos traer valores duplicados. El resultado podría ser el siguiente de acuerdo a la figura 3.1.18.3.

Tabla B
Código Comercial Código Producto

10	2241
23	5818
23	1035
39	5818
37	5818
10	2249
23	2249
23	2241

Figura 3.1.18.3

TESIS CON
FALLA DE ORIGEN

Si dividimos la tabla B entre la tabla A obtendremos como resultado una tercera tabla que:

1. Los campos que contiene son aquellos de la tabla B que no existen en la tabla A. En este caso el campo Código Comercial es el único de la tabla B que no existen en la tabla A.
2. Un registro se encuentra en la tabla resultado si y sólo si está asociado en tabla B con cada fila de la tabla A (figura 3.1.17.4):

Tabla Resultado
Código Comercial

23

Figura 3.1.18.4

El código comercial 23 es el único de la tabla B que tiene asociados todos los posibles códigos de producto de la tabla A.

Asignación.

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

3.1.19 Cálculo relacional.

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

1. Una lista de resultados
2. Una sentencia de calificación

La lista de resultados está clara, son aquellos registros que cumplen las condiciones que deseamos. La sentencia de calificación contiene las condiciones

que deseamos que cumplan los registros de la lista de resultados. La diferencia entre el cálculo y el álgebra radica en que el cálculo realiza la operación en un único paso, sin necesidad de tener que obtener tablas intermedias, el álgebra realiza las operaciones paso a paso.

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

El cálculo relacional incluye un concepto nuevo denominado cuantificador, los cuantificadores tratan de averiguar el número de registros afectados por una determinada operación, incluso antes de realizarla. Se dividen en:

Cuantificadores existenciales.

Son aquellos que tratan de averiguar el número de registros que devolvería un tipo de consulta.

Cuantificadores universales.

Son aquellos que indican que una condición se aplica a todas las filas de algún tipo. Se usa para brindar la misma capacidad que la operación división del álgebra relacional.

3.2 Características, ventajas y desventajas de POSTGRESQL 7.2

3.2.1 ¿Qué es PostgreSQL?



Fig. 3.2.1 Logotipo de PostgreSQL.

PostgreSQL es un ORDBMS (Sistema de Gestión de Bases de Datos Objeto-Relacionales), desarrollado desde el año de 1977. Comenzó como un proyecto

denominado *Ingres* en la Universidad Berkeley de California. *Ingres* fue más tarde desarrollado comercialmente por la *Relational Technologies/Ingres Corporation*.

En 1986 otro equipo dirigido por *Michael Stonebraker* de Berkeley continuó el desarrollo del código de *Ingres* para crear un sistema de bases de datos objeto-relacionales llamado *Postgres*. En 1996, *Postgres* fue renombrado a *PostgreSQL*. El proyecto *PostgreSQL* está en un continuo proceso de desarrollo por la comunidad de código abierto.

3.2.2 Versión Open Source (Código Abierto).

PostgreSQL está considerado como un sistema avanzado de bases de datos de código abierto, está liberado bajo la licencia BSD, lo que significa que cualquiera puede disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente.

Código abierto, por definición, significa que una persona puede obtener el código fuente, usar el programa, y modificarlo libremente sin las limitaciones del software propietario. En el mundo de las bases de datos de código abierto significa que se tiene acceso real a números de benchmarking y estadísticas de rendimiento, datos que otras compañías como Oracle no facilitan. El código abierto se puede modificar para adaptarlo a necesidades propias.

3.2.3 Soporte Comercial.

Existen varias compañías que proporcionan servicios de consultoría a la comunidad PostgreSQL.

- Command Prompt, Inc. (<http://www.commandprompt.com/>). *Command Prompt, Inc.* es una compañía de servicios Linux y soporte PostgreSQL. Proporcionan soporte sobre Linux y PostgreSQL, incluyendo

programación en C & C++, Java, PHP, Perl, y su propio servidor de aplicaciones LXP para PostgreSQL.

- Cybertec Geschwinde & Schvnig OEG (<http://postgres.cybertec.at/>) *Cybertec* proporciona cursos, soporte, consultoría, y sistemas finales de coste optimizado.
- DbExperts (<http://www.dbexperts.com.br/>). *dbExperts* ofrece cursos de formación, soporte especializado para desarrolladores y productos comerciales para PostgreSQL. *dbExperts* está ubicado en Brasil.
- PostgreSQL, Inc. (<http://www.pgsql.com/>) *PostgreSQL, Inc.* ofrece soporte para PostgreSQL, hosting de bases de datos y materiales promocionales.
- Software Research Associates (<http://osb.sra.co.jp/>) *Software Research Associates* ofrece diversos servicios para ayudar a los clientes con sistemas basados en software de código abierto.

3.2.4 Soporte de la Comunidad.

El soporte a los usuarios de PostgreSQL se da a través de listas de correo desde www.postgresql.org. A través de dichas listas es posible mantener contacto con personas especializadas en el tema y recibir orientación técnica, información para la solución de problemas, respuestas a preguntas mas frecuentes (FAQ's), etc.

3.2.5 Características de PostgreSQL.

PostgreSQL posee muchas características que antes sólo se podían ver en grandes productos comerciales como DB2 u Oracle. Lo que permite que muchas organizaciones, incluyendo grandes corporaciones, instituciones gubernamentales y pequeños negocios utilicen PostgreSQL para manejar en línea sus datos más valiosos y aplicaciones de misión crítica.

- **DBMS Objeto-Relacional.**

TESIS CON
FALLA DE ORIGEN

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar rutinas y reglas complejas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y arreglos.

- **Extensible.**

PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

- **Soporte SQL.**

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

- **Integridad referencial.**

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la BD.

- **API Flexible.**

La flexibilidad del API de PostgreSQL permite contar con soporte en desarrollo para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

- **Lenguajes Procedurales.**

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al

lenguaje procedurales de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje de procedimientos.

- **MVCC**

MVCC, o Control de Concurrencia Multiversión (Multi Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. En algunos Sistemas Manejadores de Bases de Datos (DBMS) con capacidades SQL, tal como MySQL o Access, es necesario esperar para acceder a la información de la base de datos. La espera está provocada por usuarios que se encuentran escribiendo en la base de datos, es decir; el usuario que desea leer información está bloqueado o los usuarios están actualizando registros.

Mediante el uso de MVCC, PostgreSQL evita este problema por completo. MVCC está considerado como una alternativa mejor que el bloqueo a nivel de fila, porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

- **Cliente-Servidor**

PostgreSQL se basa en una arquitectura proceso-por-usuario cliente/servidor. Similar al método que utiliza Apache 1.3.x para manejar procesos, en donde existe un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectarse a PostgreSQL.

TESIS CON
FALLA DE ORIGEN

- **Write Ahead Logging (WAL)**

Esta característica de PostgreSQL conocida como *Write Ahead Logging* incrementa la dependencia de la base de datos al registro de cambios antes de que éstos sean escritos en la misma, garantiza que de existir un fallo habrá un registro de las transacciones a partir del cual podremos restaurar la base de datos. Es muy útil cuando existe una falla que origina una "caída" en la base y recuperar datos con el último registro. Una vez que el sistema ha quedado restaurado, el usuario puede continuar trabajando desde el punto en que lo dejó antes de la falla.

3.2.6 Catálogo del sistema PostgreSQL.

Mediante tablas internas PostgreSQL utiliza catálogos para mantener el sistema. En la Tabla 3.2.5.1 se muestra el catálogo del sistema PostgreSQL. Cada base de datos tiene estas mismas tablas, salvo por la primera que es única, ya que almacenan cada una de las partes que componen la base de datos.

Nombre del catálogo	Descripción
Pg_database	Bases de datos
Pg_class	Clases o tablas
Pg_attribute	Atributos o campos de la clase o tabla
Pg_index	Índices secundarios
Pg_proc	Procedimientos (en C y en SQL)
Pg_type	Tipos de datos (del sistema y definidos por el usuario)
Pg_operator	Operadores (del sistema y definidos por el usuario)
Pg_aggregate	Agregados y funciones agregadas
Pg_am	Métodos de acceso
Pg_amop	Operadores de métodos de acceso

Pg_amproc	Funciones de soporte para métodos de acceso
Pg_opclass	Clases de operadores de métodos de acceso

Tabla. 3.2.5.1

En resumen PostgreSQL tiene estas características técnicas:

- Cumple completamente con ACID
- Cumple con ANSI SQL
- Integridad referencial
- Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de réplica
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby.
- Reglas, vistas y triggers.
- Unicode.
- Secuencias.
- Herencia.
- Outer Joins.
- Sub-selects.
- Una API abierta.
- Procedimientos almacenados.
- Soporte nativo SSL.
- Lenguajes procedurales.
- Respaldo en caliente.
- Bloqueo a nivel fila.
- Índices parciales y funcionales.
- Autenticación por kerberos de forma nativa.
- Soporte para consultas con UNION, UNION ALL y EXCEPT.
- Extensiones para SHA1, MD5, XML y otras funcionalidades.
- Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.

- Sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario, y rápido desarrollo de nuevos tipos.
- Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL.

3.2.7 Ventajas de PostgreSQL.

- Instalación ilimitada:

Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, no hay costo asociado a la licencia del software. Lo que se traduce en :

- Modelos de negocios más rentables con instalaciones a gran escala.
- No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.
- Soporte
- Disminución en costos de operación
- Estabilidad y confiabilidad
- Multiplataforma:
PostgreSQL está disponible para cualquier versión de Unix y una versión nativa de Windows.
- Diseñado para ambientes de alto volumen:
PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mucho mejor respuesta en ambientes de grandes volúmenes.
- Herramientas gráficas de diseño y administración de bases de datos:
Existen varias herramientas gráficas de alta calidad para administrar las

bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect).

3.2.8 Desventajas de PostgreSQL.

Se pueden enunciar algunas desventajas con respecto a PostgreSQL.

- Consume recursos y carga el sistema.
- El límite del tamaño de cada fila de las tablas es de 8kb (se puede ampliar a 32kb recompilando, pero con un costo añadido en el rendimiento).
- Es 2 veces más lenta que MySQL.
- Disminuyen las funciones en PHP.

En cuanto a consideraciones de estabilidad del servidor, PostgreSQL tiende a desperdiciar memoria y sobrecargar el sistema. Pero para sistemas más serios en las que la consistencia de la BD es fundamental (BD con información realmente importante, bancos, etc.). PostgreSQL es una mejor opción pese a su lentitud.

3.3 Características, ventajas y desventajas de PHP.

PHP (acrónimo de "PHP Hypertext Preprocessor") es un lenguaje *open source* interpretado de alto nivel insertado en páginas HTML y ejecutado en el servidor.

PHP fue creado como un lenguaje para Internet lo que le ha permitido adoptar constantemente las nuevas tecnologías disponibles para la red de redes a medida que surgen, y al mismo tiempo, conservar los aciertos de las alternativas precedentes, como Perl.

PHP se ejecuta en el servidor, lo que significa acceder a los recursos del mismo por ejemplo una base de datos. El programa PHP es ejecutado en el servidor y el

resultado enviado al navegador. El resultado es normalmente una página HTML⁶, pero igualmente podría ser una página WML⁷.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte (ver figura 3.3.1), es independiente del navegador, sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

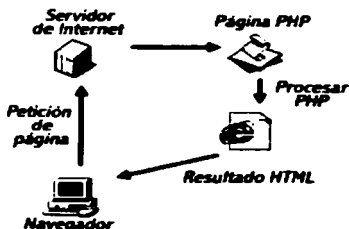


Fig. 3.3.1 Funcionamiento de PHP

PHP ha sido desarrollado principalmente para usarse conjuntamente con el servidor web libre Apache en cualquiera de las plataformas que éste soporta. De hecho, está implementado como un módulo de forma que aprovecha la máxima integración con el servidor y velocidad posibles (ver figura 3.3.2).


PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape y iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros

⁶ HyperText Markup Language. Lenguaje de programación para crear páginas web.

⁷ WML Wireless Markup Language. Lenguaje de programación equivalente para hacer páginas web soportadas por celulares.

Uno de los puntos principales de PHP es su eficiente API para construir aplicaciones de Bases de Datos. PHP no sólo puede usar muchos tipos de bases de datos de forma nativa, sino que permite interfaz ODBC y JDBC. PHP es muy utilizado sobre todo con PostgreSQL y MySQL, bases de datos libres, haciendo entre Apache, PHP y una de ambas la solución "libre" para crear sitios web dinámicos de alta calidad, sin dependencia tecnológica de ninguna empresa en particular. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. En la figura 3.3.3 podemos observar la pantalla de las características del sistema operativo donde se instaló PHP y el comando de configuración.

PHP Version 4.2.2



System	Linux deffy.perf.redhat.com 2.4.18-11smp #1 SMP Thu Aug 15 06:41:59 EDT 2002 i686 i686 i386 GNU/Linux
Build Date	Sep 3 2002 05:25:09
Configure Command	./configure '--host=i686-pc-linux-gnu' '--build=i686-pc-linux-gnu' '--target=i386-radhat-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/usr/com' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--prefix=/usr' '--with-config-file-path=/etc' '--enable-force-cgi-redirect' '--disable-debug' '--enable-pic' '--disable-rpath' '--enable-inline-optimization' '--with-bz2' '--with-db3' '--with-curl' '--with-dom=/usr' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--with-gd' '--enable-gd-native-ttf' '--with-ttf' '--with-xml' '--with-gettext' '--with-ncurses' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-png' '--with-pear' '--with-regex=system' '--with-xml' '--with-expat-dir=/usr' '--with-zlib' '--with-layout=GNU' '--enable-bcmath' '--enable-xml' '--enable-ftp' '--enable-magic-quotes' '--enable-safe-mode' '--enable-sockets' '--enable-sycteam' '--enable-sycteam' '--enable-sycteam' '--enable-track-vars' '--enable-trans-aid' '--enable-yp' '--enable-zlib' '--without-cc' '--with-pear=/usr/share/pear' '--with-imap=shared' '--with-imap-ssl' '--with-

Fig. 3.3.3 Pantalla de información de la versión e instalación de PHP

3.3.1 Características de PHP.

Las cuatro grandes características: **Velocidad, estabilidad, seguridad y simplicidad.**

- **Velocidad:**

No sólo posee velocidad en ejecución, sino además no crea demoras en la máquina. Por esta razón no requiere demasiados recursos de sistema. PHP se integra perfectamente a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.

- **Estabilidad:**

La velocidad no sirve de mucho si el sistema es vulnerable a fallas. Ninguna aplicación es 100% libre de bugs, sin embargo gracias al soporte que la comunidad de programadores ofrece es posible reducirlos. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

- **Seguridad:**

El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, éstos pueden ser configurados desde el archivo `php.ini`

- **Simplicidad:**

Permite generar código fuente productivo en menor tiempo. Ya que el contar con experiencia en lenguajes como C y C++ reduce la curva de aprendizaje para los programadores.

3.3.2 Ventajas de PHP.

Con PHP se puede hacer casi cualquier cosa como con scripts CGI, procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies. Quizá la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía web para una base de datos es una tarea relativamente simple con PHP. Las siguientes bases de datos están soportadas actualmente:

- Adabas D
- Ingres
- Oracle (OCI7 and OCI8)
- dBase
- InterBase
- PostgreSQL
- Empress
- FrontBase
- Solid
- FilePro
- mSQL
- Sybase
- IBM DB2
- MySQL
- Velocis
- Informix
- ODBC
- Unix dbm

PHP también tiene soporte para gran cantidad de estándares de red: LDAP, Corba, Calc, FTP, IMAP, POP3, NNTP, NIS, SMNP, HTTP y derivados.

También se pueden abrir sockets de red directos (**raw sockets**) e interactuar con otros protocolos.

- **Soporte XML:**

Considerado como el próximo estándar para Web XML (eXtensible Markup Language), es soportado directamente por PHP a través de un grupo de funciones especiales que puede ser usado para escribir completas aplicaciones basadas en XML.

- **Servicios de Directorio LDAP:**

El protocolo LDAP sirve para acceder y compartir enormes volúmenes de información organizados en directorios. Luego de montar un servidor LDAP libre como OpenLDAP, PHP incluye funciones especiales para acceder directamente a su contenido, permitiéndonos desarrollar aplicaciones Web listas para LDAP fácilmente.

- **Documentos PDF:**

El Portable Document Format creado por Adobe Systems se ha convertido en el formato estándar para la distribución de documentos electrónicos. Con PHP es posible generar documentos PDF fácilmente, gracias a su completo conjunto de funciones ClipPDF.

- **Expresiones Regulares:**

Quizá no tan populares, las funciones para manejar expresiones regulares en PHP son tan completas como las de Perl, permitiéndonos hacer complejas operaciones

con strings (o cadenas de caracteres), imprescindibles para escribir todo tipo de aplicaciones.

Entre otras cosas PHP tiene la capacidad de generar gráficos rápidamente, soporte para mensajería IMAP, programación orientada a objetos, cookies, y una excelente extensibilidad, que le permite a terceros desarrollar sus propias librerías.

3.3.3 Desventajas de PHP.

PHP es un lenguaje que se caracteriza más por sus ventajas que por sus desventajas, aunque se pueden señalar algunos inconvenientes como por ejemplo el manejo de errores que no es tan sofisticado como java, ASP o ColdFusion. No existe un IDE ni un depurador (debugger), aunque el depurador ya está por llegar.

PHP es un lenguaje orientado a objetos pero su mayor desventaja es que no tiene la misma portabilidad que Java. Lamentablemente PHP no ofrece la posibilidad de declarar las propiedades privadas, por lo que el programar en forma "encapsulada" se toma más como una filosofía de programación que una obligación.

Por otro lado cuando un objeto se crea de una clase que deriva de otra se ejecuta sólo el constructor de la misma clase, y no la del padre. Este es un defecto de PHP, ya que es clásico en los lenguajes orientados a objetos que exista lo que se llama encadenamiento de constructores.

En PHP no hay destructores y tampoco se pueden crear clases abstractas. En términos generales no es un verdadero lenguaje orientado a objetos, es un lenguaje híbrido donde se puede utilizar programación orientada a objetos, por otro lado PHP no soporta la persistencia de objetos. En programación orientada a objetos los objetos persistentes son objetos que mantienen su estado y funcionalidad a través de múltiples invocaciones de la aplicación.

Todas estas desventajas ya se han tomado en cuenta para la nueva versión de PHP la cual llevará el nombre de PHP5.

2.3.4 Características de PHP5.

Muy pocas veces se vio tanta expectativa ante el lanzamiento de una nueva versión de PHP. Las características que propician la programación orientada a objetos (POO) bajo PHP han sido revisadas y mejoradas, tomando en cuenta las necesidades que los desarrolladores clamaban como fundamentales. Objetos por referencia, clases abstractas, interfaces, variables y funciones privadas y protegidas, son sólo algunas de ellas, harán que PHP sea aún mucho más poderoso y flexible que antes.

3.4 Características, ventajas y desventajas del servidor de web Apache versión 1.3

El **Apache HTTP⁸ Server Project** (Proyecto de Servidor HTTP Apache, fig. 3.4.1) es un esfuerzo de la **Apache Software Foundation** (Fundación de Software Apache) encaminado a desarrollar y mantener un servidor de HTTP **open source** (de código fuente libre) para sistemas operativos modernos que incluyen UNIX y Windows NT. El objetivo de este proyecto es proporcionar un servidor robusto, comparable a los comerciales, seguro, extensible y eficiente que ofrezca servicios de HTTP en sincronía con los estándares HTTP actuales.⁹ El proyecto es gestionado conjuntamente por un grupo de voluntarios diseminados por todo el mundo, usando internet para comunicarse, planear y desarrollar el servidor y la documentación correspondiente. Estos voluntarios son conocidos como el **Apache Group** (Grupo Apache). Además, cientos de usuarios contribuyen al proyecto con ideas, código y documentación.



Fig. 3.4.1 Logotipo del proyecto de servidor HTTP Apache

⁸ HTTP - Hyper Text Transfer Protocol (Protocolo de transferencia de hipertexto)

⁹ <http://www.apache.org/>

3.4.1 Características de Apache.

El servidor de web Apache en su versión 1.3 tiene las siguientes características:¹⁰

- Bases de datos DBM para autenticación.- Que permiten configurar páginas protegidas con contraseñas con una enorme cantidad de usuarios autorizados, sin saturar el servidor.
- Respuestas personalizadas a errores y/o problemas.- Brinda la facilidad de configurar archivos, que pueden ser utilizados para que sean devueltos por el servidor en respuesta a errores y/o problemas. Por ejemplo, podría configurarse un pequeño programa para que intercepte los errores **500 Server** (Error 500 de servidor) y realizar algún tipo de diagnóstico sobre la marcha que informe al usuario y al administrador al mismo tiempo.
- Directivas para **DirectoryIndex** (índices de directorios) múltiples.- Ofrecen la posibilidad de nombrar de distintas formas a los archivos principales (index.html) con los que el servidor debe responder ante una petición de directorio de tipo URL¹¹. El archivo que responde puede ser un simple archivo index.html o ejecutar un programa de tipo index.cgi¹² dependiendo de lo que el servidor encuentre en el directorio en cuestión.
- Reescritura de URL y creación de alias flexible e ilimitada.- Apache no tiene un límite fijo en el número de alias y redireccionamientos que pueden ser declarados en el archivo de configuración del servidor. Además contiene un poderoso motor de reescritura que puede ser utilizado para resolver la mayoría de problemas relacionados con la manipulación de URLs.
- Negociación de contenido.- Es la habilidad de servir a clientes (navegadores) automáticamente variando el nivel de sofisticación y conformidad del HTML, con documentos que ofrecen la mejor

¹⁰ <http://httpd.apache.org/docs/misc/FAQ.html>

¹¹ **URL - Uniform Resource Locator** (Localizador de Recursos Uniforme)

¹² **CGI - Common Gateway Interface** (Interfaz de salida común)

representación de la información que el cliente (navegador) es capaz de aceptar.

- **Hosts** (Servidores locales) virtuales.- Esta es una característica que permite al servidor distinguir entre peticiones hechas a distintas direcciones IP o a distintos nombres (mapeados hacia el mismo equipo). Apache también ofrece hosting virtual masivo configurable dinámicamente.
- **Logs** (Archivos bitácora) configurables.- El servidor apache puede ser configurado para generar logs en el formato deseado. Además, en la mayoría de arquitecturas UNIX, Apache puede enviar dichos archivos a un pipe permitiendo, de esta forma, una rotación de logs, filtrado de hits (teclazos en página html), división en tiempo real de múltiples hosts virtuales a diferentes archivos log y resolución asíncrona de **DNS**¹³ sobre la marcha.
- Servidor **proxy** (apoderado o sustituto).- Las versiones posteriores a la 1.1 de Apache contienen un módulo proxy, que si es incluido en la compilación del núcleo, hará que Apache actúe como un servidor proxy.
- Soporte DSO¹⁴.- Es un ingenioso mecanismo llamado carga/enlace dinámico de objetos compartidos y dinámicos que proporciona un modo de construir un pedazo de código en un formato especial y que permite cargarlo en tiempo de ejecución en el espacio de direcciones de un programa ejecutable. Hasta la versión 1.3, el sistema de configuración de Apache tiene soporte opcional para sacar provecho del método DSO: Compilación del programa central de Apache produciendo una librería DSO para ser compartida y compilación de módulos de Apache en archivos de una librería DSO que serán cargados explícitamente durante la ejecución.

3.4.2 Ventajas de Apache.

- Es un servidor de web poderoso y flexible.

TESIS CON
FALLA DE ORIGEN

¹³ DNS - Domain Name Server (Servidor de Nombre de Dominio)

¹⁴ DSO - Dynamic Shared Object (Objetos Compartidos Dinámicos)

- Existe una encuesta¹⁵ realizada por Netcraft.com en agosto del 2002 que lo define como el servidor de Web más utilizado en Internet con un 63.98% de utilización sobre otros servidores de Web desde abril del año 1996.
- Implementa los últimos protocolos y cumple de conformidad la especificación HTTP/1.1 (RFC1626)¹⁶.
- Es altamente configurable y extensible con módulos de terceras partes.
- Puede ser personalizado al permitir la escritura de módulos utilizando el API¹⁷ de Apache.
- El código fuente completo se encuentra disponible gratuitamente y viene con una licencia sin restricciones.
- Se ejecuta sobre diversas plataformas: UNIX, Windows NT/9x, Netware 5.x y posteriores, Linux, etc.
- El producto se actualiza constantemente por miles de programadores en todo el mundo, lo que provoca la eliminación de errores y/o huecos de seguridad.
- Fomenta la retroalimentación de nuevas ideas, reporte de errores y envío de parches por parte de usuarios en general.
- Implementa muchas características nuevas en periodos de tiempo relativamente cortos.

3.4.3 Desventajas de Apache.

- Apache no incluye SSL¹⁸, ya que este transporte de información requiere de cifrado de datos y muchos gobiernos tienen restricciones sobre la importación, exportación y uso de tecnología de cifrado. Si Apache incluyera SSL en su paquete base, su distribución involucraría todo tipo de asuntos de carácter legal y burocrático, lo que imposibilitaría mantenerlo

¹⁵ http://news.netcraft.com/archives/web_server_survey.html

¹⁶ <http://www.rfc-editor.org/rfc/rfc2616.txt>

¹⁷ **API** – **Application Program Interface** (Interfaz de programa aplicativo)

¹⁸ **SSL** - **Secure Socket Layer** (Capa segura para la estructura de datos que representa un extremo de comunicación)

como software libre. Sin embargo, es posible implementar SSL sobre Apache, existen proyectos relacionados a este tema.

- Apache no incluye un motor de búsqueda, aunque éstos se le pueden agregar, hay motores de búsqueda que están disponibles de manera gratuita.
- Apache es únicamente un servidor de Web (HTTP), actualmente los servidores de aplicaciones son los que acaparan el mercado ya que se utilizan en infinidad de sistemas Web. Apache utilizado en combinación con un servidor de aplicaciones resulta una excelente alternativa; pero utilizado únicamente como servidor de Web limita su potencial al no tener el soporte de integración de bases de datos relacionales y motores de ejecución de otros lenguajes.
- Apache no incluye soporte para ASP¹⁹, por lo que este tipo de tecnología no puede ser utilizada. Sin embargo existen algunos productos en el mercado que permiten añadir esta funcionalidad.
- El paquete base del servidor Apache no incluye soporte para Java (JSP, EJB, servlets de Java²⁰). Estas características pueden agregarse como módulos separados del proyecto de Apache, concretamente dentro del proyecto Jakarta que es un servidor de aplicaciones gratuito.

3.5 Redes y Comunicaciones.

3.5.1 Redes.

Una red es un conjunto de computadoras (nodos) interconectados con algún hardware especial, capaces de comunicarse electrónicamente entre sí a través de un medio de comunicación digital.

Existen varias topologías de red para intercomunicar computadoras, entre ellas las más conocidas son :

¹⁹ ASP – Active Server Pages (Páginas de Servidor Activas)

²⁰ JSP – Java Server Pages (Páginas de Servidor de Java); EJB – Enterprise Java Beans (Componentes de java de propósito general)

- **Estrella.** Esta red intercomunica todas las computadoras en un hardware llamado concentrador(hub). En el concentrador es en donde se lleva a cabo toda la conmutación de información enviada por parte de los nodos de la red, es decir, el concentrador funciona como una oficina de correos, en donde se reciben los mensajes de todos los puntos de la red y se direccionan hacia el nodo correspondiente. En la figura 3.5.1.1 podemos apreciar la topología de estrella.

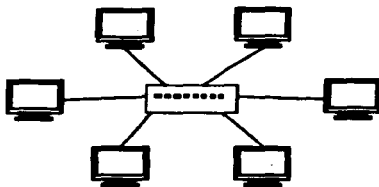


Figura 3.5.1.1 Topología de red estrella

- **Bus.** Esta red comparte un cable, generalmente coaxial, para transmitir toda la información entre los nodos que la conforman. En esta topología de red todos los nodos compiten por el medio de transmisión, por lo que es necesario implementar algoritmos que eviten las colisiones. El más popular de estos algoritmos es el CSMA/CD. En la figura 3.5.1.2 se puede observar la topología de bus.

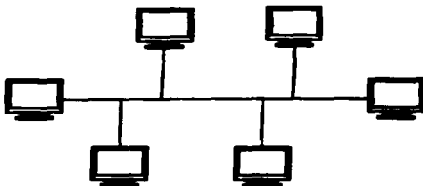


Figura 3.5.1.2 Topología de red bus.

El **CSMA/CD** funciona de la siguiente manera: cuando una computadora desea mandar información primero escucha el cable de la red para revisar que no se esté usando en ese preciso momento (**Carrier-Sense**). Esto se oye muy sencillo, pero el problema reside en que dos o mas computadoras al escuchar que no se está usando el cable pueden mandar en el exacto momento su información (**Multiple Access**), y como solamente puede haber uno y sólo un mensaje en transito, en el cable se produce una colisión. Entonces las computadoras detectan la colisión y deciden reenviar su información a un intervalo al azar, es importante que sea al azar ya que si ambas computadoras tuvieran el mismo intervalo fijo se produciría un ciclo vicioso de colisiones y reenvíos (**Collision Detection**).

- **Token Ring.** En esta red un *token* (Ficha Virtual) es pasado de computadora a computadora como si fuera una papa caliente. Cuando una computadora desea mandar información debe de esperar a que le llegue el *token* vacío, cuando le llega utiliza el *token* para transmitir a otra computadora, cuando la otra computadora recibe la información, regresa el *token* a la computadora que envió con el mensaje de que fue recibido el paquete. Así se libera el *token* para volver a ser usado por cualquier otra computadora. Debido a que una computadora requiere el *token* para enviar información no hay colisiones, el problema reside en el tiempo que debe esperar una computadora para obtener el *token* sin utilizar. En la figura 3.5.1.3 es posible apreciar la topología de *token-ring*.

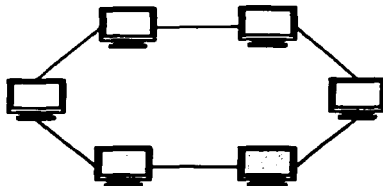


Figura 3.5.1.3 Topología de red token-ring

- **Malla (mesh).** La topología de malla utiliza conexiones redundantes entre los dispositivos de la red así como una estrategia de tolerancia a fallas. Cada dispositivo en la red está conectado a todos los demás (todos conectados con todos). Este tipo de tecnología requiere mucho cable (cuando se utiliza el cable como medio, pero puede ser inalámbrico también). Pero debido a la redundancia, la red puede seguir operando si una conexión se rompe.

Las redes de malla, obviamente, son más difíciles y caras para instalar que las otras topologías de red debido al gran número de conexiones requeridas. En la figura 3.5.1.4 podemos observar la topología de malla.

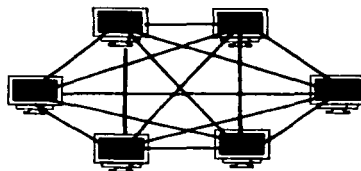


Figura 3.5.1.4 Topología de malla

3.5.2 Redes virtuales.

Es posible comunicar redes de diferentes topologías, logrando así tener redes de mayor tamaño, sin embargo, los mecanismos de comunicación entre las distintas topologías y el hardware tan diferente obligan a que la comunicación entre estas redes se lleve a cabo mediante circuitos virtuales.

Un circuito o red virtual se crea o establece cuando un nodo de una de las redes establece comunicación con otro nodo, que se encuentra en alguna otra red. Para ello, es necesario establecer un "camino" a través de las diferentes redes hasta llegar al destino.

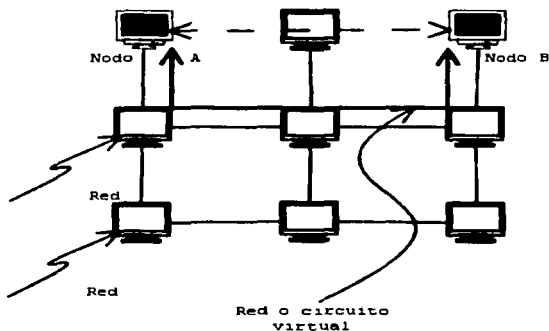


Figura 3.5.2.1 Ejemplo de una red o circuito virtual.

Como se puede observar en la figura 3.5.2.1 para que la comunicación entre los nodos A y B se lleve a cabo, se genera uno más paquetes con la información a enviar en el nodo origen (A). A continuación, el paquete es enviado al nodo de conexión con otra red, el cual se encargará de definir el camino necesario para llegar hasta el nodo destino (B) que se encuentra en una red distante. Entonces se comunica con los otros nodos de conexión a red de las demás redes para ubicar en cuál de ellas se encuentra el nodo destino y queda entonces establecido el "camino". A partir de ese momento, los paquetes que se envíen seguirán este circuito hasta que se termine la "conversación". Este procedimiento, como puede observarse, se lleva a cabo con equipos especiales de red, así como con algoritmos que realizan la búsqueda y establecimiento de los circuitos. Dado que estos equipos forman parte de los elementos de comunicaciones se detallan en la siguiente sección.

3.5.3 Comunicaciones.

Como se mencionó anteriormente, existen diversos equipos que tienen como función la conexión con otras redes. A continuación se enlistan algunos de ellos, con una breve descripción de su funcionamiento y su alcance.

- **Concentrador.** Este es el más sencillo de los elementos de comunicación de redes. Su función consiste en recibir todos los paquetes enviados por los distintos equipos que conforman una red y redireccionarlos. Estos equipos son los que se utilizan en las redes de topología de estrella y solamente cumplen con la función de comunicar los equipos dentro de la misma red. A los concentradores también se les conoce como repetidores activos, dado que repiten la señal que reciben, pero activamente porque solamente la repiten hacia el nodo correspondiente.
- **Puente (Bridge).** Un puente es un dispositivo (lógico o físico) que interconecta dos redes cuya topología, además, puede ser diferente. Sin embargo, no es capaz de definir lo que hay del otro lado de la red, por lo que no puede distinguir en qué parte de la red se encuentra el equipo destino.
- **Switch digital.** En la actualidad para construir redes de área amplia (WAN) se busca utilizar una infraestructura de conmutación de paquetes (switches), con tecnologías como X.25, Frame Relay y actualmente con ATM (Asynchronous Transfer Mode) de conmutación de celdas.

En términos de conmutación todos los ruteadores funcionan como equipos de acceso a una red conmutada y cada uno de los sitios se conectan al proveedor de servicio y de esta manera hacen uso de la infraestructura de telecomunicaciones para transmitir otro tipo de información. Actualmente hay varios tipos de conmutación que tienen diferentes grados de control de la red.

- **Ruteador.** Es un elemento de red que sirve para conectar dos o más redes separadas lógicamente (subredes).

Los ruteadores se utilizan cuando se debe mantener separada la información de distintas subredes, pero al mismo tiempo tener enlaces ocasionales entre ellas; cada subred tiene asignada una dirección de red diferente.

Los ruteadores realizan una función muy similar a la de un puente. Típicamente, los ruteadores tienen un procesamiento más intenso que los puentes. Como resultado, sus velocidades de procesamiento no son tan altas. Por otro lado, los ruteadores tienen una capacidad mucho más sofisticada de selección de rutas óptimas de transmisión.

- **Gateway (pasarela).** Es un dispositivo que conecta 2 redes de tipo diferente, el cual realiza la conversión de protocolos de una red a otra, también puede utilizarse para lograr acceso a diversas sesiones en un *mainframe*.

3.5.4 La familia de protocolos TCP/IP.

La Internet es un conjunto de redes interconectadas que intercambian información mediante una familia de protocolos conocidos como "protocolos TCP/IP"²¹. Dentro de los protocolos más comunes están:

ARP (Address Resolution Protocol), DNS (Domain Name Service), ICMP (Internet Control Messaging Protocol), RPC (Remote Procedure Call), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), SNMP (Simple Network Management Protocol), NFS (Network File System), HTTP (Hyper Text Transfer Protocol), NNTP (Network News Type Protocol), Gopher, Archie, NTP (Network

²¹ TCP/IP (Transfer Control Protocol / Internet Protocol, Protocolo de Control de Transferencias / Protocolo Internet) Son solamente 2 de los protocolos que utiliza la Internet. Debido a su popularidad y la cantidad de protocolos que están basados en ellos, a los protocolos de Internet se les conoce como de la familia de TCP/IP.

Time Protocol), HTTPS (Hyper Text Transfer Protocol-Secure), SFTP (Secure File Transfer Protocol), TFTP (Trivial File Transfer Protocol), SSH (Secure Shell), Telnet, etc.

En este trabajo se utilizan los siguientes protocolos:

- HTTP. Creado para transferir páginas de hipertexto, mejor conocidas como páginas Web. La forma en la que se utiliza este protocolo es con un visualizador de páginas web (web browser) que establece una conexión con un servidor del protocolo, realiza la solicitud y recibe y despliega el archivo.
- HTTPS. Funciona de manera similar al protocolo base http, pero envía y recibe información encriptada utilizando el conjunto de funciones SSL(Secure Socket Layer).
- FTP. Este protocolo nos sirve para tomar y colocar archivos en un servidor. Para ello, el protocolo establece una comunicación con el servidor y a través de comandos especializados, copia los archivos a través de la red.
- SSH. Establece una conexión segura con un servidor, el cual brinda un intérprete de comandos. Con este protocolo podemos interactuar en tiempo real con el servidor al cual estemos accediendo.

TESIS CON
FALLA DE ORIGEN

3.6 Plataforma Linux.

En la actualidad el sistema operativo Linux se ha convertido en una importante opción para las empresas, organizaciones, universidades, instituciones de gobierno como la SEP en el caso de México porque brinda seguridad en los sistemas de las mismas, es un sistema operativo de distribución gratuita, multiusuario, multitarea, multiproceso que hacen que los sistemas que trabajen bajo él, puedan operar con estabilidad. Podemos ver que con la nueva ola de virus informativos y gusanos que se propagan por Internet y que han causado grandes daños financieros a grandes compañías las empresas están optando por un sistema operativo estable, robusto y confiable para sus sistemas, aplicaciones con la finalidad de tener disponible sus servicios en todo momento. Es por estas características que decimos contar con Linux como sistema operativo para desarrollar nuestro trabajo de tesis.

3.6.1 Características de Linux.



Fig. 3.6. 1 Logotipo de Linux

Linux es un sistema operativo que cuenta con las siguientes características:

- **Multitarea:** varios programas ejecutándose al mismo tiempo.
- **Multiusuario:** varios usuarios pueden acceder al mismo tiempo el mismo software desde distintos equipos.
- **Multiplataforma:** Se ejecuta en distintos tipos de CPU's, no sólo Intel.
- **Funciona en modo protegido 386.**
- **Tiene protección de la memoria entre procesos,** de manera que uno de ellos no pueda colgar el sistema.
- **Carga de ejecutables por demanda:** Linux sólo lee de disco aquellas partes de un programa que están siendo usadas actualmente.

- Política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha (se sigue denominando intercambio, es en realidad un intercambio de páginas). Un total de 16 zonas de intercambio de 128Mb de tamaño máximo pueden ser usadas en un momento dado con un límite teórico de 2Gb para intercambio.
- La memoria se gestiona como un recurso unificado para los programas de usuario, de tal forma que toda la memoria libre puede ser usada para caché y éste puede a su vez ser reducido cuando se ejecuten grandes programas.
- Bibliotecas compartidas de carga dinámica (DLL's) y bibliotecas estáticas también.
- Se realizan volcados de estado (core dumps) para posibilitar los análisis post-mortem, permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.
- Casi totalmente compatible con POSIX, System V y BSD a nivel fuente.
- Mediante un módulo de emulación de iBCS2, casi completamente compatible con SCO, SVR3 y SVR4 a nivel binario.
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los *drivers*, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- Control de tareas POSIX.
- Pseudo-terminales (pty's).

TESIS CON
FALLA DE ORIGEN

- Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Por supuesto, si el ordenador ya tiene una FPU (unidad de punto flotante), será usada en lugar de la emulación, pudiendo incluso compilar su propio kernel sin la emulación matemática y conseguir un pequeño ahorro de memoria.
- Soporte para muchos teclados nacionales o adaptados y es bastante fácil añadir nuevos dinámicamente.
- Consolas virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video). Se crean dinámicamente y se puede tener hasta 64.
- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V, y tiene un avanzado sistema de archivos propio con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud.
- Acceso transparente a particiones MS-DOS (o a particiones OS/2 FAT) mediante un sistema de archivos especial: no se requiere ningún comando especial para usar la partición MS-DOS, parece un sistema de archivos común de Unix (excepto por algunas restricciones en los nombres de archivo, permisos, y esas cosas). Las particiones comprimidas de MS-DOS 6 no son accesibles en este momento, y no se espera que lo sean en el futuro. El soporte para VFAT (WNT, Windows 95) ha sido añadido al núcleo de desarrollo y estará en la próxima versión estable.
- Un sistema de archivos especial llamado UMSDOS permite que Linux sea instalado en un sistema de archivos DOS.
- Soporte en sólo lectura de HPFS-2 del OS/2 2.1
- Sistema de archivos de CD-ROM que lee todos los formatos estándar de CD-ROM.
- TCP/IP, incluyendo ftp, telnet, NFS, etc.
- Appletalk disponible en el actual núcleo de desarrollo.

- Software cliente y servidor Netware disponible en los núcleos de desarrollo.

3.7 Seguridad.

Linux pertenece a la clase de software que podemos denominar *software libre*. Esto tiene un significado determinante en cuanto a su desarrollo:

- Ha de ser distribuido con el código fuente, lo cual supone que los fuentes estarán al alcance siempre de quien quiera ojearlos.
- Puede ser modificado libremente, siendo distribuido después bajo su misma licencia (generalmente GPL), de modo que cualquiera puede tomar el código de los programas y modificarlos a placer para que cumplan unas funciones específicas o para solventar fallos.

Estos dos hechos son esenciales a la hora de plantearnos la seguridad en un producto de software libre. Las consideraciones son muy dependientes del estilo de seguridad que nos interese implantar en nuestros sistemas, sin embargo hay unos puntos clave que debemos tener en cuenta:

- Se tiene acceso al código fuente: El acceso al código fuente en el software libre (y por tanto también en Linux) es total, de modo que cualquiera puede intentar buscar vulnerabilidades mediante una auditoría de código, sin embargo del mismo modo que cualquiera puede buscar fallos para usarlos de manera inversa, cualquiera puede realizar auditorías de código para mejorar la seguridad de un producto de software libre que esté usando.

Que el código fuente sea accesible, propicia que la seguridad no sea llevada a cabo mediante una política de *security through obscurity* ("seguridad a través de la oscuridad") y la mejora de ésta depende de que todo el mundo tenga los fallos a la vista y que la seguridad no pase por "esconder" los fallos a la vista de ojos mal intencionados.

TESIS CON
FALLA DE ORIGEN

- Se puede modificar libremente: El hecho de que el código libre pueda ser modificado libremente facilita de una manera muy notable el hecho de que si aparece cualquier fallo, el parche para dicho fallo aparecerá en un periodo muy corto de tiempo. Esto es debido a que mucha gente está usando ese producto y es posible que muchos de ellos sean capaces de solucionar ese fallo y posteriormente poner un parche público. En algunos casos de productos comerciales los parches o las soluciones para vulnerabilidades de seguridad tardan mucho tiempo o aparecen juntas en paquetes de actualizaciones mucho después de que hayan aparecido los fallos.

Linux proporciona los niveles exigibles de seguridad física para un sistema operativo:

- Un arranque seguro
- Posibilidad de bloquear las terminales
- Capacidad de un sistema multiusuario real.

3.7.1 Seguridad en el arranque.

Cuando alguien inicia el sistema operativo Linux se encuentra con una pantalla de *login*: el sistema está pidiendo que se identifique. Si es un usuario conocido, podrá iniciar una sesión y trabajar con el sistema. Si no lo es, no tendrá opción de hacer absolutamente nada. Además, el sistema registra todos los intentos de acceso (fallidos o no), por lo que no pasarán desapercibidos intentos repetidos de acceso no autorizado.

LILO (Linux Loader) es el encargado de cargar el sistema operativo en memoria y pasarle información para su inicio. A su vez, se le puede pasar parámetros a LILO para modificar su comportamiento.

Por ejemplo, si alguien en el indicador de LILO añade *init single*, el sistema se inicia en modo monousuario y proporciona un shell de root sin contraseña. Si en

su entorno de trabajo cree necesario evitar que alguien pueda iniciar el sistema de esta forma, deberá utilizar el parámetro *restricted* en el fichero de configuración de LILO (habitualmente */etc/lilo.conf*). Este parámetro le permite iniciar normalmente el sistema, salvo en el caso de que se hayan incluido argumentos en la llamada a LILO, que solicita una clave. Esto proporciona un nivel de seguridad razonable: permite iniciar el sistema, pero no manipular el arranque. Si se tienen mayores necesidades de seguridad puede incluirse la opción *password*. De esta forma se necesitará una clave para iniciar el sistema. En estas condiciones, sólo podrá iniciar el sistema quien conozca la clave.

3.7.2 Bloqueo de la consola.

En los entornos Unix es conocido el truco de ejecutar en una terminal que alguien ha dejado inocentemente abierta un guión que simule la pantalla de presentación al sistema, entonces un usuario incauto introducirá su nombre y clave, que quedarán a merced del autor del engaño.

Si es necesario que un usuario deje su computadora, existen dos programas que bloquean la consola con la finalidad de evitar que vean su información. Dichos programas son *xlock* y *vlock*.

xlock bloquea la pantalla cuando nos encontramos en modo gráfico. Está incluido en la mayoría de las distribuciones Linux que soportan X. En general se puede ejecutar *xlock* desde cualquier *xterm* de su consola y bloqueará la pantalla de forma que necesitará su clave para desbloquearla.

Vlock es un programa que permite cerrar alguna o todas las consolas virtuales de Linux. Se puede bloquear sólo aquella en la que se está trabajando o todas. Si sólo se cierra una, las otras se pueden abrir y utilizar la consola, pero no se podrá usar *vty* hasta que no la desbloquee.

3.7.3 Cuentas de usuario, grupos.

Cada usuario del sistema está definido por una línea en el archivo `/etc/passwd` y cada grupo por otra línea en el fichero `/etc/group`. Cada usuario pertenece a uno o varios grupos y cada recurso pertenece a un usuario y un grupo. Los permisos para un recurso se pueden asignar al propietario, al grupo y a otros (resto de los usuarios). Para mantener un sistema seguro y funcional, tendremos que realizar las combinaciones necesarias entre el propietario y grupo de un recurso con los permisos de los propietarios, grupos y otros. Por ejemplo, la unidad de disco flexible tiene las siguientes características:

```
brw-rw-r-- 1 root floppy 2,0 may 5 1998 /dev/fd0
```

- Propietario: root con permiso de lectura y escritura.
- Grupo: floppy con permiso de lectura y escritura.
- Otros: resto de los usuario con permiso de lectura.

Cuando deseemos que un usuario escriba en la unidad de disco, sólo tendremos que incluirlo en el grupo *floppy*. Cualquier otro usuario que no pertenezca al grupo *floppy* (salvo *root*) sólo podrá leer el disquete.

El administrador tiene que conocer las necesidades de cada uno de sus usuarios y asignarle los mínimos privilegios para que pueda realizar su trabajo.

3.7.4 Seguridad de las claves.

La seguridad de una sola cuenta puede comprometer la seguridad de todo el sistema. Esto es una realidad ante la que debemos protegernos.

Por un lado tenemos que asegurarnos de que nuestros usuarios utilizan claves sólidas:

- No deben ser una palabra conocida.
- Deberían contener letras, números y caracteres especiales.
- Deben ser fáciles de recordar y difíciles de adivinar.

Para comprobar que este requisito se verifica en nuestro sistema, podemos usar los mismos mecanismos que utilizan los atacantes. Existen varios programas que van probando varias palabras de diccionario, claves habituales y combinaciones de caracteres, que son cifrados con el mismo algoritmo que usa el sistema para mantener sus claves; después son comparadas con el valor de la clave cifrada que queremos averiguar hasta que el valor obtenido de un cifrado coincide con una clave cifrada. Posteriormente notificaremos al usuario que su clave es débil y le solicitaremos que la modifique.

Por otro lado, los passwords cifrados se almacenan en el archivo `/etc/passwd`. Cualquier usuario del sistema tiene permiso de lectura sobre este archivo. Hoyos de seguridad en los navegadores permiten que se puedan leer archivos arbitrarios de una máquina (evidentemente, que el usuario de navegador tenga permiso para leer), de manera que lleguen hasta un hacker que cree páginas web que exploten estos agujeros. Para solucionar vulnerabilidades podemos recurrir a contraseñas en la sombra (*shadow passwords*), un mecanismo consistente en extraer las claves cifradas del archivo `/etc/passwd` y situarlas en otro archivo llamado `/etc/shadow`, que sólo puede leer el usuario `root` y dejar el resto de la información en el original `/etc/passwd`. El archivo `/etc/shadow` sólo contiene el nombre de usuario y su clave, e información administrativa, como cuándo expira la cuenta, etc. El formato del archivo `/etc/shadow` es similar al siguiente:

usuario : clave : último : puede : debe : aviso : expira : desactiva : reservado

- **usuario:** El nombre del usuario.
- **clave:** La clave cifrada
- **último:** Días transcurridos del último cambio de clave desde el día 1/1/70
- **puede:** Días transcurridos antes de que la clave se pueda modificar.
- **tiene:** Días transcurridos antes de que la clave tenga que ser modificada.

TESIS CON
FALLA DE ORIGEN

- **aviso:** Días de aviso al usuario antes de que expire la clave.
- **expira:** Días que se desactiva la cuenta tras expirar la clave.
- **desactiva:** Días de duración de la cuenta desde el 1/1/70.
- **reservado:** sin comentarios.

Esto quiere decir que cuando se ejecute el proceso correspondiente va a tener los privilegios del propietario del comando (es decir, el *root*), no del usuario que lo lanzó. En otras palabras, el proceso generado por *passwd* pertenece a *root*. A primera vista, esto puede parecer una seria brecha de seguridad, y lo es, si el programa funciona correctamente, no tiene por qué dar problemas; pero pequeños defectos en el programa pueden ser utilizados por alguien para tratar de ejecutar otro código distinto con los privilegios de este proceso.

Se debe tener en cuenta que algunos programas (como *passwd*) tienen que tener el bit SUID. Se debe comprobar en los lugares habituales, que ninguno de los programas propiedad del *root* o SUID que utiliza en su sistema tiene un fallo de seguridad conocido que pueda ser explotado.

3.7.5 Seguridad del root.

Una primera norma que siempre se deberá tener presente es usar la cuenta de *root* sólo para realizar tareas concretas y breves, el resto hacerlo como usuario *normal*.

- No se debe usar la cuenta de *root* por norma. Hay que evitarlo. Se debe intentar primero cualquier acción como un usuario normal, y ver que si no se tiene el permiso requerido, usar el comando "su" si es necesario.
- Se deben ejecutar los comandos de forma segura verificando previamente la acción que se va a realizar. Por ejemplo si se quiere ejecutar *rm borrar.**, se debe ejecutar primero *ls borrar.** y si es lo que se pretende, hay que modificar el mandato y ejecutarlo.

- Ciertos mandatos admiten una opción (-i) para actuar de forma interactiva. Debe activarse, si no lo está ya, añadiendo estas líneas al archivo de recursos para el *shell*:
 - alias rm='rm -i'
 - alias cp='cp -i'
 - alias mv='mv -i'
- El directorio actual no está, por omisión, en la ruta de búsqueda de ejecutables (PATH). Esto garantiza que no lanzaremos, sin darnos cuenta un ejecutable que esté en el directorio actual llamado, por ejemplo *ls*.
- Se debe evitar que la clave del *root* viaje por una red sin cifrar. Se debe utilizar *ssh* u otro canal seguro.
- Hay que limitar las terminales desde las que se puede conectar *root*. Es preferible limitarlo a la consola del sistema. Esto se puede decidir en */etc/security*. Si se necesita una sesión remota como *root*, hay que entrar como usuario normal y luego usar *su*.

3.7.6 El árbol de directorios.

A continuación se tienen algunas consideraciones para la seguridad del árbol de directorios.

- Si el sistema va a dar servicio a múltiples usuarios que requieren almacenamiento para datos privados, sería conveniente que el directorio */home* tuviera su propia partición.
- Si el equipo va a ser un servidor de correo, impresión, etc., el directorio */var* o incluso */var/spool* podrían tener su propia partición.
- Algunos directorios son necesarios en la partición raíz. Contienen datos que son necesarios durante el proceso de arranque del sistema. Son */dev*, */etc*, */bin*, */sbin*, */lib*, */boot*.
- El directorio */usr/local* contiene los programas compilados e instalados por el administrador. Resulta conveniente usar una partición propia para proteger

estos programas personalizados de futuras actualizaciones del sistema. Este criterio también se puede aplicar al directorio `/opt`.

3.7.7 Permisos.

Linux, como sistema multiusuario, asigna un propietario y un grupo a cada fichero (y directorio) y unos permisos al propietario, al grupo y al resto de los usuarios. La forma más rápida de comprobar esta característica es usar el comando `ls -la`. Así nos aparece el tipo de archivo, el propietario, el grupo, los permisos e información adicional. Por supuesto, el sistema de archivos tiene que admitir esta característica, como es el caso del sistema de archivos `ext2` (Linux nativo). En los sistemas de archivos pensados para entornos monousuario, como `msdos` o `vfat`, no tenemos esta característica, por lo que son inseguros y su uso no es aconsejable bajo Linux.

3.7.8 Permisos de archivos y directorios.

En general, cualquier sistema UNIX divide el control de acceso a archivos y directorios en tres elementos: propietario, grupo y otros. Tanto el propietario como el grupo son únicos para cada archivo o directorio. Eso sí, a un grupo pueden pertenecer múltiples usuarios. Otros hace referencia a los usuarios que ni son el propietario ni pertenecen al grupo.

3.7.9 Limitar el espacio asignado a los usuarios.

Lo primero que tendríamos que hacer es asegurarnos de que nuestro núcleo tiene soporte para las cuotas de usuarios.

```
# dmesg | grep quotas
VFS: Diskquotas version dquot_6.4.0 initialized
```

En caso contrario, el núcleo no ha sido compilado con soporte para el sistema de cuotas para usuarios. Así que será necesario compilar un nuevo núcleo Linux.

Es necesario editar el fichero `/etc/fstab` y añadir `usrquota` o `grpquota` en la partición o particiones en las que nos interese limitar el espacio de almacenamiento.

El sistema de cuotas de Linux permite limitar el número de bloques y el número de i-nodos que un usuario puede tener. Los valores a modificar son los límites que están puestos entre paréntesis (que ahora valen 0). Se puede especificar cualquier cantidad (en Kbytes).

3.7.10 Seguridad en la red, compilando el núcleo.

Linux tiene la gran ventaja de tener disponible el código fuente del núcleo; en realidad Linux propiamente dicho es sólo el núcleo. Esto nos permite la posibilidad de crear núcleos a medida de nuestras necesidades. Y parte de nuestras necesidades será la mejora de la seguridad.

Para compilar el núcleo primero tendremos que configurar las opciones que nos interesen. Los fuentes del núcleo se guardan habitualmente en el directorio `/usr/src/linux`, y una vez situados en él, tendremos que ejecutar `«make menuconfig»` (o `«make xconfig»` si estamos en modo gráfico). Así aparecen todas las opciones de configuración. Dentro de ellas hay que enfocarse en las que están relacionadas con la seguridad, viendo una breve explicación de lo que hacen y cómo se usan.

Como el núcleo controla las características de red del sistema, es importante que el núcleo tenga las opciones que garanticen la seguridad y que el propio núcleo no pueda ser comprometido. El mecanismo de filtrado de conexiones se realiza en el núcleo del sistema operativo y si ha sido compilado con estas opciones. Normalmente lo está. Este filtrado se realiza a nivel de red y transporte: cuando llega un paquete por un interfaz de red se analiza siguiendo los filtros de entrada. Este paquete puede ser aceptado, denegado o rechazado, en este último caso se avisa al remitente. Si los filtros de entrada aceptan el paquete pasa al sistema si era su destino final o pasa por los filtros de reenvío o enmascaramiento, donde se vuelve a repetir una de las acciones. Por último, los paquetes que proceden del

propio sistema o los que han sido aceptados por los filtros de reenvío o enmascaramiento pasan al filtro de salida.

3.7.11 Comunicaciones seguras.

Por último, nos interesará mantener unas comunicaciones seguras para garantizar la privacidad e integridad de la información. Actualmente existen las herramientas necesarias para cada necesidad.

Podemos usar cifrados simétricos como pgp y gpg para documentos, correo electrónico y comunicaciones sobre canales inseguros

Podemos crear canales de comunicación seguros de distinto tipo:

- SSH (Secure Shell), stelnnet: SSH y stelnnet son programas que le permiten efectuar conexiones con sistemas remotos y mantener una conexión cifrada. Con esto evitamos, entre otras cosas, que las claves circulen por la red sin cifrar.
- Cryptographic IP Encapsulation (CIPE): CIPE cifra los datos a nivel de red. El viaje de los paquetes entre hosts se hace cifrado. A diferencia de SSH que cifra los datos por conexión, lo hace a nivel de socket. Así un conexión lógica entre programas que se ejecutan en hosts diferentes está cifrada. CIPE se puede usar en tunnelling para crear una Red Virtual Privada. El cifrado a bajo nivel tiene la ventaja de poder hacer trabajar la red de forma transparente entre las dos redes conectadas en la RVP sin ningún cambio en el software de aplicación.
- SSL: o Secure Sockets Layer, fue diseñado y propuesto en 1994 por Netscape Communications Corporation junto con su primera versión del Navigator como un protocolo para dotar de seguridad a las sesiones de navegación a través de Internet. Sin embargo, no fue hasta su tercera versión, conocida como SSL v3.0 que alcanzó su madurez, superando los problemas de seguridad y las limitaciones de sus predecesores. En su estado actual proporciona los siguientes servicios:

- Cifrado de datos: la información transferida, aunque caiga en manos de un atacante, será indescifrable, garantizando así la confidencialidad.
- Autenticación de servidores: el usuario puede asegurarse de la identidad del servidor al que se conecta y al que posiblemente envíe información personal confidencial.
- Integridad de mensajes: se impide que modificaciones intencionadas o accidentales en la información mientras viaja por Internet pasen inadvertidas.
- Opcionalmente, autenticación de cliente: permite al servidor conocer la identidad del usuario, con el fin de decidir si puede acceder a ciertas áreas protegidas.

3.7.12 Firewalls.

Los *firewalls* son una de las medidas más utilizadas para el control de las conexiones a nuestras redes. Linux desde su versión 2.0 incorpora en el kernel métodos de filtrado de paquetes que han ido evolucionando en las distintas versiones, a cada una de las cuales se han ido añadiendo nuevas características. Estos firewalls nos permitirán filtrar conexiones de red por direcciones ip, tipo de paquetes, puertos y un largo etcétera

En el kernel 2.0, aparece la utilidad *ipfwadm*, posteriormente sustituida por *ipchains* en 2.2 e *iptables* en la actual versión del núcleo 2.4. Esta última versión incorpora sofisticados mecanismos de seguimiento de conexiones, filtrado de paquetes y una modularidad impresionante que permite a quien lo desee incorporar nuevos módulos de políticas de filtrado de paquetes.

TESIS CON
FALLA DE ORIGEN

3.8 Metodologías utilizadas.

3.8.1 Bases de datos relacionales (back end)

Para la base de datos se utilizó el modelado relacional que permite mantener por sus características, la integridad de datos, independencia lógica, independencia física y flexibilidad necesaria para tener un sistema estable.

3.8.2 RUP - Rational Unified Process (front end).

Captura varias de las mejores prácticas en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones.

Es una guía de cómo utilizar de manera efectiva UML (Unified Modeling Language).

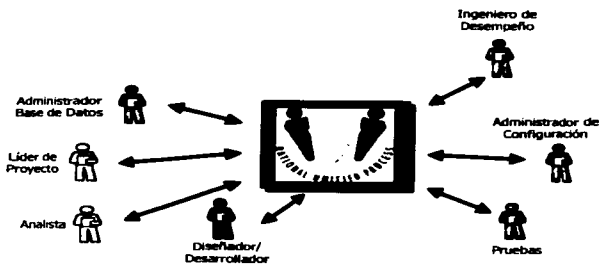
Provee a cada miembro de un equipo un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo.

Crea y mantiene modelos, en lugar de enfocarse en la producción de una gran cantidad de papeles de documentación.

3.8.2.1 Incremento de la productividad en equipo.

Todos los miembros del equipo comparten
una base de conocimiento,
un Proceso,
una vista de cómo desarrollar software,
un lenguaje de modelado (UML).

TESIS CON
FALLA DE ORIGEN



3.8.2.2 Seis mejores prácticas. Figura 2.8.2.1.1 Miembros del equipo.

RUP describe cómo utilizar de forma efectiva procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de software, conocidos como "mejores prácticas".

- Administración de requerimientos
- Desarrollo iterativo.
- Modelo visual.
- Verificación de la calidad.
- Arquitecturas con componentes.
- Control de cambios.

TESIS CON
FALLA DE ORIGEN

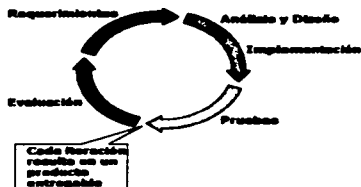


Figura 3.8.2.2.1 Administración de requerimientos.

3.8.2.3 Desarrollo iterativo de software.

Dados los sistemas de software sofisticados de la actualidad, no es posible hacer de manera secuencial la definición completa del problema, diseñar la solución completa, construir el software y por último probarlo.

El descubrimiento de defectos en fases posteriores de diseño dan como resultado un aumento en los costos y/o la cancelación del proyecto.

El tiempo y dinero gastados en la implementación de un diseño fallido, no son recuperables.

3.8.2.4 Características del desarrollo iterativo.

- Permite un entendimiento incremental del problema a través de refinamientos sucesivos.
- Habilita una fácil retroalimentación de usuario.
- Metas específicas permiten que el equipo de desarrollo mantenga su atención en producir resultados.
- El progreso es medido conforme avanzan las implementaciones.

3.8.2.5 Administración de requerimientos.

- Licitar, organizar y documentar la funcionalidad y restricciones requeridas.
- Llevar un registro y documentación de cambios y decisiones.
- Los requerimientos de negocio son fácilmente capturados y comunicados a través de casos de uso.
- Los casos de uso son instrumentos importantes de planeación.

Los casos de uso dirigen el trabajo desde el análisis hasta las pruebas



Figura 3.8.2.5.1 Modelos utilizados para el desarrollo del Sistema de Administración de Recursos Humanos.

3.8.2.6 Arquitectura basada en componentes.

Se enfoca en el pronto desarrollo de una arquitectura ejecutable robusta.

- Resistente al cambio mediante el uso de interfaces bien definidas.
- Intuitivamente comprensible.
- Promueve un reuso de software más efectivo.
- Es derivada a partir de los casos de uso más importantes.

3.8.2.7 Modelo visual del software.

- Captura la estructura y comportamiento de arquitecturas y componentes.
- Muestra como encajan de forma conjunta los elementos del sistema.
- Mantiene la consistencia entre un diseño y su implementación.
- Promueve una comunicación sin ambigüedades.

3.8.2.8 Verificación de la calidad del software.

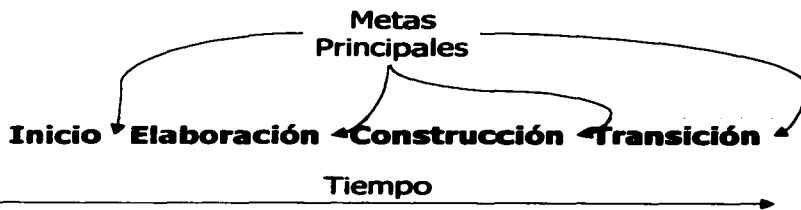
- Crea pruebas para cada escenario (casos de uso) para asegurar que todos los requerimientos están propiamente implementados.
- Verifica la calidad del software con respecto a los requerimientos basados en la confiabilidad, funcionalidad, desempeño de la aplicación y del sistema.
- Prueba cada iteración
- Los problemas del software son de 100 a 1000 veces más costosos de encontrar y reparar después del desarrollo

3.8.2.9 Fases en RUP.

Inicio – Define el alcance del proyecto.

Elaboración – Plan del proyecto, especificación de características, arquitectura base.

Construcción – Construir el producto.



Transición – Transición del producto a la comunidad del usuario

Figura 3.8.2.9.1 Fases de RUP

Fase de inicio

- Propósito

TESIS CON
FALLA DE ORIGEN

Establecer casos de negocios para un nuevo sistema o para alguna actualización importante de un sistema existente. Especificar el alcance del proyecto.

- **Resultado**

Una visión general de los requerimientos del proyecto, i.e., los requerimientos principales.

Un modelo inicial de casos de uso y modelo del dominio (10-20%).

Un caso de negocios inicial, incluyendo:

Evaluación inicial de riesgos.

Una estimación de los recursos requeridos.

Fase de elaboración

- **Propósito**

Analizar el dominio del problema.

Establecer una buena arquitectura.

Lidiar con los elementos de riesgo más altos del proyecto.

Desarrollar un plan comprensivo mostrando cómo el proyecto será completado.

Resultado

- Un modelo del dominio y de casos de uso 80% completo.
- Requerimientos suplementarios que capturen los requerimientos no funcionales y cualesquiera requerimientos que no estén asociados con un caso de uso específico.
- Una lista de riesgos revisada.

Fase de construcción

- **Propósito**

Desarrollar producto de software completo el cual estará listo para ser transferido al usuario.

- **Productos**

TESIS CON
FALLA DE ORIGEN

- Un modelo completo de diseño y casos de uso.
- Liberaciones de productos ejecutables de funcionalidad incremental.
- Documentación de usuario.
- Una liberación "beta" del producto.

Fase de transición

Hacer la transición final del producto de software al usuario

- Productos

Liberaciones ejecutables de producto.

"Pruebas beta" para validar el nuevo sistema vs. las expectativas del usuario.

Manuales de usuario actualizados.

Documentación de desarrollo actualizada.

¿Está el usuario satisfecho?

Gastos reales de los recursos vs. gastos previstos ¿Aceptables?

Iteraciones

Cada fase en RUP puede descomponerse en iteraciones. Una iteración es un ciclo de desarrollo completo dando como resultado una entrega de producto ejecutable (interna o externa).

Liberaciones

Inicio	Elaboración		Construcción			Transición	
Iteración Preliminar	Iteración de Arquitectura	Iteración de Arquitectura	Iteración de Desarrollo	Iteración de Desarrollo	Iteración de Desarrollo	Iteración de Transición	Iteración de Transición

Figura 3.9.2.9.2 Iteraciones en el proceso RUP

CAPÍTULO IV. DESARROLLO E IMPLANTACIÓN

*"Genius is one percent inspiration and
ninety-nine percent perspiration (el genio
es uno por ciento de inspiración y noventa
y nueve por ciento transpiración)".*

-- Thomas Alva Edison

4.1 Aplicación de la metodología elegida.

Para el diseño de la base de datos (Back-End), se utilizó el modelado relacional que permite mantener por sus características, la integridad de datos, independencia lógica, independencia física y flexibilidad necesarias para tener un sistema estable. Por convención y experiencia adquirida, esta metodología se caracteriza por ser una solución unificada, de alto nivel y con un sólido fundamento teórico al problema del almacenamiento, recuperación y mantenimiento de datos.

4.1.1 Diagrama de contexto.

La figura 4.1.1.1 resume las entradas y salidas que definen el Sistema de Administración de Recursos Humanos.

Las entidades Usuario, Administrador y Operador realizan diversas actividades que están contenidas en la entrada como solicitud operaciones y con sus correspondientes salidas que representan las respuestas a los procesos realizados en el Sistema.

Por otra parte, las entidades Empresa Capacitadora y Empresa ingresan datos relevantes al Sistema obteniendo como respuesta las confirmaciones de esas entradas.

TESIS CON
FALLA DE ORIGEN

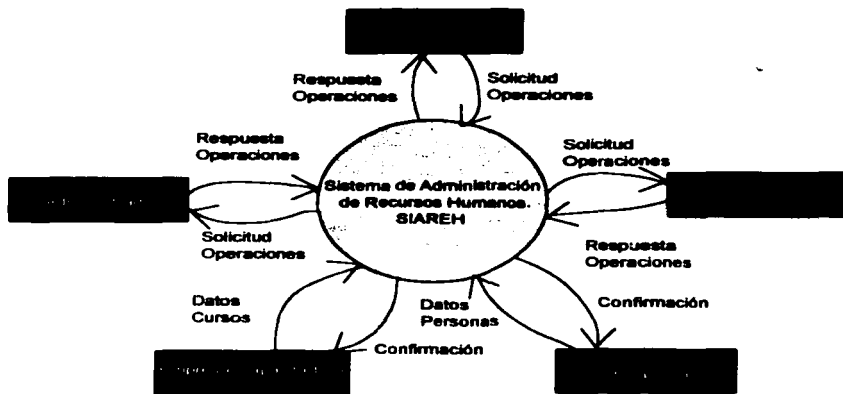


Diagrama 4.1.1.1. Diagrama de Contexto de SIAREH

4.1.2 Diagrama de flujo de procesos.

Definición:

El diagrama de flujo de procesos representa el camino o flujo que deben seguir los datos o procesos que cambian o transforman los datos en un sistema. Las entidades externas representan los límites del sistema, pueden ser la fuente y/o el destino de los datos. Los datos fluyen a través de otros elementos que funcionan como depósitos de datos, los cuales conservan los mismos para su tratamiento.

La figura 4.1.2.1 representa el diagrama de flujo de procesos de nivel 1.

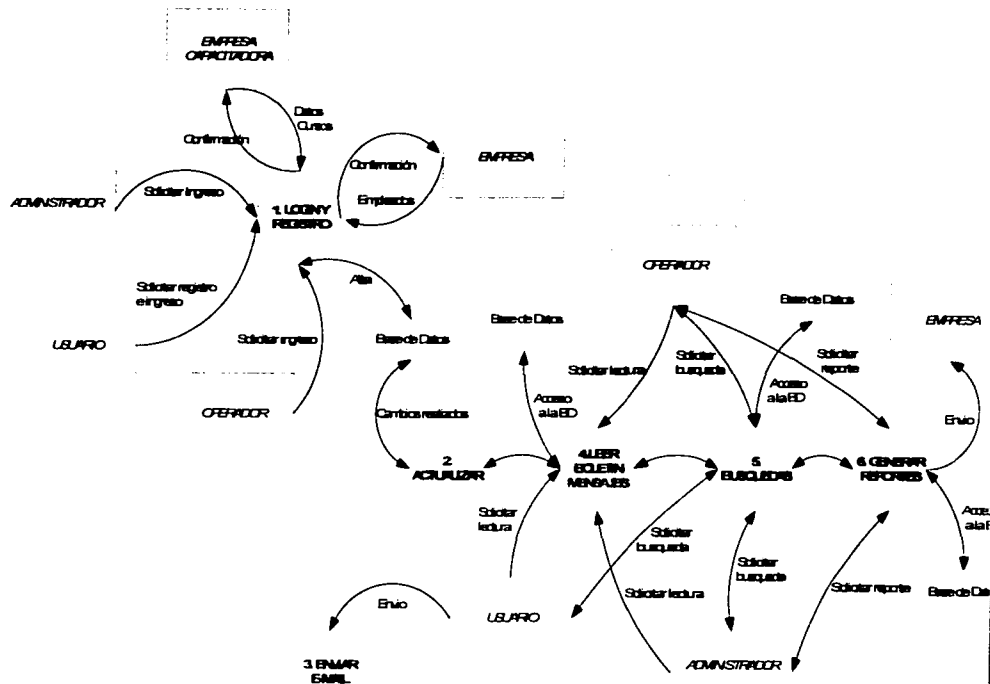


Figura 4.1.2.1 Diagrama de flujo de procesos Nivel 1

TESIS CON FALLA DE ORIGEN

4.1.3 Diccionario de datos.

Es una lista de los nombres de los datos y su definición, tipo de variable, longitud, rango, etc. La tabla 4.1.3.1 muestra el diccionario de datos del sistema.

Acronimo	Tipo y Longitud	Llave	Tabla	Descripción
ID_Boletin	Numeric(4,0)	SI	Boletin	PK de la tabla Boletin.
Descripcion	Varchar(255)	NO	Boletin	Descripción del
Texto	Varchar(100)	NO	Boletin	Catalogo
ID_Colaborador	Numeric(4,0)	SI	Capacitación_Interna	Contenido del Boletin
ID_Programación	Numeric(4,0)	SI	Capacitación_Interna	FK de la tabla
Evaluación	Varchar(255)	NO	Capacitación_Interna	Capacitación Interna
ID_Categoría	Numeric(4,0)	SI	Categoría	FK de la tabla
Descripción	Varchar(255)	NO	Categoría	Capacitación Interna
ID_Cliente	Numeric(4,0)	SI	Cliente	Resultado de la
Descripción	Varchar(255)	NO	Cliente	evaluación
ID_Colaborador	Numeric(4,0)	SI	Colaborador(Persona)	PK de la tabla
ID_Estado_Civil	Numeric(4,0)	SI	Colaborador(Persona)	Colaborador
ID_Nacionalidad	Numeric(4,0)	SI	Colaborador(Persona)	FK de la tabla
ID_Categoría	Numeric(4,0)	SI	Colaborador(Persona)	Colaborador
ID_Estado	Numeric(4,0)	SI	Colaborador(Persona)	FK de la tabla
ID_Boletin	Numeric(4,0)	SI	Colaborador(Persona)	Colaborador
ID_Nivel	Numeric(4,0)	SI	Colaborador(Persona)	FK de la tabla
ID_Experiencia_Laboral	Numeric(4,0)	SI	Colaborador(Persona)	Colaborador
Nombre	Varchar(25)	NO	Colaborador(Persona)	FK de la tabla
Ap_Paterno	Varchar(25)	NO	Colaborador(Persona)	Colaborador
Ap_Materno	Varchar(25)	NO	Colaborador(Persona)	Nombre de el
Dirección	Varchar(155)	NO	Colaborador(Persona)	Colaborador
Teléfono	Varchar(20)	NO	Colaborador(Persona)	Apellido Paterno del
				Colaborador
				Apellido Materno del
				Colaborador
				Dirección del
				Colaborador
				Teléfono del
				colaborador

Email	Varchar(50)	NO	Colaborador(Persona)	Email del colaborador
RFC	Varchar(15)	NO	Colaborador(Persona)	RFC del colaborador
Fecha	Datetime	NO	Colaborador(Persona)	Fecha de Nacimiento del Colaborador
Sexo	Varchar(18)	NO	Colaborador(Persona)	Sexo del Colaborador
Status	Varchar(18)	NO	Colaborador(Persona)	Es ¿colaborador o candidato?
Login	Varchar(18)	NO	Colaborador(Persona)	Login de acceso del colaborador
Password	Varchar(18)	NO	Colaborador(Persona)	Password del colaborador
Sueldo	Money	NO	Colaborador(Persona)	Sueldo del Colaborador
Tarifa_Por_Hora	Money	NO	Colaborador(Persona)	Tarifa por hora
Años	Numeric(2)	NO	Colaborador(Persona)	Años en los que termino sus estudios
Nombre_Estudios	Varchar(50)	NO	Colaborador(Persona)	Carrera profesional que estudió
Generación	Varchar(20)	NO	Colaborador(Persona)	Generación a la que pertenece
Diplomado	Varchar(255)	NO	Colaborador(Persona)	Estudios de Diplomado
Indicador_Bloqueado	Boolean	NO	Colaborador(Persona)	Indicador de bloqueo. ¿Usuario bloqueado?
Proyectos Anteriores	Varchar(255)	NO	Colaborador(Persona)	Indica en que proyectos ha estado antes
Experiencia Laboral	Numeric(3)	NO	Colaborador(Persona)	La experiencia laboral en años
ID_Colaborador	Numeric(4,0)	SI	Candidato Habilidad	FK de la tabla candidato habilidad
ID_Habilidad_Tecnica	Numeric(4,0)	SI	Candidato Habilidad	FK de la tabla candidato habilidad
ID_Colaborador	Numeric(4,0)	SI	Persona Idioma	FK de la tabla persona idioma
ID_Idioma	Numeric(4,0)	SI	Persona Idioma	FK de la tabla persona idioma
ID_Colaborador_Linea_Puesto	Numeric(4,0)	SI	Colaborador_Linea_Puesto	FK de la tabla colaborador línea puesto
ID_Colaborador	Numeric(4,0)	SI	Colaborador_Linea_Puesto	FK de la tabla colaborador línea puesto
ID_Puesto	Numeric(4,0)	SI	Colaborador_Linea_Puesto	FK de la tabla colaborador línea puesto
ID_Linea_De_Servicio	Numeric(4,0)	SI	Colaborador_Linea_Puesto	FK de la tabla colaborador línea puesto
ID_Experiencia_Laboral	Numeric(4,0)	SI	Experiencia Laboral	PK de la tabla Experiencia Laboral
Telefono	Numeric(25)	NO	Experiencia Laboral	Teléfono de la empresa anterior
Salario	Money	NO	Experiencia Laboral	Salario en la empresa anterior
Puesto_Jefe_Inmediato	Varchar(100)	NO	Experiencia Laboral	Puesto del jefe

**TESIS CON
FALLA DE ORIGEN**

Puesto_Desempeñado	Varchar(100)	NO	Experiencia Laboral	inmediato en la empresa
Nombre_Jefe_Inmediato	Varchar(25)	NO	Experiencia Laboral	Puesto desempeñado en la empresa
Motivo_Separacion	Varchar(255)	NO	Experiencia Laboral	Nombre del jefe inmediato
Fecha_Separacion	Datetime	NO	Experiencia Laboral	Motivos de separación de la empresa
Fecha_Ingreso	Datetime	NO	Experiencia Laboral	Fecha de la separación
Nombre_Empresa	Varchar(150)	NO	Experiencia Laboral	Fecha de ingreso en la empresa
Dirección_Empresa	Varchar(255)	NO	Experiencia Laboral	Nombre de la empresa anterior
Ap_Paterno_Jefe_Inmediato	Varchar(25)	NO	Experiencia Laboral	Dirección de la empresa anterior
Ap_Materno_Jefe_Inmediato	Varchar(25)	NO	Experiencia Laboral	Apellido paterno del jefe inmediato
Antigüedad	Numeric(3)	NO	Experiencia Laboral	Apellido materno del jefe inmediato
ID_Curso	Numeric(4,0)	SI	Curso	Tiempo que se laboro en la empresa
ID_Empresa_Proveedor	Numeric(4,0)	SI	Curso	PK de la tabla curso
_Capacitacion	Varchar(100)	NO	Curso	FK de la tabla curso
Nombre	Varchar(100)	NO	Curso	Nombre del curso
Duracion	Varchar(100)	NO	Curso	Duración en horas del curso
Vigencia	Varchar(100)	NO	Curso	Vigencia del Curso
Perfil	Varchar(100)	NO	Curso	Perfil de la persona para tomar el curso
ID_Empresa_Proveedor	Numeric(4,0)	SI	Empresa_Proveedor_Capacitación	PK de la tabla Empresa_Proveedor_Capacitación
_Capacitacion	Varchar(100)	NO	Empresa_Proveedor_Capacitación	Nombre de la Empresa capacitadora
Nombre	Varchar(100)	NO	Empresa_Proveedor_Capacitación	Dirección de la Empresa capacitadora
Dirección	Varchar(100)	NO	Empresa_Proveedor_Capacitación	Persona a cargo de la Empresa capacitadora
Representante	Varchar(100)	NO	Empresa_Proveedor_Capacitación	Teléfono de la Empresa capacitadora
Telefono	Varchar(45)	NO	Empresa_Proveedor_Capacitación	PK de la tabla (catalogo) Estado
ID_Estado	Numeric(4,0)	SI	Estado	PK de la tabla (catalogo) País
ID_Pais	Numeric(4,0)	SI	País	Descripción del catálogo
Descripcion	Varchar(255)	NO	País	PK de la tabla (catalogo) Estado Civil
ID_Estado_Civil	Numeric(4,0)	SI	Estado_Civil	Descripción del catálogo
Descripcion	Varchar(255)	NO	Estado_Civil	PK de la tabla (catalogo) Nivel Técnico
ID_Habilidad_Tecnica	Numeric(4,0)	SI	Habilidad Técnica	Habilidad Técnica
Nivel_Tecnico	Varchar(100)	NO	Habilidad Técnica	Nivel Técnico del Candidato
ID_Idioma	Numeric(4,0)	SI	Idioma	PK de la tabla Idioma

Descripcion	Varchar(18)	NO	Idioma	Descripcion , posibles opciones
Porcentaje_Hablado	Varchar(18)	NO	Idioma	Porcentaje de idioma hablado
Porcentaje_Escrito	Varchar(18)	NO	Idioma	Porcentaje de idioma escrito
Porcentaje_Lectura	Varchar(18)	NO	Idioma	Porcentaje de idioma leído
ID_Institucion	Numeric(4,0)	SI	Institución	PK de la tabla institución
ID_Colaborador	Numeric(4,0)	SI	Institución	FK de la tabla institución
Descripcion	Varchar(255)	NO	Institución	Descripcion de la institución donde se curso estudios.
Nombre	Varchar(60)	NO	Institución	Nombre institución.
ID_Linea_De_Servicio	Numeric(4,0)	SI	Linea de Servicio	PK de tabla linea de servicio
ID_Cliente	Numeric(4,0)	SI	Linea de Servicio	(Departamento).
Descripcion	Varchar(255)	NO	Linea de Servicio	FK de la tabla Linea de Servicio.
ID_Mensaje	Numeric(4,0)	SI	Mensaje	Descripcion del Departamento
Valor_Actual	Varchar(18)	NO	Mensaje	PK de la tabla Mensaje
Valor_Anterior	Varchar(18)	NO	Mensaje	Valor actual del mensaje
Fecha_Solicitud	Datetime	NO	Mensaje	Valor anterior del mensaje
Tipo_Opcion	Varchar(18)	NO	Mensaje	Fecha en que fue solicitado
Decision	Varchar(60)	NO	Mensaje	Opción de tipo de mensaje
SQL	Varchar(255)	NO	Mensaje	Decision
ID_Mensajes	Numeric(4,0)	SI	Mensaje_Persona	Consulta SQL
ID_Colaborador	Numeric(4,0)	SI	Mensaje_Persona	FK de la tabla transitiva
ID_Nacionalidad	Numeric(4,0)	SI	Nacionalidad	FK de la tabla transitiva
Descripcion	Varchar(255)	NO	Nacionalidad	PK de la tabla Nacionalidad (catalogo)
ID_Nivel	Numeric(4,0)	SI	Nivel	Descripcion del catálogo
Descripcion	Varchar(255)	NO	Nivel	PK de la tabla Nivel (catalogo)
ID_Pais	Numeric(4,0)	SI	Pais	Descripcion del catalogo
Descripcion	Varchar(255)	NO	Pais	FK de la tabla Pais (catalogo)
ID_PreRequisitos	Numeric(4,0)	SI	Prerrequisitos	Descripcion del catálogo
ID_Habilidad_Tecnica	Numeric(4,0)	SI	Prerrequisitos	PK de la tabla Pre requisitos
				FK de la tabla Pre requisitos

ID_Curso	Numeric(4,0)	SI	Prerrequisitos	FK de la tabla Pre requisitos
ID_Programacion	Numeric(4,0)	SI	Programación	PK de la tabla Programación
ID_Curso	Numeric(4,0)	SI	Programación	FK de la tabla Programación
Fecha_Inicio	Datetime	NO	Programación	Fecha de inicio de curso
Fecha_Termino	Datetime	NO	Programación	Fecha de termino
Sede	Varchar(100)	NO	Programación	Lugar donde se impartirá el curso
Horario	Varchar(100)	NO	Programación	Horario en el que se imparte el curso
Instructor	Varchar(100)	NO	Programación	Instructor a cargo del curso
Precio	Money	NO	Programación	Costo del curso
ID_Proyecto	Numeric(4,0)	SI	Proyecto	PK de la tabla Proyecto
Nombre	Varchar(255)	NO	Proyecto	Nombre del Proyecto
Descripcion	Varchar(255)	NO	Proyecto	Descripción del mismo, de que se trata.
Ubicacion	Varchar(255)	NO	Proyecto	Lugar en donde se realizara el proyecto.
Cliente	Varchar(100)	NO	Proyecto	Cliente
Costo_Estimado	Money	NO	Proyecto	Costo estimado del proyecto
ID_Proyecto	Numeric(4,0)	SI	Proyecto_Linea_Servicio	FK de Proyecto_Linea_Servicio.
ID_Linea_De_Servicio	Numeric(4,0)	SI	Proyecto_Linea_Servicio	FK de Proyecto_Linea_Servicio.
ID_Puesto	Numeric(4,0)	SI	Puesto	PK de la tabla Puesto
Descripcion	Varchar(255)	NO	Puesto	Descripción del puesto
ID_Rol_Colaborador	Numeric(4,0)	SI	Rol_Colaborador	PK de la tabla Rol_Colaborador
ID_Rol	Numeric(4,0)	SI	Rol_Colaborador	FK de la tabla Rol_Colaborador
ID_Proyecto	Numeric(4,0)	SI	Rol_Colaborador	FK de la tabla Rol_Colaborador
ID_Colaborador	Numeric(4,0)	SI	Rol_Colaborador	FK de la tabla Rol_Colaborador
ID_Rol	Numeric(4,0)	SI	Rol_Perfil	PK de la tabla Rol
Descripcion	Varchar(255)	NO	Rol_Perfil	Descripción.

Tabla 4.1.3.1 Diccionario de Datos

4.1.4 Diagrama Entidad - Relación.

El Diagrama Entidad Relación (DER), es un diagrama que muestra de manera abstracta el modelado de la distribución de los datos que se van a almacenar para un sistema. La figura 4.1.4.1 muestra el Diagrama Entidad Relación del Sistema.

TESIS CON
 FALLA DE ORIGEN

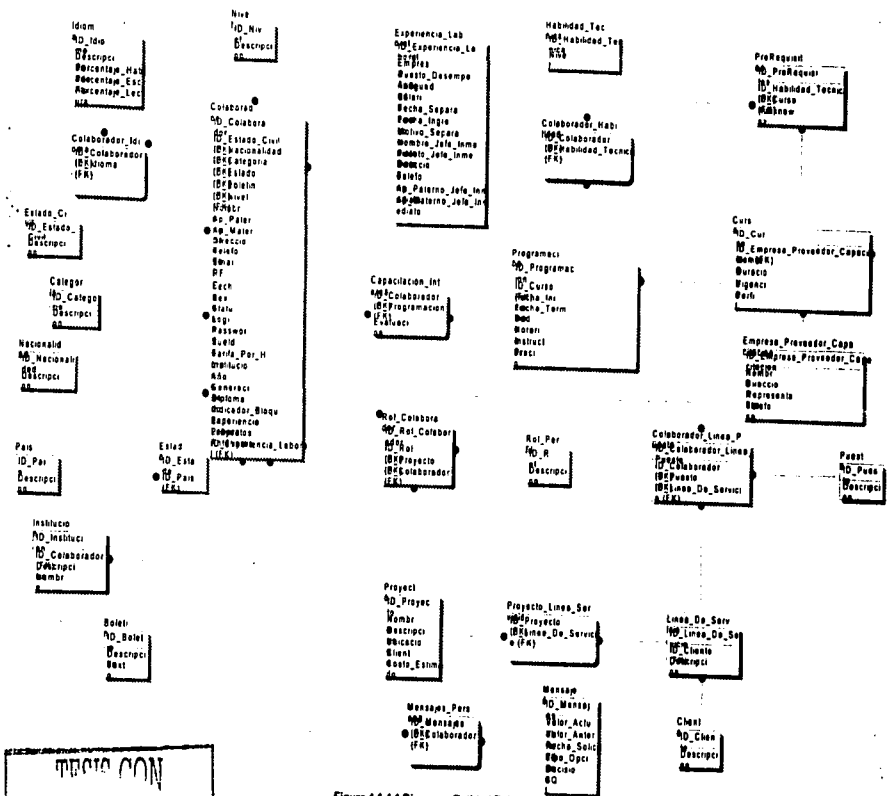


Figura 4.1.4 Diagrama Entidad Relación

TRABAJO CON
FALLA DE ORIGEN

4.1.5 Normalización.

La normalización es un proceso estándar que consiste en realizar ciertos pasos para evitar redundancia, inconsistencias, pérdidas en los datos y la imposibilidad de realizar inserciones en la base de datos.

Tomando en cuenta la tabla 4.1.5.1 se ejemplificará el proceso de normalización.

<i>ID_Persona</i>	<i>Nombre</i>	<i>Carrera</i>
1	Alberto Rodríguez	Ing. en computación
2	Teresita de Jesús	Mercadotecnia, Secretariado
3	Yoakin Martínez	Psicología, Economía

Tabla 4.1.5.1

Primera Forma Normal (1FN)

Se dice que una tabla se encuentra en primera forma normal (1NF) si y solo si cada uno de los campos contiene un único valor para un registro determinado.

<i>ID_Persona</i>	<i>Nombre</i>
1	Rodríguez
2	Teresita de Jesús
3	Yoakin

Tabla 4.1.5.2

<i>ID_Persona</i>	<i>Carrera</i>
1	Ing. en computación
2	Mercadotecnia
2	Secretariado
3	Psicología
3	Economía

Tabla 4.1.5.3

Las tablas 4.1.5.2 y 4.1.5.3 representan la primera forma normal que indica que en la tabla se deben tener valores atómicos (únicos) en los campos.

Segunda Forma Normal (2FN)

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

Tomando en cuenta la tabla 4.1.5.4 demostraremos la segunda forma normal.

<i>ID_Persona</i>	<i>ID_Linea de Servicio</i>	<i>Nombre</i>	<i>Linea de Servicio</i>	<i>Años</i>
1	6	Rodríguez	Desarrollo	4
2	3	Teresita de Jesús	Ventas	2
3	5	Yoakin	Servicios Médicos	9
4	3	Iván	Ventas	3
2	6	Teresita de Jesús	Desarrollo	5

Tabla 4.1.5.4

<i>ID_Persona</i>	<i>Nombre</i>
1	Rodríguez
2	Teresita de Jesús
3	Yoakin
4	Iván

Tabla 4.1.5.5

<i>ID_Linea de Servicio</i>	<i>Linea de Servicio</i>
3	Ventas
5	Servicios Médicos
6	Desarrollo

Tabla 4.1.5.6

<i>ID_Persona</i>	<i>ID_Linea de Servicio</i>	<i>Años</i>
1	6	4
2	3	2
3	5	9
4	3	3
2	6	5

Tabla 4.1.5.7

Las tablas 4.1.5.5, 4.1.5.6 y 4.1.5.7 ejemplifican la Segunda forma normal en donde todos los campos dependen directamente de la llave.

 TESIS CON
FALLA DE ORIGEN

Tercera Forma Normal (3FN)

Se dice que una tabla está en tercera forma normal si y sólo si los campos de la tabla dependen únicamente de la clave, dicho en otras palabras los campos de las tablas no dependen unos de otros.

Para la tercera forma normal tomamos la tabla 4.1.5.8

<i>ID_Persona</i>	<i>Nombre</i>	<i>Carrera</i>	<i>Institución</i>
1	Rodríguez	Ing. en Computación	UNAM
2	Teresita de Jesús	Mercadotecnia	ENEP-Aragón
3	Yoakin	Psicología	UAM

Tabla 4.1.5.8

<i>ID_Persona</i>	<i>Nombre</i>	<i>Carrera</i>
1	Rodríguez	Ing. en Computación
2	Teresita de Jesús	Mercadotecnia
3	Yoakin	Psicología

Tabla 4.1.5.9

<i>Carrera</i>	<i>Institución</i>
Ing. en Computación	UNAM
Mercadotecnia	ENEP-Aragón
Psicología	UAM

Tabla 4.1.5.10

De esta manera se puede observar que las tablas 4.1.5.9 y 4.1.5.10 se encuentran en tercera forma normal, ya que los campos de las tablas dependen únicamente de la llave por tanto no dependen unos de otros.

En los temas siguientes, se establecen conceptos que se utilizan para la metodología Orientada a Objetos, se hará uso del **RUP** (Rational Unified Process - Proceso Unificado de Rational -) para la construcción del Front-end.

4.1.6 Modelo de requisitos.

El modelo de requisitos tiene como objetivo delimitar el sistema y capturar la funcionalidad que debe ofrecer desde la perspectiva del usuario. Este modelo puede funcionar como un contrato entre el desarrollador y el cliente o usuario del sistema, y por lo tanto proyecta lo que el cliente desea según la percepción del desarrollador.

El modelo de requisitos es el primer modelo a desarrollarse, sirviendo de base para la formación de todos los demás modelos en el desarrollo de software. En general, cualquier cambio en la funcionalidad del sistema es más fácil de hacer, y con menores consecuencias a este nivel, que posteriormente. El modelo de requisitos que desarrollaremos se basa en la metodología Objectory (Jacobson et al. 1992), basada principalmente en el modelo de casos de uso. Este modelo de casos de uso y el propio modelo de requisitos son la base para los demás modelos.

- **Requisitos:** El modelo de casos de uso sirve para expresar el modelo de requisitos, el cual se desarrolla en cooperación con otros modelos como se verá más adelante.
- **Análisis:** La funcionalidad especificada por el modelo de casos de uso se estructura en el modelo de análisis, que es estable con respecto a cambios, siendo un modelo lógico independiente del ambiente de implementación.
- **Diseño:** La funcionalidad de los casos de uso ya estructurada por el análisis es realizada por el modelo de diseño, adaptándose al ambiente de implementación real y refinándose aún más.
- **Implantación:** Los casos de uso son implantados mediante el código fuente en el modelo de implantación.
- **Pruebas:** Los casos de uso son comprobados a través de las pruebas de componentes y pruebas de integración.

TESIS CON
FALLA DE ORIGEN

- **Documentación:** El modelo de casos de uso debe ser documentado a lo largo de las diversas actividades, dando lugar a distintos documentos como son los manuales de usuario, manuales de administración, etc.

Modelo de casos de uso.

El modelo de casos de uso describe un sistema en término de sus distintas formas de utilización, cada uno de estas formas es conocida como un caso de uso. Cada caso de uso o flujo se compone de una secuencia de eventos iniciada por el usuario. Para ello se define el concepto de actor, correspondiente al tipo de usuario que está involucrado en la utilización de un sistema, siendo el actor una entidad externa al propio sistema. Juntos, el actor y el caso de uso representan los dos elementos básicos de este modelo lo cual se muestran de manera gráfica en la **Figura 4.1.1.3.1** de acuerdo a la notación UML.



Figura 4.1.1.3.1 Gráfico de actor y caso de uso

Actores.

Los actores son entidades distintas a los usuarios, en el sentido que los usuarios son las personas reales que utilizan el sistema, mientras que los actores representan un cierto papel que una persona real puede jugar. Utilizando terminología orientada a objetos, se considera al actor como una clase de usuario, mientras que los usuarios se consideran como objetos o instancias de esa clase. Incluso, una misma persona puede aparecer como diferentes instancias de diferentes actores.

4.1.6.1 Diagrama de estados.

Son los que muestran los cambios de estado en los objetos.

Diagrama de estados para usuario Administrador

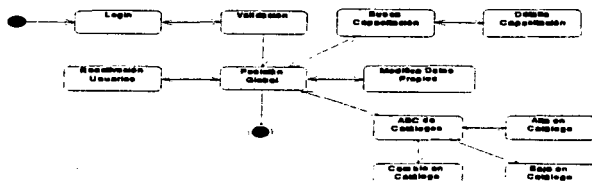


Figura 4.1.6.1.1 Diagrama de estado del administrador

Diagrama de estados de operador

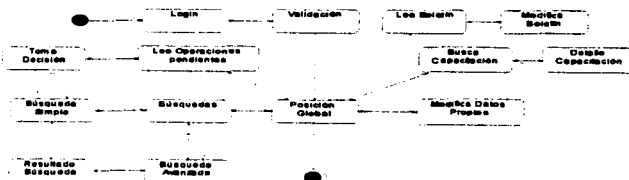


Figura 4.1.6.1.2 Diagrama de estado del operador

TESIS CON
FALLA DE ORIGEN

Extensión.

Un concepto importante que se utiliza para estructurar y relacionar casos de uso es la extensión. La extensión especifica cómo un caso de uso puede insertarse en otro para extender la funcionalidad del anterior. El caso de uso donde se va a insertar la nueva funcionalidad debe ser un flujo completo, por lo cual éste es completamente independiente del caso de uso a ser insertado. De esta manera, el caso de uso inicial no requiere consideraciones adicionales en relación a los casos de uso que puedan ser insertados, evitando complejidad adicional.

Inclusión.

Una relación adicional entre casos de uso es el de inclusión la cual apoya la reutilización de los casos de uso.

Las asociaciones extensión e inclusión entre casos de uso son asociaciones de clases por lo tanto se puede escoger entre ellas. De hecho, en la mayoría de los casos, la selección es bastante obvia y no causa problemas. Un criterio importante es ver qué tanto se acoplan las funcionalidades de los casos de uso. Si el caso de uso a ser extendido es útil por sí mismo, la relación debe ser descrita utilizando extensión. Si los casos de uso son fuertemente acoplados, la inserción debe tomar lugar para obtener un curso completo, la relación debe ser descrita utilizando inclusión.

En el sistema de Administración de recursos humanos se utilizan los actores y ya identificados como punto de partida. Dado que el Usuario es el actor primario se comienza con él. El sistema tiene que dar ciertos servicios al usuario, como consultas e ingreso de datos propios. Podemos definir nuestros casos de uso principales, nótese que los nombres de los casos de uso deben corresponder a acciones y no tanto a sustantivos.

TESIS CON
FALLA DE ORIGEN

4.1.6.2.1 Tablas de los casos de uso.

Documentación.

Parte fundamental del modelo de casos de uso es una descripción textual detallada de cada uno de los casos de uso identificados. Estos documentos son sumamente críticos ya que a partir de estos se desarrollará el sistema completo.

Las descripciones de los casos de uso representan todas las posibles interacciones de los actores con el sistema únicamente en base a eventos enviados o recibidos por los actores. En esta etapa no se incluyen eventos internos al propio sistema ya que esto será tratado durante el análisis y únicamente agregaría complejidad innecesaria en esta etapa. El formato del documento que utilizaremos consiste en una primera columna con los aspectos a describirse y una segunda columna con las descripciones correspondientes, a continuación se muestra en las tablas 4.1.6.2.1.1, 4.1.6.2.1.2, 4.1.6.2.1.3 y 4.1.6.2.1.4 la descripción de los casos de uso para el Sistema de Administración de Recursos Humanos.

TESIS CON
FALLA DE ORIGEN

Caso de uso:	Decisión del operador
Actores:	Operador, Base de Datos
Propósito:	Definir si las modificaciones realizadas por los usuarios se ejecutan o se cancelan
Resumen:	El operador define si las modificaciones realizadas por los usuarios se llevan a cabo se cancelan a través de un formulario
Precondiciones:	Opción del menú principal.
Flujo principal:	Del Front-End hacia el Back-End
PostCondiciones:	N/A
Subflujos:	N/A
Excepciones:	E1: Error al acceder a la Base de Datos E2: Error al ejecutar la consulta E3: Error al recuperar datos E4: Al no tener información válida en el formulario

Tabla 4.1.6.2.1.1 Descripción del caso de uso Decisión del operador del Sistema de Administración de Recursos Humanos.

Caso de uso:	Búsqueda avanzada
Actores:	Base de datos, Operador
Propósito:	Realizar una búsqueda en base a un determinado conjunto de datos de entrada
Resumen:	Este caso de uso es iniciado por el front end cuando el usuario envía un formulario y realiza una búsqueda en la base de datos
Precondiciones:	Opción del Menú principal
Flujo Principal:	De front-end hacia la base de datos. Este proceso involucra algunos elementos del back end.
Subflujos:	N/A
PostCondiciones:	N/A
Excepciones:	E1: No poder crear formularios. E2: No rellenar la información correctamente. E3: No poder leer la base de datos. E4: Si no existen registros de la búsqueda se le informa al actor Front-End

Tabla 4.1.6.2.1.2 Descripción del caso de uso Búsqueda avanzada del Sistema de Administración de Recursos Humanos.

TESIS CON
FALLA DE ORIGEN

Caso de uso:	Generación de Reportes
Actores:	Operador y Base de Datos
Propósito:	Realizar una consulta a la base de datos para obtener como respuesta reportes sobre los usuarios.
Resumen:	Este caso de uso es iniciado por el operador, cuyo inicio se efectúa al presionar el botón de generar reporte en la pantalla correspondiente después de haber seleccionado los campos requeridos.
Precondiciones:	Pantalla principal, Menú del operador.
Flujo principal:	De operador hacia la base de datos. Además involucra una consulta al registro.
Subflujos:	N/A
PostCondiciones:	N/A
Excepciones:	E1: No poder crear el reporte. E2: No rellenar la información correctamente . E3: No poder leer la base de datos . E4: Si no existen registros de la búsqueda .

Tabla 4.1.6.2.1.2 Descripción del caso de uso Generación de reportes del Sistema de Administración de Recursos Humanos.

Caso de uso:	Usuario existente
Actores:	Usuario, Base de datos, Administrador, Operador
Propósito:	Realizar acceso a la pantalla principal en base a un conjunto de datos de entrada.
Resumen:	Este caso de uso es iniciado cuando el usuario existente registra su password de entrada Y se realiza una consulta en la base de datos para permitir su acceso a la pantalla principal
Precondiciones:	N/A
Flujo principal:	Del front-end a la base de datos.
Subflujos:	S1: El usuario existe S2: El usuario es operador
Postcondiciones:	Pantalla principal con menú de usuario.
Excepciones:	E1: Al no realizarse correctamente la búsqueda de datos de entrada . E2: Al no teclear correctamente los datos de entrada.

Tabla 4.1.6.2.1.3 Descripción del caso de uso Usuario existente del Sistema de Administración de Recursos Humanos.

4.1.6.2 Diagrama de casos de uso para el usuario.

Diagrama de casos de uso para usuarios

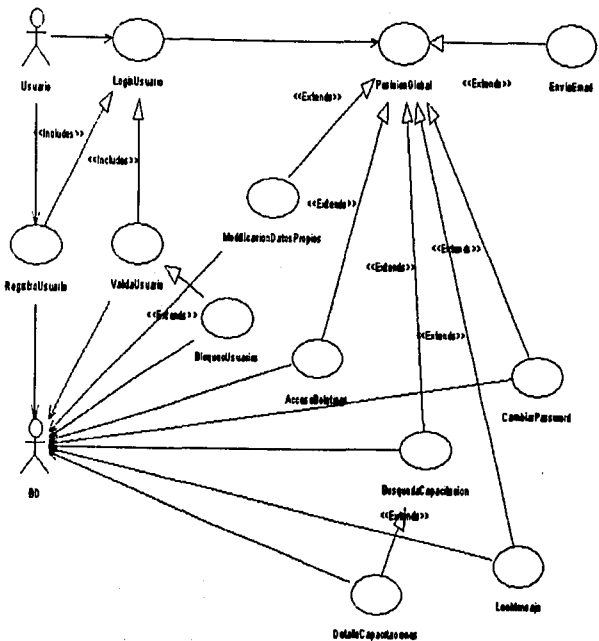


Figura 4.1.6.2.1. Diagrama de caso de uso Usuario

4.1.6.3.2 Diagrama de detalle de capacitaciones.

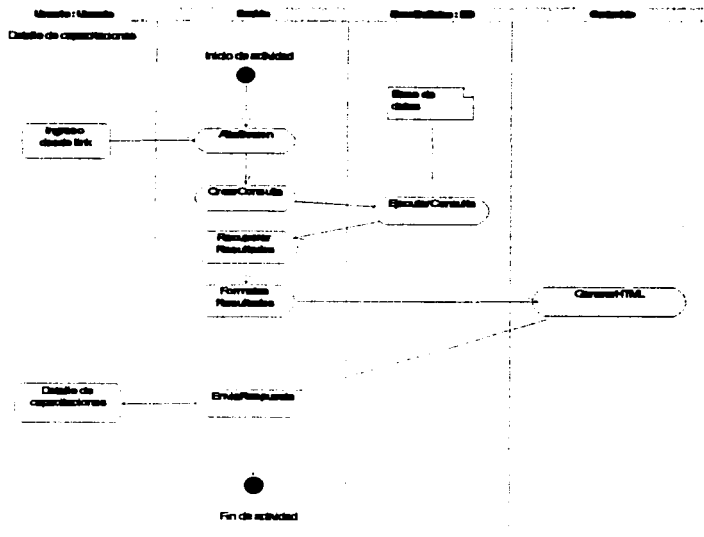


Figura 4.1.6.3.2.1 Diagrama de actividad Detalle de capacitaciones

TESIS CON
FALLA DE ORIGEN

4.1.6.3.3 Diagrama de actividad modificación de password.

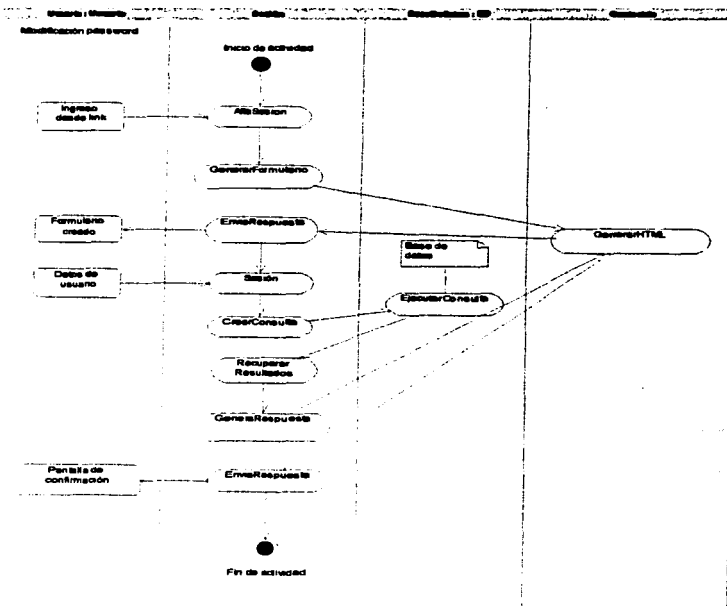


Figura 4.1.6.3.3 Diagrama de actividad modificación de password.

TESIS CON
FALLA DE ORIGEN

4.1.7 Modelo de análisis.

El objetivo del modelo de análisis es comprender y generar una arquitectura de objetos para el sistema con base a lo especificado en el modelo de requisitos. Durante esta etapa no se considera el ambiente de implantación, lo cual incluye al lenguaje de programación, manejador de base de datos, distribución o configuración de hardware, etc.

4.1.7.1 Identificación de clases.

Identificación de clases según estereotipos.

Para llevar a cabo la transición del modelo de requisitos al modelo de análisis se deben identificar los objetos necesarios para implementar todos los casos de uso.

La arquitectura de objetos debe considerar los tres tipos de estereotipos de objetos. Para lograr esto se deben identificar primero las clases de interface, las de entidad y finalmente las de control. Por otro lado, los objetos entidad e interface deben contener funcionalidad más bien local, limitando su efecto en los demás objetos.

Las clases interfaz son bastante fáciles de identificar, donde se cuenta con al menos tres estrategias:

- Se pueden identificar con base en los actores.
- Se pueden identificar con base en las descripciones de las interfaces del sistema que acompañan al modelo de requisitos.
- Se pueden identificar con base en las descripciones de los casos de uso y extraer la funcionalidad que es específica a las interfaces.

Entidad

Se utilizan clases entidad para modelar la información que el sistema debe manejar a corto y largo plazo. La información a corto plazo existe por lo general durante la ejecución del caso de uso, mientras que la información a largo plazo sobrevive a los casos de uso, por lo cual es necesario guardar esta información en alguna base de datos.

TESIS CON
FALLA DE ORIGEN

Las clases entidad se identifican en los casos de uso, donde la mayoría se identifican del modelo del dominio del problema en el modelo de requisitos. Es por lo tanto esencial trabajar de forma organizada cuando se modelan clases de tipo entidad.

La siguiente es una lista de las operaciones típicas que deben ser ofrecidas por una clase de tipo entidad:

- Guardar y traer información.
- Comportamiento que debe modificarse si el objeto entidad cambia.
- Crear y remover el objeto entidad.

Control.

Las clases de tipo control típicamente actúan como "pegamento" entre los otros tipos de clases y por lo tanto proveen la comunicación entre los demás tipos de clases. Son típicamente las más efímeras de todos los tipos de clases, dependiendo de la existencia del propio caso de uso.

Las clases de tipo control se identifican directamente de los casos de uso. Como primera aproximación, se asigna una clase de control a cada caso de uso concreto y abstracto. Dado que se asigna inicialmente el comportamiento a las clases interfaz y entidad para cada caso de uso, el comportamiento restante se asigna a las clases de control.

4.1.7.1.1 Selección de clases.

Selección de Clases

A partir de los tipos de clases que se deben identificar, se realizó la selección de clases que intervienen en el Sistema de Administración de Recursos Humanos, describiéndolas en las tablas 4.1.7.1.1.1, 4.1.7.1.1.2, 4.1.7.1.1.3

CASO DE USO	ACTORES	CLASE INTERFAZ
Alta catálogos	Usuario, Base de datos	Usuario, Base de Datos
Baja catálogos	Usuario, Base de datos	Usuario, Base de Datos
Modificación catálogos	Usuario, Base de datos	Usuario, Base de Datos
Reactivar usuarios	Usuario, Base de datos	Usuario, Base de Datos
CASO DE USO		CLASE ENTIDAD
Alta catálogos	Usuario, Base de datos	Catálogo
Baja catálogos	Usuario, Base de datos	Catálogo
Modificación catálogos	Usuario, Base de datos	Catálogo
Reactivar usuarios	Usuario, Base de datos	Persona

CASO DE USO		CLASE CONTROL
Alta catálogos	Usuario, Base de datos	Manejador de Catálogo
Baja catálogos	Usuario, Base de datos	Manejador de Catálogo
Modificación catálogos	Usuario, Base de datos	Manejador de Catálogo
Reactivar usuarios	Usuario, Base de datos	Persona

Tabla 4.1.7.1.1.1 Selección de clases tipo interfaz, control e identidad del administrador

CASO DE USO	ACTORES	CLASE INTERFAZ
Búsqueda avanzada	Usuario, Base de datos	Usuario, Base de datos
Búsqueda simple	Usuario, Base de datos	Usuario, Base de datos
Decisión	Usuario, Base de datos	Usuario, Base de datos
Modificar boletines	Usuario, Base de datos	Usuario, Base de datos

CASO DE USO	ACTORES	CLASE ENTIDAD
Búsqueda avanzada	Usuario, Base de datos	Persona
Búsqueda simple	Usuario, Base de datos	Persona
Decisión	Usuario, Base de datos	Mensaje
Modificar boletines	Usuario, Base de datos	Boletín

CASO DE USO	ACTORES	CLASE CONTROL
Búsqueda avanzada	Usuario, Base de datos	Manejador de búsqueda
Búsqueda simple	Usuario, Base de datos	Manejador de búsqueda
Decisión	Usuario, Base de datos	Manejador de decisión
Modificar boletines	Usuario, Base de datos	Manejador de Boletín

Tabla 4.1.7.1.1.2 Selección de clases tipo interfaz, control e identidad del Operador.

CASO DE USO	ACTORES	CLASE INTERFAZ
Login usuario	Usuario	Login
Registro usuario	Usuario, Base de datos	Registro Base de datos
Valida usuario	Base de datos	Base de datos
Bloques usuario	Base de datos	Base de datos
Posición global	Usuario, Base de datos	Usuario, Base de datos
Enviar e-mail	Usuario, Base de datos	Usuario, Base de datos
Cambiar password	Usuario, Base de datos	Usuario, Base de datos
Leer mensaje	Usuario, Base de datos	Usuario, Base de datos
Búsqueda de capacitac.	Usuario, Base de datos	Usuario, Base de datos
Detalle de capacitac.	Usuario, Base de datos	Usuario, Base de datos
Acceso a boletines	Usuario, Base de datos	Usuario, Base de datos
Modificación datos propios	Usuario, Base de datos	Usuario, Base de datos

CASO DE USO	ACTORES	CLASE ENTIDAD
Login usuario	Usuario	
Registro usuario	Usuario, Base de datos	Registro usuario
Valida usuario	Base de datos	Registro usuario, Sesión usuario
Bloques usuario	Base de datos	Sesión
Posición global	Usuario, Base de datos	Sesión
Cambiar password	Usuario, Base de datos	Cambiar password
Leer mensaje	Usuario, Base de datos	Mensajes usuarios

Búsqueda de capacitac.	Usuario, Base de datos	Capacitación
Detalle de capacitac.	Usuario, Base de datos	Curso
Acceso a boletines	Usuario, Base de datos	Sesión, boletín
Modificación datos propios	Usuario, Base de datos	Sesión, persona

CASO DE USO	ACTORES	CLASE CONTROL
Login usuario	Usuario	
Registro usuario	Usuario, Base de datos	Manejador registro usuario
Valida usuario	Base de datos	Manejador valida usuario
Bloques usuario	Base de datos	Manejador bloqueo usuario
Posición global	Usuario, Base de datos	Manejador principal
Cambiar password	Usuario, Base de datos	Manejador cambiar password
Leer mensaje	Usuario, Base de datos	Manejador mensaje
Búsqueda de capacitac.	Usuario, Base de datos	Manejador capacitación
Detalle de capacitac.	Usuario, Base de datos	Manejador curso
Acceso a boletines	Usuario, Base de datos	Manejador boletines
Modificación datos propios	Usuario, Base de datos	Manejador datos propios Manejador búsqueda Manejador Catálogos

Tabla 4.1.7.1.1.3 Selección de clases tipo interfaz, control e identidad del Caso de uso Posición Global.

TESIS CON
FALLA DE ORIGEN

4.1.7.1.3 Diagrama dinámico de clases.

Diagramas de secuencia.

Los diagramas de secuencia pueden generarse a partir de las propias clases o incluso a partir de la interacción entre subsistemas. Esto es a menudo una técnica muy útil ya que se puede diseñar un subsistema mostrando sólo las interfaces de los subsistemas relacionados. A un nivel más detallado se pueden mostrar las interacciones internas del subsistema entre los propios objetos. Normalmente, los *eventos* en el diagrama corresponden a protocolos y se especifican exactamente como se venían en el código final.

4.1.7.1.3.1 Diagramas de secuencia del Sistema de Administración de Recursos Humanos.

En este punto se muestran algunos diagramas de secuencia o dinámicos utilizados para el sistema de Administración de Recursos Humanos, los cuales serán los de Login, Registro de datos y Navegación de usuario.

En la Figura 4.1.7.1.3.1.1 que es el diagrama de secuencia del Login, se puede observar cómo el usuario empieza el recorrido enviando datos a la clase Login, posteriormente en la clase Manejador Acceso Usuario se hará el movimiento de la validación del usuario a la clase Base de datos, la que a su vez regresará la respuesta al usuario para que pueda ingresar a la clase Manejador Principal, ahí se desplegará la clase Menú Principal. También en Menú Principal se utiliza la clase Manejador de Mensajes en la que se hará envío y recepción de éstos junto con la clase Base de datos, además de hacer uso de la lectura de boletines; todo lo anterior con sus respectivas respuestas.

TESIS CON
FALLA DE ORIGEN

En la figura 4.1.7.1.3.1.2 se muestra el diagrama de secuencia de Registro de datos, en primera instancia el usuario se registra a través de la clase Registro, posteriormente la clase Manejador Acceso Usuario realizará el movimiento correspondiente hacia la clase Base de datos, obteniendo la respectiva validación del usuario; la clase Manejador Acceso Usuario registrará la sesión en la clase Sesión obteniendo su respuesta para poder enviar al usuario a la clase Manejador Principal en donde visualizará el Menú Principal.

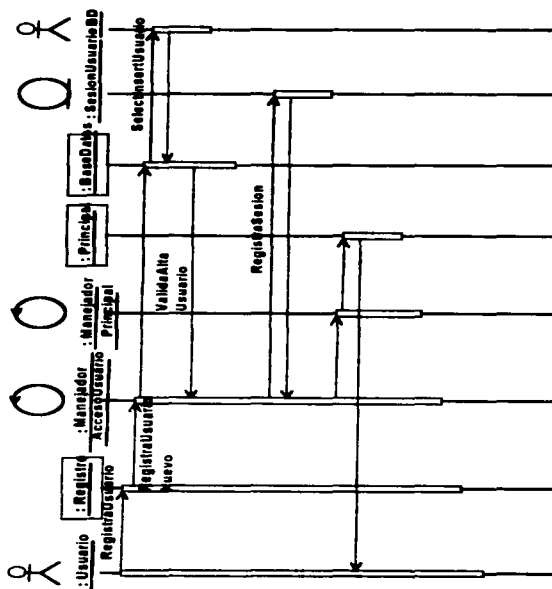


Figura 4.1.7.1.3.1.2 Diagrama de secuencia de Registro.

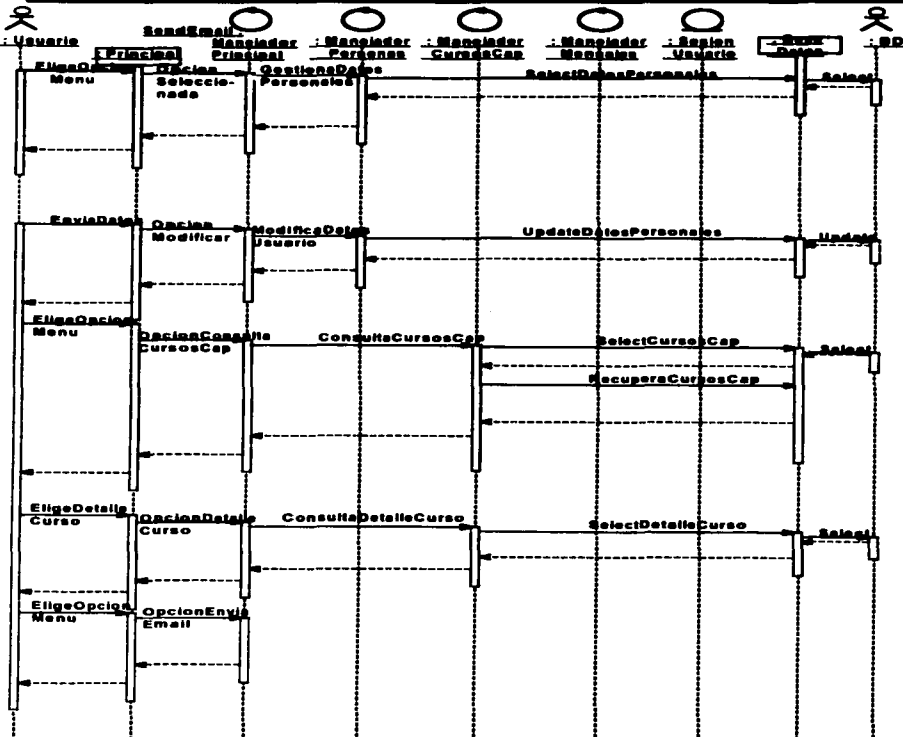


Figura 4.1.7.1.3.1.3 Diagrama de secuencia del usuario dentro del sistema

En la figura 4.1.7.1.3.1.3 se observa el diagrama de secuencia del usuario, en el que se elige la clase Menú principal, se selecciona la opción deseada en la clase Manejador Principal, la clase Manejador de personas gestiona datos personales, en la clase Base de datos, se buscan dichos datos personales y se envían las respuestas correspondientes.

También tiene como opción Modificar datos en la clase Manejador de Personas, que envía esta secuencia a la clase base de datos realizándose la actualización correspondiente, si la opción fue seleccionada.

En la clase Menú principal se elige Consulta de cursos de capacitación, para este movimiento la clase Manejador Principal enviará la petición a la clase Manejador de Cursos de Capacitación, la que a su vez se conectará con la clase Base de datos para realizar la búsqueda enviando resultados al usuario.

Al elegir la opción Detalle de curso en la clase Manejador Principal, ésta envía la petición a la clase Manejador de Cursos de Capacitación y también apoyándose en la clase Base de datos realizará la búsqueda pertinente enviando resultados al usuario.

Se escoge además la opción Enviar e-mail, la cual se encuentra en la clase Manejador Principal, obteniendo también la respuesta correspondiente.

4.1.8 Modelo de diseño.

El modelo de diseño es un refinamiento y formalización adicional del modelo de análisis donde se toman en cuenta las consecuencias del ambiente de implementación. El resultado del modelo de diseño son especificaciones muy detalladas de todos los objetos, incluyendo sus operaciones y atributos.

TESIS CON
FALLA DE ORIGEN

4.1.8.1 Tarjetas CRC.

Tarjetas de Clase.

Las tarjetas de clases (también conocidas como tarjetas CRC: Clase-Responsabilidad-Colaboración) permiten al diseñador visualizar las diferentes clases de manera independiente y detallada.

La tarjeta se divide en tres secciones:

- Encabezado. Consiste del nombre de la *clase*, el *módulo* al que pertenece la clase, las **propiedades** de la clase (abstracta o concreta), el **estereotipo** de la clase (entidad, interface o control), una lista de **superclases** y otra lista de **subclases**.
- Dos columnas debajo del encabezado, correspondientes a las **responsabilidades** (a la izquierda) y **colaboraciones** (a la derecha) de la clase. Eventualmente en la columna izquierda se incluirá información sobre los **contratos**. El número de filas en estas dos columnas es extensible y no está limitado.

Clase:	
Módulo:	
Propiedades:	
Estereotipo:	
Superclase:	
Subclase:	

Tabla 4.1.8.1.1 Especificación de Tarjetas CRC.

En la Figura 4.1.8.1.2 se muestra una tarjeta de la clase **Manejador Principal**, aplicada para el **Sistema de Administración de Recursos Humanos** en donde se especifican los conceptos anteriormente mencionados como son: el modulo al que pertenece, sus propiedades, el estereotipo, sus responsabilidades y colaboraciones.

Clase: Manejador Principal	
Módulo: Operación	
Propiedades: Concreta	
Estereotipo : Control	
Superclase:	
Subclase :	
Envía despliegue de posición global	Principal (interfaz)
Envía obtención de mensajes	ManejadorMensajes
Envía consulta datos de sesión	SesiónUsuario (entidad)
Envía gestión de datos personales	ManejadorPersonas
Envía gestion CursosCap	ManejadorCursosCap
Envía e-mail	
Envía gestión de búsquedas	ManejadorBúsquedas
Envía gestión de mensajes boletín y decisión	ManejadorMensajes
Envía gestión de catálogos	ManejadorCatalogos
Envía reactivación de usuario	ManejadorPersonas

Tabla 4.1.8.1.2 Tarjeta de la Clase Manejador Principal

4.1.8.1.2 Diagramas de colaboración.

Se utilizan los diagramas de colaboración en donde se despliegan las colaboraciones entre clases y subsistemas de forma gráfica.

Los responsabilidades de una clase se ofrecen como contratos de una clase servidor. Los contratos se muestran como círculos externos a la clase a la cual pertenecen relacionada con la clase mediante una liga de realización de interfaz, como se muestra en la Figura 4.1.8.1.2.1



Figura 4.1.8.1.2.1 Diagrama de clase con contrato

Se dibuja un círculo por contrato y se numeran los círculos con el número asignado al contrato correspondiente.

Las colaboraciones entre clases se representan por una flecha correspondiente a una asociación de *utilización* del cliente al contrato apoyado por el servidor, como se muestra en la Figura 4.1.8.1.2.2 (la flecha de la derecha proveniente de Clase 2).



Figura 4.1.8.1.2.2 Diagrama de colaboración donde Clase 2 es cliente del Contrato 1 de Clase 1.

Si dos clases colaboran con un mismo contrato de otra clase, se dibujan las flechas al mismo círculo. Por el contrario, se dibujan las flechas a círculos distintos correspondientes a diferentes contratos.

Los diagramas de colaboración para aplicaciones moderadamente grandes pueden volverse grandes y complejos. Por lo tanto, se puede ocultar información para simplificarlos ligeramente. Los subsistemas se muestran en el diagrama de colaboración dibujando rectángulos envolviendo las clases y contratos que las forman, como se ve en la Figura 4.1.8.1.2.3

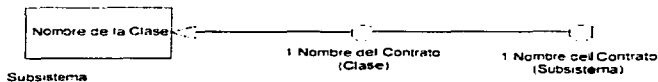


Figura 4.1.8.1.2.3 Diagrama de subsistema encapsulando clases y contratos.

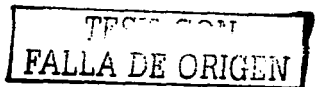
4.2 Diseño y Construcción del Back End.

4.2.1 Diseño y construcción de la base de datos (back end).

Para construir una base de datos en PostgreSQL existen varios mecanismos, ya sea desde la línea de comandos, desde el intérprete de comandos de PostgreSQL o con herramientas gráficas de edición y mantenimiento de las bases de datos.

- Para crear una base de datos desde la línea de comandos (del sistema operativo), se ejecuta el comando `createdb`, que se instala junto con PostgreSQL. Al ejecutarse con el nombre de la nueva base de datos, el comando genera la sentencia SQL necesaria para crear una base de datos (`CREATE DATABASE`), se conecta al servidor de base de datos (`postmaster`) e intenta ejecutar la sentencia. Para que la creación de una base de datos sea exitosa, es necesario que el usuario cuente con permisos de creación de bases de datos dentro del sistema y que la conexión sea posible (que el puerto esté bien configurado).
- Para crear una base de datos desde el intérprete de comandos "psql", primero hay que conectarse con el servidor de base de datos, ejecutando el comando `psql` especificando el nombre de usuario y contraseña para poder acceder al sistema. Una vez dentro, se debe ejecutar la sentencia "CREATE DATABASE" y el nombre de la base de datos que queremos crear. Nuevamente, es necesario que el usuario con el que se estableció la conexión tenga permisos de creación de bases de datos.
- Para crear una base de datos utilizando una herramienta gráfica solamente es necesario ejecutarla.

En nuestro caso, la creación de la base de datos, las tablas y las relaciones necesarias la efectuamos con la herramienta PGAccess, que se distribuye libremente bajo la licencia GPL (GNU Public License, Licencia Pública GNU). Para su instalación, gracias a la plataforma sobre la cual estamos desarrollando (DEBIAN GNU Linux), solamente se ejecutó el comando `apt-get install pgaccess`.



Al ejecutar el programa para crear la base de datos, se siguieron los siguientes pasos:

1. Del menú "Base de Datos" se selecciona la opción nueva base de datos, como se muestra en la figura 4.2.1.1

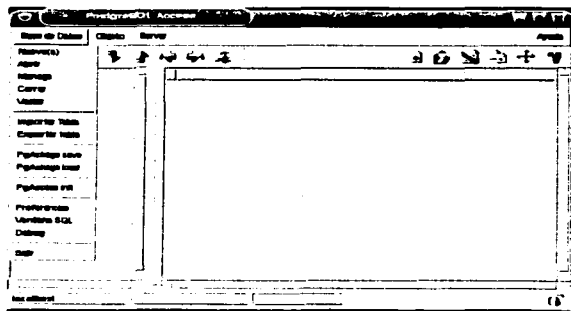


Figura 4.2.1.1 Creación de una base de datos

2. En la ventana que se abre, teclear el nombre de la nueva base de datos como se muestra en la figura 4.2.1.2 Al dar clic en el botón "Crear" la base de datos estará creada y lista para ser utilizada.

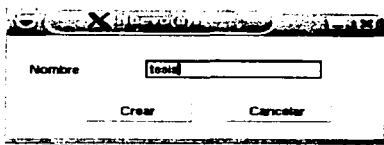


Figura 4.2.1.2 Nombre de la nueva base de datos

3. Una vez que la base de datos fue creada, es necesario dar clic en el botón de conectar base de datos, para que aparezca una ventana que contiene información de la conexión a la base de datos, como el host, el puerto, el nombre de la base de datos y el nombre de usuario. En la figura 4.2.1.3 podemos observar la opción del menú "Objeto".

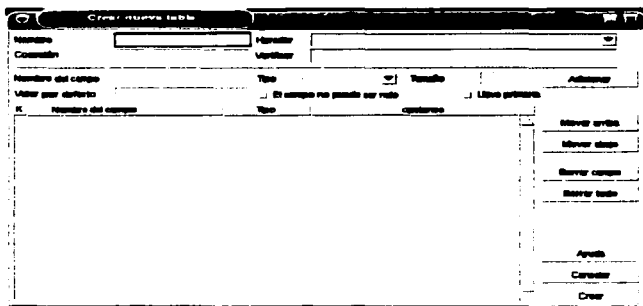


Figura 4.2.1.4 Ventana de tabla nueva

5. Una vez que todos los campos de la tabla han sido completados (como se muestra en la figura 4.2.1.5), se da clic en el botón "Crear" para que se genere la tabla.

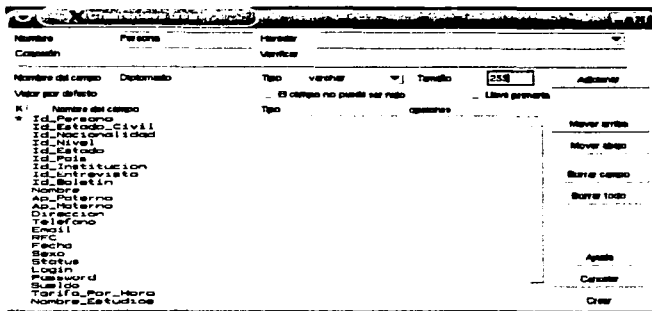


Figura 4.2.1.5 Campos de la tabla Personas

6. Los pasos 4 y 5 se repiten para todas las tablas de la base de datos hasta visualizar una pantalla como la que se muestra en la figura 4.2.1.6

Nombre	Propiedades	OID	Tipo	Actividad
Actividad	activ	24823	1000	3
Actividad_Instancia	activ	24828	1000	3
Actividad_Materia	activ	24877	1000	2
Actividad_Materia	activ	24880	1000	4
Actividad_Materia	activ	24881	1000	2
Actividad_Materia	activ	24882	1000	3
Actividad_Materia	activ	24883	1000	3
Actividad_Materia	activ	24884	1000	4
Actividad_Materia	activ	24885	1000	6
Actividad_Materia	activ	24886	1000	3
Actividad_Materia	activ	24887	1000	3
Actividad_Materia	activ	24888	1000	2
Actividad_Materia	activ	24889	1000	2
Actividad_Materia	activ	24890	1000	3
Actividad_Materia	activ	24891	1000	3
Actividad_Materia	activ	24892	1000	2
Actividad_Materia	activ	24893	1000	2
Actividad_Materia	activ	24894	1000	2
Actividad_Materia	activ	24895	1000	2
Actividad_Materia	activ	24896	1000	2
Actividad_Materia	activ	24897	1000	2
Actividad_Materia	activ	24898	1000	2
Actividad_Materia	activ	24899	1000	2
Actividad_Materia	activ	24900	1000	2
Actividad_Materia	activ	24901	1000	2
Actividad_Materia	activ	24902	1000	2
Actividad_Materia	activ	24903	1000	2
Actividad_Materia	activ	24904	1000	2
Actividad_Materia	activ	24905	1000	2
Actividad_Materia	activ	24906	1000	2
Actividad_Materia	activ	24907	1000	2
Actividad_Materia	activ	24908	1000	2
Actividad_Materia	activ	24909	1000	2
Actividad_Materia	activ	24910	1000	2
Actividad_Materia	activ	24911	1000	2
Actividad_Materia	activ	24912	1000	2
Actividad_Materia	activ	24913	1000	2
Actividad_Materia	activ	24914	1000	2
Actividad_Materia	activ	24915	1000	2
Actividad_Materia	activ	24916	1000	2
Actividad_Materia	activ	24917	1000	2
Actividad_Materia	activ	24918	1000	2
Actividad_Materia	activ	24919	1000	2
Actividad_Materia	activ	24920	1000	2
Actividad_Materia	activ	24921	1000	2
Actividad_Materia	activ	24922	1000	2
Actividad_Materia	activ	24923	1000	2
Actividad_Materia	activ	24924	1000	2
Actividad_Materia	activ	24925	1000	2
Actividad_Materia	activ	24926	1000	2
Actividad_Materia	activ	24927	1000	2
Actividad_Materia	activ	24928	1000	2
Actividad_Materia	activ	24929	1000	2
Actividad_Materia	activ	24930	1000	2
Actividad_Materia	activ	24931	1000	2
Actividad_Materia	activ	24932	1000	2
Actividad_Materia	activ	24933	1000	2
Actividad_Materia	activ	24934	1000	2
Actividad_Materia	activ	24935	1000	2
Actividad_Materia	activ	24936	1000	2
Actividad_Materia	activ	24937	1000	2
Actividad_Materia	activ	24938	1000	2
Actividad_Materia	activ	24939	1000	2
Actividad_Materia	activ	24940	1000	2
Actividad_Materia	activ	24941	1000	2
Actividad_Materia	activ	24942	1000	2
Actividad_Materia	activ	24943	1000	2
Actividad_Materia	activ	24944	1000	2
Actividad_Materia	activ	24945	1000	2
Actividad_Materia	activ	24946	1000	2
Actividad_Materia	activ	24947	1000	2
Actividad_Materia	activ	24948	1000	2
Actividad_Materia	activ	24949	1000	2
Actividad_Materia	activ	24950	1000	2
Actividad_Materia	activ	24951	1000	2
Actividad_Materia	activ	24952	1000	2
Actividad_Materia	activ	24953	1000	2
Actividad_Materia	activ	24954	1000	2
Actividad_Materia	activ	24955	1000	2
Actividad_Materia	activ	24956	1000	2
Actividad_Materia	activ	24957	1000	2
Actividad_Materia	activ	24958	1000	2
Actividad_Materia	activ	24959	1000	2
Actividad_Materia	activ	24960	1000	2
Actividad_Materia	activ	24961	1000	2
Actividad_Materia	activ	24962	1000	2
Actividad_Materia	activ	24963	1000	2
Actividad_Materia	activ	24964	1000	2
Actividad_Materia	activ	24965	1000	2
Actividad_Materia	activ	24966	1000	2
Actividad_Materia	activ	24967	1000	2
Actividad_Materia	activ	24968	1000	2
Actividad_Materia	activ	24969	1000	2
Actividad_Materia	activ	24970	1000	2
Actividad_Materia	activ	24971	1000	2
Actividad_Materia	activ	24972	1000	2
Actividad_Materia	activ	24973	1000	2
Actividad_Materia	activ	24974	1000	2
Actividad_Materia	activ	24975	1000	2
Actividad_Materia	activ	24976	1000	2
Actividad_Materia	activ	24977	1000	2
Actividad_Materia	activ	24978	1000	2
Actividad_Materia	activ	24979	1000	2
Actividad_Materia	activ	24980	1000	2
Actividad_Materia	activ	24981	1000	2
Actividad_Materia	activ	24982	1000	2
Actividad_Materia	activ	24983	1000	2
Actividad_Materia	activ	24984	1000	2
Actividad_Materia	activ	24985	1000	2
Actividad_Materia	activ	24986	1000	2
Actividad_Materia	activ	24987	1000	2
Actividad_Materia	activ	24988	1000	2
Actividad_Materia	activ	24989	1000	2
Actividad_Materia	activ	24990	1000	2
Actividad_Materia	activ	24991	1000	2
Actividad_Materia	activ	24992	1000	2
Actividad_Materia	activ	24993	1000	2
Actividad_Materia	activ	24994	1000	2
Actividad_Materia	activ	24995	1000	2
Actividad_Materia	activ	24996	1000	2
Actividad_Materia	activ	24997	1000	2
Actividad_Materia	activ	24998	1000	2
Actividad_Materia	activ	24999	1000	2
Actividad_Materia	activ	25000	1000	2

Figura 4.2.1.6 Tablas de la base de datos tesis

7. Una vez concluida la creación de las tablas es necesario generar sus relaciones, lo cual se hace creando un nuevo diagrama (de la última opción del menú izquierdo) como se muestra en la figura 4.2.1.7

TESIS CON
FALLA DE ORIGEN

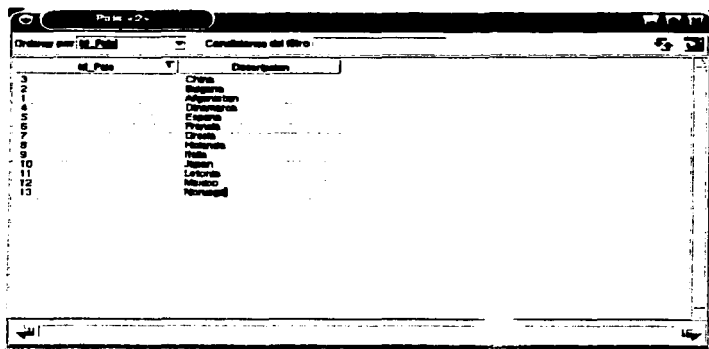


Figura 4.2.1.8 Vista del contenido de una tabla

9. Para la creación de vistas en la base de datos se sigue un procedimiento similar a la creación de tablas, utilizando la herramienta gráfica. En la figura 4.2.1.9 se muestra la pantalla inicial de creación de una nueva vista.

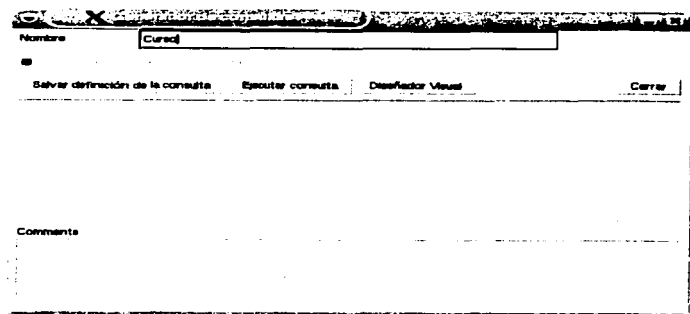


Figura 4.2.1.9 Ventana de creación de una nueva vista

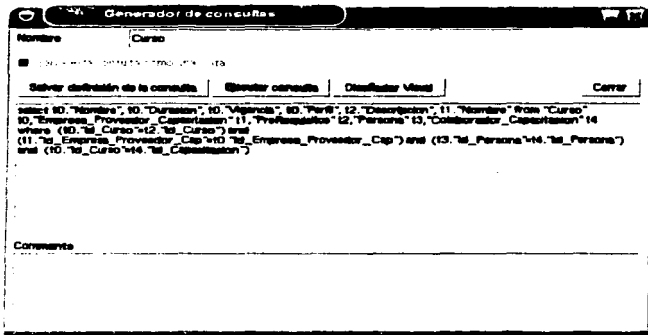


Figura 4.2.1.11 Consulta SQL almacenada

4.2.2 Seguridad en la base de datos.

PostgreSQL maneja seguridad distinta a la proporcionada por el sistema operativo, es decir, los usuarios de la base de datos son distintos a los del sistema UNIX. Los usuarios creados para el sistema manejador de base de datos son generales para cualquier base de datos, por lo que es muy importante manejar correctamente los permisos que tengan directamente sobre las tablas, vistas, consultas y otros elementos de cada base de datos.

De igual forma que para la creación de bases de datos existen diversas formas de creación de usuarios, y se enlistan a continuación:

- Al instalar la PostgreSQL se instalan también programas que se ejecutan desde la línea de comandos para la creación, modificación y eliminado de usuarios de la base de datos. Para crear un usuario con esta opción, solamente es necesario teclear el comando *createuser nombre*.

- Otra forma de creación de usuarios para la base de datos es el comando SQL *CREATE USER nombre* que realiza las mismas funciones que el programa de línea de comandos.
- Finalmente, la creación de usuarios de bases de datos puede ser realizada utilizando las herramientas gráficas que se tengan a disposición. Además de ser una forma mejor controlada para crear los usuarios, es fácil también asignar, modificar o revocar los permisos asignados para cada usuario en la base de datos seleccionada. En la figura 4.2.2.1 se muestra la ventana de control de los usuarios del sistema manejador.

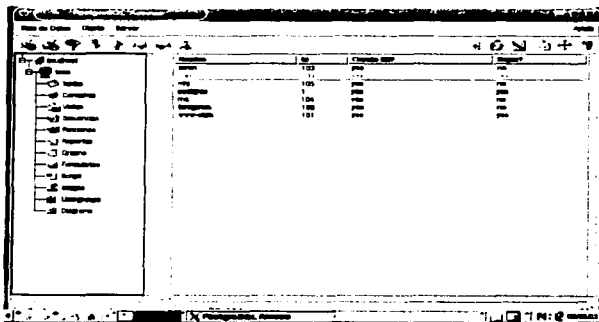


Figura 4.2.2.1 Ventana de manejo de usuarios

Con estos mecanismos de creación de usuarios independientes al sistema operativo la seguridad de las bases de datos queda exclusivamente a manos del administrador del sistema de base de datos, con lo que se consigue un mejor uso y seguridad de la base de datos. Para nuestro sistema esto es primordial, ya que los usuarios del mismo no deben tener acceso directo a la base de datos, y posiblemente para acceder al sistema sea necesario que tengan un usuario válido de sistema operativo. De igual forma, todos los

cambios que se deban realizar a la información contenida en el sistema deben ser realizados utilizando el sistema mismo, de ninguna manera es recomendable realizar modificaciones a la base de datos, aunque con los permisos correspondientes, es posible.

Para poder tener acceso a la información, tanto de consulta como registro y actualización es necesario tener un usuario que funcione como interfaz entre el sistema y la base de datos. Para ello, el servidor web provee un usuario que será el encargado de ejecutar todos los accesos al sistema de archivos y la base de datos. Este usuario debe estar dado de alta en el sistema de base de datos para que pueda manipular la información necesaria y evitar así que cada usuario del sistema se convierta por necesidad en un usuario de la base de datos. Esto, además de permitir un mejor control de acceso a los usuarios del sistema, facilita la labor del administrador de la base de datos al no tener que manejar un gran número de usuarios.

4.2.3 Objetos de control y su relación con la base de datos.

Con el fin de que el sistema tenga acceso a la información contenida en la base de datos, es necesario crear objetos de control que sean encargados de llevar a cabo los accesos, ya sea de consulta como de modificación o creación de registros. Dichos objetos se encargan de manejar el flujo de la información, tanto la que llega desde las interfaces de usuario como la interfaz de base de datos.

A continuación se enlistan los objetos de control del sistema y se hace una breve descripción de su relación con la base de datos.

- **Manejador de acceso del usuario.** Este objeto controla el acceso del usuario al sistema basándose para ello en los datos de entrada al sistema (nombre de usuario y contraseña), los cuales hay que validar con la información contenida en la base de datos, en la tabla "Persona"

- **Manejador principal.** Este objeto se encarga de generar la página de respuesta al usuario, dependiendo de la página de origen, del nivel de acceso del usuario y de la página solicitada por el usuario. Para ello, el manejador debe llevar a cabo operaciones para definir el contenido del menú, que se encuentra en la base de datos.
- **Manejador de mensajes.** Este objeto debe verificar la identidad del usuario en la tabla "Persona" para después ubicar en la tabla "Mensaje" y su relación aquellos mensajes que han sido enviados al usuario. También se encarga de afectar la tabla "Mensaje" cuando el usuario envíe un mensaje entrante hacia otro usuario o cuando se realicen modificaciones a la información contenida en la base de datos.
- **Manejador de capacitaciones.** Este objeto es en realidad un motor de búsquedas, pues es el encargado de llevar a cabo búsquedas en la tabla "Curso" de acuerdo a los parámetros de entrada que el usuario haya enviado.
- **Manejador de personas.** Este objeto se encarga de obtener y controlar la información de una persona, ya sea un candidato o un colaborador (empleado) para ser desplegada al usuario o modificada en la base de datos.
- **Manejador de búsquedas.** Este objeto es el motor principal de búsquedas, pues maneja las búsquedas de personas, tanto en su modalidad simple como avanzada, y para ello se encarga de realizar consultas a toda la base de datos, dependiendo de los criterios que el usuario (el operador en este caso) haya introducido. También se encarga de formatear los resultados para generar enlaces dinámicos a un mayor detalle de la información.
- **Manejador de catálogos.** Este objeto se encarga de extraer la información de los catálogos para desplegarla al usuario, o de realizar modificaciones que solamente un administrador u operador pueden efectuar. El manejador de catálogos afecta directamente a los catálogos que se encuentren contenidos en la base de datos del sistema.

4.3. Diseño y Construcción del Front End.

Para crear un aplicación en PHP lo primero que necesitamos es un editor de texto. El editor de texto al cual nos referiremos es *vi* de UNIX.

vi

Vi es un programa que se usa para editar archivos que se componen con texto, como una carta, un programa en C o un archivo de configuración del sistema o en este caso para un programa en PHP. Aunque hay disponibles muchos editores de texto, el único editor que es seguro que se va a encontrar en cualquier sistema UNIX es *vi*. El editor *vi* no es el editor más fácil de usar, ni es muy autoexplicativo. Sin embargo, como *vi* es tan común en el mundo UNIX/Linux, y algunas veces necesario, merece un tratamiento aquí. La forma de empezar a utilizar el editor de texto *Vi* será escribiendo en la línea de comandos la siguiente instrucción:

Vi nombre_del_archivo

Los comandos se realizan mediante teclas, ejemplo: *a*, entra en modo texto a la derecha del cursor. Las teclas o secuencias de teclas están "encerradas" entre guiones, ejemplo: *-a-*, entra en modo texto a la derecha del cursor. Otras órdenes se realizan mediante la tecla CONTROL y otra, ejemplo: *Ctrl-F*, avanza una página, estas últimas no están encerradas. A continuación se mencionan algunos comandos; cabe aclarar que son los comandos básicos ya que *vi* cuenta con una gran cantidad de comandos para hacer una aplicación.

Comandos de desplazamiento.

Ctrl-D : Avanza media página

Ctrl-U : Retrocede media página

Ctrl-F : Avanza una página también *AvPág*

Ctrl-B : Retrocede una página también *RePág*

-l- : avanza un char, también flecha derecha

-h- : retrocede un char, también flecha izquierda

TESIS CON
FALLA DE ORIGEN

- k- : sube una línea, también flecha arriba
- j- : baja una línea, también flecha abajo

Comandos de edición.

Para volver a modo comando desde el modo de edición, apretar la tecla ESC.

- a- : inserta a la derecha del cursor
- A- : inserta al final de la línea
- i- : inserta a la izquierda del cursor
- I- : inserta al inicio de la línea
- o- : abre una nueva línea debajo del cursor
- O- : abre una nueva línea arriba del cursor
- d0- : borra desde el inicio de la línea hasta el cursor
- u- : deshace el último comando.

Comandos de copiar y pegar.

El vi también permite copiar y pegar. En vi cuando se copia y pega se usa el buffer llamado paste, en este buffer se copian las líneas que se marcan.

- yy- : copia al paste la línea del cursor, -nyy- copia n líneas al paste
- p- : copia del paste debajo del cursor.

Para desarrollar una aplicación en PHP lo primero que se debe de hacer es crear el archivo con el editor de texto Vi (ver figura 4.3.1)

Vi index.php

```
File Edit View Terminal Go Help
[root@localhost root]# vi index.php
```


Una vez que el archivo fue creado podemos empezar a realizar el desarrollo de un programa en PHP, recordemos que el código de PHP es insertado dentro del código HTML. Para poder entender mejor el programa que a continuación se desarrollará, será la página principal del sistema de gestión de recursos humanos en la que el usuario, operador o administrador del sistema ingresarán en un formulario su *login* y *password*; cada uno tiene sus respectivos permisos y cabe aclarar que si el usuario no está registrado en el sistema tendrá la opción en un formulario de proporcionar los datos mínimos necesarios para ingresarlo al sistema, por ejemplo su nombre y apellidos así como el login y el password de su preferencia.

Lo primero que se coloca son las etiquetas de cabecera del cuerpo de html de esta forma:

```
.....
```

las etiquetas en rojo son las cabeceras de un programa en html

```
...../
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Bienvenido al sistema de gestión de recursos humanos</TITLE>
```

```
</HEAD>
```

```
.....
```

Ahora bien la etiqueta en azul es el cuerpo del programa

```
...../
```

```
<BODY>
```

```
.....
```

Ahora las etiquetas en verde son el código del diseño de la cabecera del sistema el cual estará implícito en todas las pantallas del mismo

```
...../
```

```
<hr>
```

```

<div align="center">
  <p><br>
    <strong><font face="Verdana, Arial, Helvetica, sans-serif">BIENVENIDO AL
  SISTEMA
    DE GESTIÓN RECURSOS HUMANOS </font></strong><font face="Verdana,
  Arial, Helvetica, sans-serif"><br>
    </font></p>
</div>
<hr>

```

.....
 las etiquetas en rojo es el código del primer formulario donde el usuario deberá
 ingresar el login y password para poder ingresar al sistema


```

<form name="form1" method="post" action="ingresar.php">
  <table width="100%" border="0">
    <tr>
      <td width="39%">&nbsp;  </td>
      <td width="56%"><table width="41%" border="1" bordercolor="#333333"
  bgcolor="#CCCCCC">
        <tr>
          <td width="42%"><div align="left"><font face="Verdana, Arial, Helvetica,
  sans-serif"><strong>Login:</strong></font></div></td>
          <td width="58%"><font face="Verdana, Arial, Helvetica, sans-serif">
            <input name="login" type="text" size="10" maxlength="10">
          </font></td>
        </tr>
      </tr>
    </tr>
  </table>

```

```

<td><div align="left"><font face="Verdana, Arial, Helvetica, sans-
serif"><strong>Password:</strong></font></div></td>
<td><font face="Verdana, Arial, Helvetica, sans-serif">
<input name="password" type="password" size="10" maxlength="10">
</font></td>
</tr>
<tr>
<td colspan="2"><div align="center"> <font face="Verdana, Arial, Helvetica,
sans-serif">
<input type="submit" value="Ingresar">
</font></div></td>
</tr>
</table></td>
<td width="5%">&nbsp;  </td>
</tr>
</table>
</form>
/.....

```

```

...../
<hr align="center" width="50%">
<table width="100%" border="0">
<tr>
<td><div align="center"><font size="2" face="Verdana, Arial, Helvetica, sans-
serif"><em><strong>Nota:
para ingresar al sistema deber&aacute; estar dado de alta en el
sistema</strong></em></font></div></td>
</tr>
<tr>
<td><div align="center"><em><strong><font size="2" face="Verdana, Arial,
Helvetica, sans-serif">Si

```

no está dado de alta entonces de clic a la siguiente liga para darse de alta en el sistema</div></td>

</tr>

<tr>

<td><div align="center">

Usuario

Nuevo</div></td>

</tr>

</table><hr>

<p> </p>

<p> </p>

<p> </p>

<p> </p>

<p> </p>

<hr>

<p align="center">Usuario

Nuevo</p>

<hr>

<form name="form2" method="post" action="AgregarUsuario.php">

<table width="100%" border="0">

<tr>

<td width="31%" height="222"> </td>

<td width="45%"><table width="100%" border="1" bgcolor="#CCCCCC">

<tr>

<td width="55%">Nombre:</td>

<td width="45%"><input name="nombre" type="text" size="20" maxlength="20"></td>

</tr>

TESIS CON
FALLA DE ORIGEN


```
<tr>
  <td><font face="Verdana, Arial, Helvetica, sans-serif"><strong>Apellido
Paterno:</strong></font></td>
  <td><input name="appat" type="text" size="20" maxlength="20"></td>
</tr>
<tr>
  <td><font face="Verdana, Arial, Helvetica, sans-serif"><strong>Apellido
Materno:</strong></font></td>
  <td><input name="apmat" type="text" size="20" maxlength="20"></td>
</tr>
<tr>
  <td><font face="Verdana, Arial, Helvetica, sans-
serif"><strong>email:</strong></font></td>
  <td><input name="email" type="text" maxlength="40"></td>
</tr>
<tr>
  <td><font face="Verdana, Arial, Helvetica, sans-
serif"><strong>Login:</strong></font></td>
  <td><input name="login" type="text" size="10" maxlength="10"></td>
</tr>
<tr>
  <td><font face="Verdana, Arial, Helvetica, sans-
serif"><strong>Password:</strong></font></td>
  <td><input name="password1" type="password" size="10"
maxlength="10"></td>
</tr>
<tr>
  <td><font face="Verdana, Arial, Helvetica, sans-
serif"><strong>Password:</strong></font></td>
  <td><input name="password2" type="password" size="10"
maxlength="10"></td>
```

```
</tr>
<tr>
  <td colspan="2"><div align="center"><font face="Verdana, Arial, Helvetica,
sans-serif">
  <input type="submit" value="Agregar al sistema">
  </font></div></td>
</tr>
</table>
</td>
<td width="24%">&nbsp;</td>
</tr>
</table>

<hr><p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
</form>
<p align="center"><strong><font face="Verdana, Arial, Helvetica, sans-
serif"></font></strong></p>
</BODY>
</html>
```

A continuación se muestran las pantallas que manejan la generación de reportes, de la Figura. 4.3.2 a la 4.3.5:

TESIS CON
FALLA DE ORIGEN

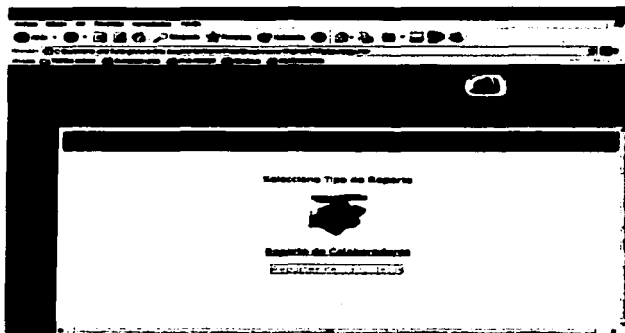


Figura 4.3.2 Pantalla principal de reportes

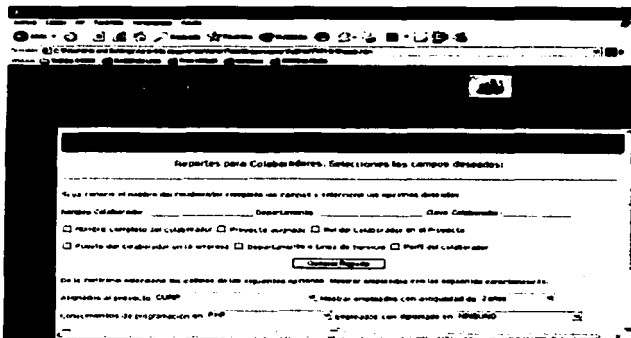


Figura 4.3.3 Reportes para coleccionadores

Reportes para Candidatos, satisficcion los campos obligados:

Si ya realizó el registro de candidatos, complete los campos siguientes y selecciona la opción de lista.

Nombre del Candidato: _____ Clase del Candidato: _____

Inscripcion completa del Candidato Inscripcion con el candidato Inscripcion Técnica

Cada la cantidad de inscripciones de la siguiente manera:

Mostrar el detalle de los datos desde Hoy _____, candidatos con el cual se realiza la inscripción en el año:

Candidatos con el cual se realiza la inscripción en el último año Candidatos que han sido habilitados de manera de presentarse en el último año.

Figura 4.3.4 Pantalla de reportes para candidatos

TESIS CON
FALLA DE ORIGEN

4.4 Pruebas e Integración del Sistema.

Se debe hacer una distinción técnica entre las pruebas del software completo (pruebas de verificación de requerimientos) y las pruebas del sistema que incorpora el software (pruebas del sistema o pruebas de validación del sistema). El objetivo de las primeras es buscar y encontrar discrepancias entre los requerimientos y la ejecución del software. Tomando en cuenta que: **Realizar un proceso de pruebas es la actividad dedicada a encontrar posibles defectos de un producto, no es determinar que un producto funcione.**

El proceso de verificación de los requerimientos comienza con el análisis de esos requerimientos y una inspección en la cual se busca evaluar la consistencia, conclusión y factibilidad de los requerimientos, tanto individualmente como en conjunto. Adicionalmente, los requerimientos deben ser revisados y validados por los distintos actores involucrados con el sistema (**stakeholders**), acción que debe aclarar los *compromisos* al respecto, tanto en el sentido de **trade-offs** (prioridades y balance) entre requerimientos como en el sentido de los compromisos que asumen los actores. Es conveniente especificar claramente qué se va a hacer para determinar que el sistema satisface sus requerimientos.

Estas especificaciones son cruciales a la hora de diseñar las pruebas de verificación. El diseño de estas pruebas requiere los siguientes pasos:

- Revisar la capacidad de verificación del requerimiento.
- Especificar el criterio de verificación.
- Hacer visibles las propiedades o elementos del software necesarios para verificar el cumplimiento del requerimiento.
- Controlar los elementos del software necesarios para llevar a cabo las pruebas.
- Elaborar el plan de pruebas.
- Ejecutar el plan de pruebas y reportar sus resultados.

4.4.1 Los procesos del desarrollo como fuente de pruebas.

Los procesos o diagramas de desarrollo son una rica fuente de pruebas para el software desarrollado. En particular, podemos (y debemos) crear pruebas de verificación de los casos de uso, pruebas que busquen errores en el manejo de las restricciones de integridad de las clases del sistema, pruebas de verificación de contratos y pruebas de verificación de diagramas de colaboración.

4.4.2 El plan de pruebas.

El estándar IEEE 829-1983²² (Tabla. 4.4.1) describe los tipos de documentos que pueden producirse durante el proceso de prueba. Puede resultar interesante comparar nuestra propuesta con el estándar.

Proyecto X	IEEE 829-1983
Plan de pruebas de un requerimiento	Plan de prueba
(opcional) Tabla de decisiones asociada al requerimiento	
Análisis de verificabilidad del requerimiento	
Criterio de verificación	
Requisitos de observabilidad	
Requisitos de controlabilidad	
Pistas	
Catálogo de modelos de defectos	
Requerimientos de prueba	Especificación de los requerimientos para el diseño de los casos de prueba
Suite de prueba	
Caso de prueba	Caso de prueba
	Descripción del procedimiento de prueba
	Descripción del elemento a probar

²² ANSI/IEEE Standard 829-1983 for Software Test Documentation. IEEE Press, 1983. Este estándar se incluye en la colección *Software Engineering Standards* publicado por IEEE Press.

Ambiente de prueba	
Bitácora de pruebas	Bitácora de pruebas
	Reporte de incidentes de prueba
Análisis de resultados y acciones recomendadas	
Resumen gerencial del proceso	Resumen de pruebas

Tabla 4.4.1 Estándar IEEE 829-1983

El propósito del plan de pruebas es explicar el alcance, enfoque, recursos requeridos, calendario, responsables y manejo de riesgos de un proceso de pruebas.

Puede haber un plan global que explique el énfasis a realizar sobre los distintos tipos de pruebas (verificación e integración).

Un plan de pruebas incluye:

- Identificador del plan.- Preferiblemente de alguna forma mnemónica que permita relacionarlo con su alcance, por ej. TP-Global (plan global del proceso de pruebas), TP-Req-Seguridad1 (plan de verificación del requerimiento 1 de seguridad), TP-Contr-X (plan de verificación del contrato asociado al evento de sistema X), TP-Unit-Despachador.iniciar (plan de prueba unitario para el método iniciar de la clase Despachador). Como todo proceso de desarrollo, está sujeto a control de configuración, por lo que debe distinguirse adicionalmente la versión y fecha del plan.
- Alcance.- Indica el tipo de prueba y las propiedades/elementos del software a ser probado.
- Elementos a probar.- Indica la configuración a probar y las condiciones mínimas que debe cumplir para comenzar a aplicarle el plan. Por un lado, es difícil y riesgoso probar una configuración que aún reporta fallas; por otro lado, si esperamos a que todos los módulos estén perfectos, puede que detectemos fallas graves demasiado tarde.

- Estrategia.- Describe la técnica, patrón y/o herramientas a utilizarse en el diseño de los casos de prueba. Por ejemplo, en el caso de pruebas unitarias de un procedimiento, esta sección podría indicar: "Se aplicará la estrategia caja-negra de fronteras de la precondición" o "Ejercicio de los caminos ciclomáticos válidos". En lo posible la estrategia debe precisar el número mínimo de casos de prueba a diseñar, por ej. 100% de las fronteras, 60% de los caminos ciclomáticos... La estrategia también hace explícito el grado de automatización que se exigirá, tanto para la generación de casos de prueba como para su ejecución.
- Categorización de la configuración.- Explica las condiciones bajo las cuales, el plan debe ser:
 - Suspendido,
 - Repetido y/o
 - Culminado.

En algunas circunstancias el proceso de prueba debe suspenderse en vista de los defectos o fallas que se han detectado. Al corregirse los defectos, el proceso de prueba previsto por el plan puede continuar, pero debe explicarse a partir de qué punto, ya que puede ser necesario repetir algunas pruebas. Los criterios de culminación pueden ser tan simples como aprobar el número mínimo de casos de prueba diseñados o tan complejo como tomar en cuenta no sólo el número mínimo, sino también el tiempo previsto para las pruebas y la tasa de detección de fallas.

- Tangibles.- Explica los documentos a entregarse al culminar el proceso previsto por el plan p. ej. subplanes, especificación de pruebas, casos de prueba, resumen gerencial del proceso y bitácora de pruebas.
- Procedimientos especiales.- Identifica el grafo de las tareas necesarias para preparar y ejecutar las pruebas, así como cualquier habilidad especial que se requiere.
- Recursos.- Especifica las propiedades necesarias y deseables del ambiente de pruebas, incluyendo las características del hardware, el software de

sistemas (p. ej. el sistema de operación), cualquier otro software necesario para llevar a cabo las pruebas, así como la colocación específica del software a probar (p. ej. qué módulos se colocan en qué máquinas de una red local) y la configuración del software de apoyo.

La sección incluye un estimado de los recursos humanos necesarios para el proceso. También se indica cualquier requerimiento especial del proceso: actualización de licencias, espacio de oficina, tiempo en la máquina de producción, seguridad, etc.

- Calendario.- Esta sección describe los hitos del proceso de prueba y el grafo de dependencia en el tiempo de las tareas a realizar.
- Manejo de riesgos.- Explica los riesgos del plan, las acciones mitigantes y de contingencia.
- Responsables.- Especifica quién es el responsable de cada una de las tareas previstas en el plan.

4.4.3 Pruebas basadas en autómatas.

UML (**Unified Modeling Language** – Lenguaje de modelado unificado) incluye el uso de autómatas o diagramas de estado. Los autómatas pueden utilizarse para describir comportamientos que dependen del estado en que se encuentra el sistema modelado. En particular en el desarrollo de un sistema orientado por objetos, un autómata puede utilizarse para describir o complementar:

- El modelo de casos de uso. En este modelo se observan situaciones donde el flujo de opciones es más complejo que cursos alternos de acciones.
- El comportamiento potencial de cada clase. En este sentido se elabora un autómata por clase que describe los estados por los que pueden pasar los objetos de una clase y las transiciones entre esos estados.
- El comportamiento de un agrupamiento de clases, subsistema o sistema de objetos. En UML tal descripción recibe el nombre de Modelo Dinámico y consta de un conjunto de autómatas anidados que pueden enviar y recibir

eventos entre si. El Modelo Dinámico es crucial en sistemas de control en tiempo real.

Un autómatas está bien formado si:

- No tiene estados equivalentes (redundantes).
- No tiene transiciones ϵ (transiciones bajo el evento nulo).
- No tiene estados muertos (estados que no son finales y no tienen transiciones de salida).
- Ciclos infinitos (al caer en un estado de un ciclo infinito no es posible salir de los estados del lazo infinito y además ninguno de esos estados es un estado final).

4.4.3.1 Modelos de defectos asociados a autómatas.

El diseño e implementación de autómatas es propenso a los siguientes defectos:

- Un estado de más, faltante o corrupto.
- Una transición incorrecta u omitida.
- Un evento incorrecto u omitido de una transición.
- Una acción incorrecta u omitida de una transición.
- Una guarnición incorrecta u omitida de una transición.
- Faltan o sobran estados iniciales.
- Faltan o sobran estados finales.

Adicionalmente están los defectos asociados a que una subclase no respeta el modelo dinámico de su superclase.

4.4.4 Modelo de casos de uso como fuente de pruebas.

Cada caso de uso describe una tarea que debe llevar a cabo un actor; por lo que pueden generarse casos de prueba para validar que las tareas se llevan a cabo según lo indica el caso de uso. Adicionalmente el caso de uso describe un flujo de

control que describe qué pasos pueden repetirse, cuáles son opcionales, qué cursos alternos hay, flujos que deben probarse:

- ✓ Casos correspondientes al curso normal del caso.
- ✓ Casos correspondientes a cursos excepcionales.
- ✓ Casos que surgen de requerimientos específicos a un elemento del caso de uso.
- ✓ Casos asociados a la prueba de características descritas en documentos asociados al caso de uso.

4.4.5 Modelo de Clases como fuente de pruebas.

Los defectos a los que está propensa la implementación de un modelo de clases incluyen:

- Errores en la multiplicidad de la asociación.
- Falta o sobra una asociación.
- Actualización anómala, típicamente en la actualización de información replicada.
- Eliminación anómala, como por ejemplo:
 - En la eliminación de un objeto que ayuda a definir una entidad débil.
 - En la eliminación de un objeto que pertenece a una asociación 1 a N (por el lado del 1).
 - En el manejo de nulificaciones (*nullifies*), cascadas y restricciones;
- Asociaciones incorrectas entre objetos.
- Inserciones incorrectas.
- En general, no se satisfacen las restricciones de integridad.

4.4.6 Contratos como fuente de pruebas.

Pueden aplicarse los modelos de defectos pertinentes sobre las precondiciones y postcondiciones de los contratos. La visibilidad de las salidas, tomando en cuenta

que las salidas de un contrato especifican qué objetos y asociaciones se crean, modifican o eliminan puede requerir escribir código especial para las pruebas.

4.4.7 Diagramas de colaboración como fuente de pruebas.

La implementación de un evento de sistema debe cumplir con su diagrama de colaboración. Para revisar si efectivamente cumple, debería ejecutarse la implementación, revisando que los mensajes enviados corresponden a lo indicado por el diagrama de colaboración. Idealmente se deberían determinar las secuencias de mensajes esperados y luego comparar estas secuencias contra las trazas o secuencias generadas por la implementación. Cabe mencionar que hay herramientas que generan estas trazas, indicando no sólo el mensaje sino el objeto que envía y el objeto que recibe.

Generar manualmente las trazas esperadas a partir de los diagramas de colaboración, no necesariamente es factible. El problema radica en que los diagramas de colaboración no especifican cómo cambia el estado de un objeto al recibir un mensaje m . Por ende, si ese objeto envía mensajes $m1$, $m2$ como parte de la reacción a m , pero sólo si se satisface una guarnición que depende del estado del objeto, es muy probable que sea imposible determinar el estado: en consecuencia no es posible determinar si $m1$, $m2$ deben agregarse a la traza esperada. Este problema sólo se resuelve si se realiza alguna de las siguientes acciones:

- Se especifican los cambios al estado ocurridos como reacción a la recepción de un mensaje y previos a cada envío de un mensaje de reacción. Adicionalmente hace falta especificar los parámetros y respuestas asociadas a los mensajes, ya que las guarniciones pueden depender de sus valores.
- Se puedan ejecutar los diagramas de colaboración a la par que la implementación, tomando valores de la implementación cuando sean necesarios.

Hasta ahora se ha sugerido generar casos de prueba a partir de la estructura de los parámetros del evento del sistema, es decir la generación de los casos utilizan una estrategia *caja negra* respecto al diagrama de colaboración. Sin embargo, el diagrama de colaboración puede verse como un diagrama de flujo de alto nivel, donde las guarniciones corresponden a puntos de decisión. Por ende podemos generar casos de prueba *caja blanca* para lograr una cierta *cobertura* del grafo asociado al diagrama de colaboración.

Plan de pruebas para el sistema de Administración de Recursos Humanos (SIAREH).

Se tienen identificados 23 casos de uso, de los cuales se utilizarán 6 para ejemplificar la ejecución del plan de pruebas utilizando los procesos de UML como fuentes de prueba (diagramas y tablas de casos de uso, diagramas de estados, diagramas de clases y contratos (tarjetas CRC)). La tabla 4.4.2 muestra el documento de plan de pruebas para todo el sistema con el mnemónico TP_SIAREH_v.0.0.²³

Proyecto SIAREH	Plan de pruebas – TP_SIAREH_v.0.0
Alcance	El plan de pruebas debe verificar que el modelo de análisis y los procesos UML que de él se desprenden para el Sistema de Administración de REcursos Humanos cumplen con los requerimientos de usuario.
Elementos a probar	(3 Diagramas de casos de uso, 3 diagramas de estados, 1 Diagramas de clases y 23 contratos)
Estrategia	Se utilizará la estrategia de caja negra para los casos de uso y diagramas de estados. Para el diagrama de clases y los contratos la estrategia será la prueba de caja blanca.
Tangibles	Debe entregarse un documento con el resultado de las pruebas dentro de tablas que contengan información concerniente a cada caso de prueba, con requerimientos, elementos probados, datos E/S, defectos encontrados, posible solución, nivel de tolerancia para el fallo, prioridad, fecha y hora de prueba, responsables, comentarios, etc.

²³ TP_SIAREH_v.0.0 – Test Plan Sistema de Administración de Recursos Humanos Versión 0.0

Procedimientos especiales	Las pruebas deben basarse en los diagramas realizados en Rational Rose por lo que el usuario que tenga la función de tester (verificador) debe tener conocimiento de la metodología RUP así como de la nomenclatura de Rational Rose.
Recursos	Se requiere un equipo de dos personas como mínimo para realizar las pruebas de la aplicación.
Calendario	Se muestra el diagrama de Gantt que ilustra el plan general de desarrollo de este sistema y que está en el punto 4.5 de este trabajo.
Responsables	Alberto Z. Rodríguez, Iván Márquez

Tabla 4.4.2 Plan de pruebas para el Sistema de Administración de Recursos Humanos (SIAREH).

Las figuras 4.4.1 y 4.4.2 muestran, respectivamente los diagramas de casos de uso y de estados que el tester debe tomar en cuenta para las pruebas a realizar.

Diagrama de casos de uso para usuarios



Fig. 4.4.1 Diagrama de casos de uso (usuario) derivado del modelo de análisis para el SIAREH.

Diagrama de estados para usuario nuevo y existente

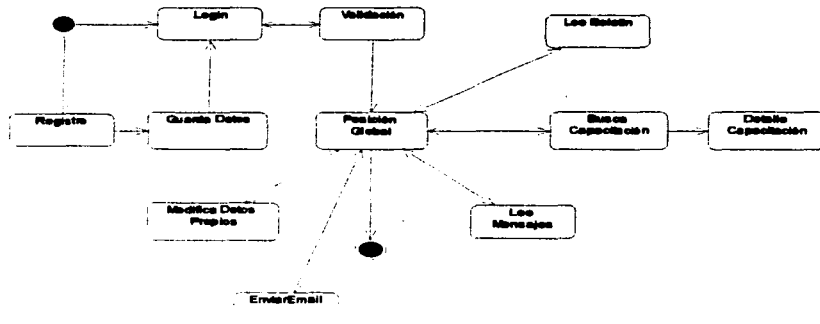


Fig. 4.4.2 Diagrama de estados derivado del modelo de análisis para el SIAREH.

La tabla 4.4.3 muestra el detalle para cinco casos de prueba de acceso al sistema que engloban el mismo número de casos de uso: Login, Registro, Validación, Bloqueo y Posición Global del usuario independientemente de su nivel.

Proyecto SIAREH	Caso de prueba TP_Acc_Usuario
Requerimiento(s)	<ul style="list-style-type: none"> * El acceso a la aplicación debe ser controlado por un nombre de usuario y contraseña. * El nombre del usuario y su contraseña deben ser únicos. No puede haber dos usuarios del sistema compartiendo cualquiera de estos dos valores. Siempre que se genere un nombre de usuario o contraseña nuevos, deberá validarse que no existan en la base de datos. De existir tales valores, se solicitará al usuario que defina un nombre de usuario o contraseña diferentes. * Los mensajes de error al no poder ingresar al sistema no deben especificar en qué se equivocó el usuario, deben ser

genéricos. Ej. "Datos erróneos, verifique su nombre de usuario y/o contraseña e intente nuevamente".

* Si el usuario falla en su intento por ingresar a la aplicación más de tres veces, es decir, proporciona una combinación de nombre de usuario y/o contraseña inválidos, su cuenta se deberá desactivar, debiendo notificar esta situación al administrador para que dicha cuenta pueda ser reactivada. El valor de intentos fallidos debe ser paramétrico.

* La contraseña debe poder ser modificada únicamente por el usuario dueño de la misma. En caso de olvido o pérdida, se deberá notificar al administrador para que éste asigne una nueva contraseña temporal, que deberá ser modificada por el usuario en cuestión a la brevedad posible.

* La aplicación debe mostrar en su página principal, dos opciones: La primera, para que los usuarios existentes puedan ingresar a su cuenta, y los usuarios que no estén registrados en el sistema, puedan hacerlo en línea utilizando la sección "Registro de usuarios nuevos".

* Una vez que el usuario ha sido registrado podrá ingresar al sistema.

* La posición global de la aplicación debe mostrar el contenido de la misma de acuerdo al nivel del usuario y, además, debe ser capaz de reconocer al usuario que se ha conectado.

LoginUsuario, RegistroUsuario, ValidaUsuario, BloqueoUsuarios, PosiciónGlobal (DCU_Usuario)

Edos_Usuario

Login Usuario: Nombre de usuario y contraseña

Registro Usuario: Datos personales, datos académicos, experiencia laboral, nombre de usuario y contraseña

Valida Usuario: Nombre de usuario y contraseña (proporcionados por el CU Login Usuario – transparentes al usuario)

Bloqueo Usuario: Número de intentos fallidos (proporcionados por el CU ValidaUsuario – transparente al usuario)

Posición Global: Nombre del usuario y nivel de acceso (proporcionados por el CU Valida Usuario – transparentes al usuario)

Login Usuario: Si el nombre de usuario y contraseña son válidos no habrá datos de salida y enviará al usuario a su pantalla principal (Posición Global) o el contrario, si éstos son inválidos, debe visualizarse un mensaje que diga: "**Datos erróneos, verifique su nombre de usuario y/o contraseña e intente nuevamente**". Si alguno(s) de los campos no

Caso(s) de uso

Diagrama de estados

Datos de entrada

Datos de salida

TESIS CON FALLA DE ORIGEN

lleva(n) información, debe observarse una validación de tipo Alert indicando esta situación. Si alguno(s) de los campos lleva caracteres inválidos debe observarse una validación de tipo Alert.

Registro usuario: Si los datos ingresados por el usuario son correctos se debe observar el mensaje de confirmación: *"El proceso de registro ha concluido con éxito. Utilice su nombre de usuario y contraseña para ingresar al sistema"*. Si alguno(s) de los campos no lleva(n) información, debe observarse una validación de tipo Alert indicando esta situación. Si alguno(s) de los campos lleva caracteres inválidos debe observarse una validación de tipo Alert.

Valida Usuario: No tiene datos de salida visibles al usuario. Si la validación es exitosa, se remite al caso de prueba Posición Global. En caso contrario regresa al caso de prueba Login Usuario e incrementa el contador de intentos. Al sobrepasar tres intentos fallidos se debe guardar un 1 en el campo Nivel de la tabla Personas, que será el indicador de bloqueo de cuenta.

Bloqueo Usuario: En caso de sobrepasar tres intentos para ingresar, el mensaje de salida debe decir: *"Por motivos de seguridad, su cuenta ha sido desactivada. Póngase en contacto con el administrador del sistema"*.

Posición Global: Debe mostrar un mensaje de bienvenida con el nombre y primer apellido del usuario que ha ingresado exitosamente al sistema. Se deben desplegar las siguientes opciones en el menú principal: **Modificación de datos propios, Enviar email con comentarios, Leer mensajes, Búsqueda de cursos de capacitación** y, como subínciso de éste último **Detalle de curso de capacitación**. Debe mostrarse, en caso de haber, el boletín como un banner que se despliega hacia arriba.

Al sobrepasar los tres intentos fallidos en el caso de prueba Login Usuario, la cuenta no se desactiva. Se puede seguir intentando sucesivamente.

Debe leerse el parámetro de intentos fallidos de la base de datos y tomarlo en cuenta para compararlo con el contador del caso de prueba **Valida Usuario** o debe verificarse si se está insertando correctamente un 1 en la tabla de Personas en el campo Nivel.

Baja. Este caso de prueba es elemental para cumplir con los mecanismos básicos de seguridad y debe resolverse cuanto antes. No se está cumpliendo con el requerimiento de desactivar la cuenta al superar cierto número de intentos por ingresar al sistema.

Defecto encontrado

Posible solución

Nivel de tolerancia para la falla

Prioridad	0 – Cero. Esta incidencia tiene la más alta prioridad
Fecha y hora de prueba	Lunes 09 de junio de 2003 – 16:30 hs
Revisado por	Alberto Rodríguez y Niegolewska
Comentarios	<p>Todos los casos de prueba fueron exitosos con excepción del bloqueo de cuenta que es parte del caso de prueba Login Usuario. Se pudo observar que las validaciones correspondientes a JavaScript dentro de los casos de prueba: Login Usuario y Registro Usuario existen y funcionan correctamente. La validación del usuario se realiza de igual forma verificando la existencia de éste en la base de datos. Las opciones desplegadas en la posición global (home) del usuario una vez que se firmó, se validó y logró ingresar son las correctas. Se debe comenzar a trabajar cuanto antes, en la resolución del bloqueo de usuarios debido a que representa una falla de seguridad en el sistema. Un ataque cibernético podría llevarse a cabo utilizando un programa que pruebe millones de combinaciones de letras y números tanto en el nombre de usuario como en la contraseña para intentar ingresar en el sistema de forma ilegal. El bloqueo de usuarios es la forma de evitar dicha situación.</p>

Tabla 4.4.3 Caso de prueba para el Login, registro, validación y posición global del usuario.

La tabla 4.4.4 muestra el detalle para el caso de prueba de modificación de datos propios de todos los usuarios del sistema.

Proyecto SIAREH	Caso de prueba TP_Mod_Dat_Prop
Requerimiento(s)	<ul style="list-style-type: none"> El usuario que originalmente solicitó algún tipo de operación, como pueden ser modificaciones en sus datos privados por ejemplo, recibirá la notificación de la decisión del operador en su propia posición global y la información habrá sido afectada siempre y cuando la decisión haya sido aprobada por el operador. Si éste último toma la decisión de rechazar el cambio, simplemente no se afectarán los datos.
Caso(s) de uso	ModificacionDatosPropios
Diagrama de estados	Edos_Usuario, Edos_Admin, Edos_Operador
Datos de entrada	Cualquier campo que incluya los formularios de información personal, académica, de experiencia profesional y/o de cursos de capacitación.

TESIS CON
 FALLA DE ORIGEN

Datos de salida	<p>Deberá visualizarse una pantalla que exhiba los datos nuevos que deberán modificarse en el(los) registro(s) de la base de datos junto con tres botones que deben tener las etiquetas: Corregir, Enviar. Si se pulsa el botón Corregir, el sistema deberá enviar al usuario a la pantalla de edición anterior. Si pulsa Enviar, deberá enviarlo a una pantalla que confirme que la operación fue exitosa, o a una pantalla de error si la operación terminó en alguna excepción controlada. La pantalla de operación exitosa debe desplegar el mensaje: <i>Su operación ha sido enviada al operador exitosamente.</i></p> <p>NOTA: Para que el cambio surta efecto en la base de datos el operador deberá aprobar la operación solicitada.</p>
Defecto encontrado	Ninguno
Posible solución	N/A
Nivel de tolerancia para la falla	N/A
Prioridad	N/A
Fecha y hora de prueba	Lunes 09 de junio de 2003 – 17:00 hs
Revisado por	Alberto Rodríguez y Niegolewska
Comentarios	El caso de prueba no presentó fallas. Se comprobó que las pantallas desplegadas fueron las esperadas y tampoco se encontraron casos en los que el sistema haya caído dentro de una excepción controlada.

Tabla 4.4.4 Caso de prueba para la modificación de datos propios del usuario.

4.5 Factibilidad Técnica y Operativa.

4.5.1 Recursos de hardware.

A continuación se describen los recursos de hardware necesarios para que las herramientas de software que SIAREH utilizará para su funcionamiento trabajen adecuadamente:

Requerimientos de hardware para Linux.

Los requerimientos de hardware para Linux son equivalentes para el servidor de web Apache, el lenguaje PHP y el manejador de bases de datos PostgreSQL, ya que éstos están contenidos en la distribución del sistema operativo y todas las herramientas estarán conviviendo físicamente en el mismo servidor.

Actualmente Linux soporta sistemas con una CPU Intel Pentium II hasta Pentium IV. Los microprocesadores que no son de manufactura Intel, como AMD y Cyrix también funcionan con Linux.

La **motherboard** (placa base) debe ser de arquitectura Pentium 4 FCPGA 478 o AMD Athlon Socket A, que son las motherboard que se pueden conseguir en el mercado actualmente.

Requisitos de memoria.

Linux, comparado con otros sistemas operativos avanzados, necesita relativamente poca memoria para funcionar sin embargo, es altamente recomendable tener 128 megabytes en memoria RAM. Cuanta más memoria se tenga, mejor será el desempeño del sistema. Linux soporta el rango completo de direcciones de 128-bits de los procesadores pentium 4; es decir, utilizará toda la memoria RAM de forma automática, y se aprovechará sin problemas, incluyendo aplicaciones que requieren más memoria como X-Window, Emacs, y demás. Sin embargo, disponer de más memoria es casi tan importante como tener un

procesador más rápido. 256 megabytes o más pueden ser necesarios si se presenta gran demanda del servicio en el sistema por parte de los usuarios.

Debe reservarse una parte del disco duro para **swapping** (espacio de intercambio) que se usa como RAM virtual. El área de "swap" no puede reemplazar a una memoria física RAM real, pero puede permitir al sistema ejecutar aplicaciones más grandes guardando en disco duro aquellas partes de código que están inactivas.

Requisitos de la controladora de disco duro.

Linux soporta todos los controladores de disco duro IDE y algunas SCSI.

La regla general para controladores que no sean SCSI, es que si pueden acceder a las unidades (disco y/o disquete) desde Windows u otro sistema operativo, deberían poder hacerlo desde Linux.

Linux también soporta un número de controladoras de disco SCSI, si bien el soporte para SCSI es más limitado a causa de la gran cantidad de estándares que existen para la interfaz de las controladoras por lo que se recomienda la utilización de una controladora genérica.

Requisitos de espacio en disco.

La cantidad de espacio en disco duro que se necesita depende en gran medida de las necesidades y de la cantidad de software que se va a instalar. Linux es relativamente pequeño con relación a las implementaciones de UNIX; se podría ejecutar un sistema completo con 100-200 megabytes de espacio en disco. Sin embargo, si se quiere disponer de espacio para expansiones y para paquetes más grandes como X Window, se necesita más espacio en disco duro. Se puede considerar un sistema grande con sitio para muchos usuarios y espacio para futuras expansiones en un rango de 1-1.5 gigabytes.

Requisitos de monitor y adaptador de video.

Linux soporta todas las tarjetas de video estándar. En general, si la combinación que se tiene de monitor y tarjeta de video funcionan bajo otro sistema operativo como Windows, debería funcionar perfectamente con Linux.

Hardware diverso.

Las secciones de arriba describen qué hardware es necesario para correr un sistema Linux. Sin embargo, se puede tener cierto número de dispositivos "opcionales" como almacenamiento en cintas (para respaldos de la base de datos), CD-ROM, tarjetas de sonido y demás.

Ratones y otros dispositivos apuntadores.

Normalmente sólo se usará un ratón bajo un entorno gráfico como el Sistema X Windows. Sin embargo, algunas aplicaciones Linux no asociadas con un entorno gráfico, hacen uso del ratón.

Linux soporta todos los ratones serie estándar.

Todos los demás dispositivos apuntadores, como los "trackball" que emulen a los dispositivos de arriba, también deberían funcionar.

Almacenamiento en CD-ROM.

Casi todas las unidades de CD-ROM usan el interfaz SCSI17. Siempre y cuando tenga un adaptador SCSI soportado por Linux, la unidad de CD-ROM debería funcionar.

Unidades de cinta.

Hay varios tipos de unidades de cinta disponibles en el mercado. La mayoría usan una interfaz SCSI, por lo que estarían soportadas por Linux.

Existen controladores en fase de desarrollo para varios otros dispositivos de cinta como las unidades Colorado que se conectan a la controladora de disquetes.

Impresoras.

Linux soporta todas las impresoras paralelo y, recientemente, USB. Si se puede acceder a la impresora por el puerto paralelo o USB desde Windows, u otro sistema operativo, debería poderse acceder a él desde Linux también. El software de impresión de Linux consiste en el software estándar de UNIX lp y lpr. Este software también permite imprimir remotamente a través de la red.

Módems.

Igual que para las impresoras, Linux soporta toda la gama de modems serie, tanto internos como externos. Hay una gran cantidad de software de telecomunicaciones disponible para Linux, incluyendo Kermit, pcomm, minicom, Seyon.

Tarjetas Ethernet.

Linux soporta un buen número de tarjetas Ethernet y adaptadores para LAN18. Las tarjetas genéricas están soportadas.

4.5.2 Recursos de software.

Instalación paso a paso de las herramientas de software requeridas.

Obtención e instalación de Linux.

Linux, al ser un software de libre distribución, no tiene organización o entidad alguna responsable de mantenerlo y distribuirlo. Por lo tanto, cualquiera es libre de agrupar y distribuir el software, mientras respete las restricciones de la GPL²⁴. El

²⁴ GPL – GNU General Public License (Licencia Pública General)

resultado final de esto es que existen muchas distribuciones de Linux, disponibles a través de FTP²⁵ anónimo o pidiéndolo por correo.

Si se cuenta con acceso a Internet, el modo más fácil de obtener Linux es a través de FTP anónimo. Por ejemplo, un servidor de archivos por FTP que posee software de Linux es sunsite.unc.edu, y se pueden encontrar varias distribuciones de Linux en el directorio: /pub/Linux/distributions

Muchas distribuciones se crean en servidores de FTP anónimo como un conjunto de imágenes de disco. Es decir, la distribución consiste en un conjunto de archivos, y cada archivo contiene la imagen binaria de un diskette. Para poder copiar el contenido de un archivo imagen en el diskette, se puede usar el programa RAWRITE.EXE bajo MS-DOS. Este programa copia, bloque a bloque, el contenido de un archivo a un diskette, sin preocuparse por el formato del disco. RAWRITE.EXE puede ser encontrado en varios servidores de FTP de Linux, como por ejemplo en uno de los sistemas que mayor soporte brindan a Linux: sunsite.unc.edu en el directorio: /pub/Linux/system/Install/rawwrite

Por lo tanto, en muchos casos, simplemente puede transferirse el conjunto de imágenes de diskette, y usar RAWRITE.EXE con cada imagen de modo que cree un conjunto de diskettes. Se arranca desde el llamado "boot diskette" y se estará listo para comenzar. El software se suele instalar directamente desde los diskettes, aunque algunas distribuciones permiten instalarlo desde una partición MS-DOS del disco duro. Algunas distribuciones permiten instalar desde una red TCP/IP.

Otras distribuciones de Linux se instalan desde un conjunto de diskettes con formato MS-DOS. Por ejemplo, la distribución Slackware de Linux sólo necesita que se creen los disquetes boot y root usando RAWRITE.EXE. El resto de diskettes se copian a diskettes con formato MS-DOS usando el comando MS-DOS COPY. El sistema instala el software directamente desde los diskettes MS-DOS.

²⁵ FTP – File Transfer Protocol (Protocolo de transferencia de archivos)

Esto ahorra el problema de tener que usar RAWRITE.EXE con muchos archivos imagen, aunque, por otro lado, se requiere contar con acceso a un sistema MS-DOS para instalar el sistema.

Cuando se transfiere el software de Linux, se debe usar el modo binario para todas sus transferencias de archivos (con la mayoría de clientes FTP, el comando "binary" activa este modo).

Una vez que se ha obtenido una distribución de Linux, estamos preparados para instalar el sistema.

A pesar de que cada distribución de Linux es diferente, el método utilizado para instalar el software es, en general, como sigue:

- Partición del(los) disco(s) duro(s). Si se tienen instalados otros sistemas operativos, se necesitará reparticionar los discos con el fin de reservar espacio para Linux.
- Arranque de la instalación de Linux. Cada distribución de Linux incluye algo para arrancar inicialmente e instalar el software, usualmente un diskette de arranque. Arrancando de esta forma, entraremos en un programa de instalación para el resto del software, o bien se permitirá seguir instalándolo manualmente.
- Creación de las particiones para Linux. Después de reparticionar el disco para reservar espacio para Linux, se deben crear particiones de Linux en dicho espacio. Esto se realiza con el programa fdisk.
- Creación de los sistemas de archivos y el espacio de intercambio. En este momento, se debe crear uno o más sistemas de archivos, utilizados para guardar los archivos, en las particiones recién creadas. Además, si se va a utilizar espacio de intercambio ("swap"), se debe crear dicho espacio en una de las particiones para Linux.

- **Instalar los programas en los sistemas de archivos.** Finalmente, se debe instalar el software en los nuevos sistemas de archivos. Después de esto, lo que queda es fácil, si todo va bien.
- **Cada distribución tiene una forma distinta de instalar Linux.** Muchas tienen un programa que guía al usuario paso a paso en este proceso. En otras, se tiene que montar los sistemas de archivos en un directorio (como /tmp) y copiar el software a este a mano. En las distribuciones en CD-ROM se puede seguir la opción de instalar una parte de lo que contiene el disco duro y dejar el resto (la mayor parte) en el CD-ROM. Algunas distribuciones ofrecen diversos mecanismos para instalar el software. Por ejemplo, se puede instalarlo directamente desde una partición MS-DOS de el disco duro, en lugar de hacerlo desde diskettes. O incluso se puede hacerlo a través de una red TCP/IP mediante FTP o NFS. Por ejemplo, la distribución Slackware solo necesita que se creen las particiones con fdisk, y el espacio de intercambio con mkswap y swapon (si se tiene 4 megabytes o menos de RAM), y a continuación se ejecuta el programa setup, que guía al usuario mediante un menú bastante autoexplicativo en la instalación del software.

Obtención e instalación de Apache, PHP y PostgreSQL.

El primer paso para obtener el software de Apache, PHP y PostgreSQL (en algunas distribuciones recientes de Linux ya vienen incluidos) es acceder a las direcciones:

- www.apache.org
- www.php.org
- www.postgresql.org

y descargar los paquetes de software correspondientes. A continuación debe crearse el siguiente directorio en el servidor de Linux:

/usr/local/era
**TESIS CON
FALLA DE ORIGEN**

donde deben copiarse los paquetes que hemos descargado y que deben ser:

- apache_1.3.12.tar.gz
- php-4.0.2.tar.gz
- postgresql-7.0.2.base.tar.gz
- postgresql-7.0.2.tar.gz
- postgresql-7.0.2.test.tar.gz

Una vez copiados los paquetes de software, éstos deben ser descomprimidos de la siguiente forma:

```
for i in *.gz; do tar -zxf $i; done
```

Si la instrucción anterior no funciona debe intentarse la extracción manualmente:

```
tar -zxf apache_1.3.12.tar.gz
```

```
tar -zxf php-4.0.2.tar.gz
```

```
tar -zxf postgresql-7.0.2.base.tar.gz
```

```
tar -zxf postgresql-7.0.2.tar.gz
```

```
tar -zxf postgresql-7.0.2.test.tar.gz
```

Posteriormente deben crearse los usuarios: **www** y **postgres**.

Debido a que PHP se va a integrar con PostgreSQL es necesario contar con ciertas bibliotecas cuando se compile PHP, por lo que la instalación de PostgreSQL debe llevarse a cabo antes que PHP. Para instalar y configurar PostgreSQL debemos acceder al directorio de PostgreSQL:

```
cd /usr/local/src/postgresql-7.0.2/
```

Una vez dentro del directorio se ejecutan los comandos de configuración e instalación:

```
./configure; make; make install
```

Se debe agregar al final del archivo /etc/ld.so.conf la siguiente línea:

```
/usr/local/pgsql/lib
```

y ejecutar el enlazador dinámico:

```
ldconfig -v
```

El siguiente paso es configurar el servidor de web Apache para que PHP pueda ser configurado posteriormente. De modo que debemos desplazarnos al directorio:

```
cd /usr/local/src/apache_1.3.12
```

Y configurar el Apache en el directorio donde será instalado:

```
./configure -prefix=/usr/local/apache
```

Ya que el Apache ha sido configurado para PHP, es hora de configurar PHP, para tal efecto, tenemos que desplazarnos al directorio

```
cd /usr/local/src/php-4.0.2
```

y se configura PHP para que trabaje con Apache y PostgreSQL:

```
./configure -with-apache=/usr/local/src/apache_1.3.12  
--with-pgsql=share  
--enable-track-vars
```

ahora se instala PHP

```
make; make install
```

y se copia el archivo de inicio de PHP:

TESIS CON
FALLA DE ORIGEN

```
cp php.init-dist /usr/local/lib/php.ini
```

Ya que PHP ha sido instalado y configurado exitosamente, debe reconfigurarse el servidor de web Apache para que tome en cuenta a PHP y posteriormente instalarse:

```
./configure --prefix=/usr/local/apache  
--activate-module=src/module/php4/libphp4.a
```

y se instala el servidor de web Apache:

```
make; make install
```

Para decirle al Apache que enlace páginas con PHP se tiene que editar el archivo `/usr/local/apache/conf/httpd.conf` agregando (o quitando comentarios, si es el caso) las siguientes líneas:

```
AddType application /x-httpd-php .php .php3 .html  
AddType application /x-httpd-php-source .phps
```

Ya que PHP, PostgreSQL y Apache han sido instalados, deben conectarse entre sí. El proceso de Apache se ejecuta usualmente como nobody por omisión. Debemos cambiar de nobody a la cuenta que creamos previamente, que es www.

Para que Apache y PostgreSQL puedan comunicarse entre sí debe editarse el archivo `/usr/local/apache/conf/httpd.conf` y debe buscarse la línea `user nobody` y substituirse por `user www`.

Hecho lo anterior, debemos ingresar al sistema firmándonos como el usuario postgres y editar el archivo `/home/postgres.bashrc` (asumiendo que ese es el shell utilizado por omisión). Se deben agregar las siguientes líneas:

```
PGDATA=/home/postgres/database  
PATH = ``$PATH:/usr/local/pgsql/bin``  
Export PGDATA PATH
```

Ahora debe ejecutarse el siguiente comando para asegurar que las variables de ambiente han sido asignadas correctamente.

```
source /home/postgres/.bashrc
```

Y se debe inicializar la base de datos con el comando initdb

Si todo salió bien, se podrán visualizar los siguientes mensajes en la consola:

```
This database system will be initialized with username "postgres".
This user will own all the data files and must also own the server
process.
Creating database system directory /home/postgres/database
Creating database system directory /home/postgres/database/base
Creating database XLOG directory /home/postgres/database/pg_xlog
Creating template database in /home/postgres/database/base/templatel
Creating global relations in /home/postgres/database/base
Adding templatel database to pg_database
Creating view pg_user.
Creating view pg_rules.
Creating view pg_views.
Creating view pg_tables.
Creating view pg_indexes.
Loading pg_description.
Vacuuming database.
Success. You can now start the database server using:

/usr/local/pgsql/bin/postmaster -D /home/postgres/database

or

/usr/local/pgsql/bin/pg_ctl -D /home/postgres/database start
```

TESIS CON
FALLA DE ORIGEN

or

```
/usr/local/pgsql/bin/pg_ctl -D /home/postgres/database start
```

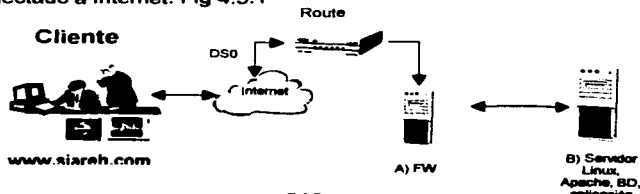
Finalmente para que PHP pueda hacer llamados a la base de datos desde archivos HTML servidos a los browsers por el servidor de web Apache, PostgreSQL debe ser instruido para que utilice la opción `-i`. El proceso de la base de datos de PostgreSQL se tendría que inicializar utilizando:

```
/usr/local/pgsql/bin/postmaster -i -D /home/postgres/database
```

4.5.3 Requisitos para arrancar el sistema.

El SIAREH necesita elementos muy básicos para poder ser implantado y para que comience su utilización vía web. Lo principal es:

- Contar con un servidor que tenga instalado y configurado el software anteriormente descrito
- Que la base de datos esté creada, poblada y que su proceso principal haya sido arrancado y esté listo para escuchar por el puerto correspondiente
- Que el proceso del servidor de web también esté en ejecución y listo para escuchar por el puerto 443 (https)
- Que la aplicación esté terminada y resida en los directorios que se han definido al servidor de web para que éste pueda resolverlos ante una petición
- Que el servidor esté conectado a internet (red con topología de bus) y que tenga un identificador (URL) para que el servidor pueda ser localizado ante una petición hecha desde cualquier equipo cliente que también esté conectado a internet. Fig 4.5.1



4.5.4 Recursos humanos necesarios para la operación del sistema.

El SIAREH es un sistema que requiere contar con un equipo humano que vigile y coadyuve en tareas diversas a todos los usuarios en la utilización del mismo durante su operación. A este equipo humano que brindará soporte a nivel herramientas de software y a nivel aplicativo se le denominará **HelpDesk** y deberá contar con una extensión telefónica específica para recepción de requerimientos. Deberán generarse **tickets** (boletos) de atención al usuario y se les deberá dar seguimiento hasta la resolución del o los requerimientos. El equipo de HelpDesk estará conformado por:

- **Webmaster (Administrador web).**- Quien es el usuario encargado de administrar el servidor de web y la aplicación. Este recurso debe conocer Apache y PHP a detalle. Debe ser capaz de instalar, configurar, levantar y dar de baja los servicios de web; gestionar puntualmente los temas relacionados con el(los) certificado(s) digital(es); rastrear errores en los diversos logs del servidor y encontrar soluciones a dichos errores, debe poder crear y modificar alias dentro del servidor; debe monitorear la actividad del servicio cuando el servidor y la aplicación estén trabajando; debe encargarse de depurar logs, verificar el desempeño y el balanceo de carga, realizar ajustes finos para la óptima operación del equipo; es importante que esta persona cuente con conocimientos de unix y redes a nivel administrativo; el webmaster también se va a encargar de la creación, modificación y/o eliminación de elementos de catálogos a nivel de la aplicación de web. Será un usuario con nivel de administrador y también deberá ser capaz de dar soporte técnico a los usuarios que encuentren dificultades durante la utilización del SIAREH. De modo que el webmaster será clave en el equipo de help desk (soporte de sistemas). El periodo de entrenamiento para un elemento con estas características es de dos semanas, considerando que el candidato cubre el perfil mencionado.
- **DBA (Administrador de bases de datos).**- Es el administrador de la base de datos y es responsable de la creación, modificación y eliminación de

TESIS CON
FALLA DE ORIGEN

elementos como tablas, vistas, índices, procedimientos almacenados, etc; es el único responsable de mantener la integridad de la información; debe ser capaz de instalar y configurar PostgreSQL así como de levantar y dar de baja los servicios de bases de datos en el servidor; vigilar el desempeño de la base de datos y realizar el **tuning** (ajuste fino) cuando sea necesario; es responsable de diseñar y desarrollar mecanismos de respaldos periódicamente y del plan de contingencia, asegurándose de que, en caso de desastre, la información se encuentre bajo resguardo y pueda recuperarse sin pérdidas substanciales (búsqueda y recuperación). Este usuario debe ser experto en la administración de bases de datos relacionales y debe tener un profundo conocimiento de PostgreSQL, es importante que también cuente con experiencia en unix para las tareas administrativas. El periodo de entrenamiento para un elemento con estas características es de una semana, considerando que el candidato cubre el perfil mencionado.

- **SYSOP (System Operator - Operador de sistema).**- Este usuario deberá encargarse de la administración exclusiva del equipo donde esté instalado el sistema operativo Linux. Será el responsable de su instalación y configuración, así como de los cambios que el sistema necesite. Debe ser quien, desde un principio, cree las particiones de disco, los sistemas de archivos, la puesta en marcha del sistema operativo, gestión (creación, modificación y eliminación) de usuarios y establecimiento de niveles de seguridad a nivel sistema operativo; servirá como apoyo para sus compañeros webmaster y DBA en tareas de instalación, configuración y **troubleshooting** (solución de errores) de herramientas ante posibles situaciones de riesgo para la aplicación y para los datos respectivamente. El SYSOP deberá monitorear constantemente el desempeño del servidor y realizará los ajustes necesarios para que el sistema operativo cuente y disponga de todos los recursos que necesita para un funcionamiento óptimo. Asimismo, será el encargado de diseñar esquemas de contingencias en caso de cualquier emergencia ocasionada por desastre. El

perfil del SYSOP es el de un administrador de sistemas unix con conocimientos sólidos de redes. El periodo de entrenamiento para un elemento con estas características es de tres semanas, considerando que el candidato cubre el perfil mencionado.

- **Operador.-** Este usuario, aunque forma parte del HelpDesk, no se considera un elemento que se dedique el cien por ciento de su tiempo a soporte de usuarios. El Operador es el experto en los temas de negocio. Conoce a detalle la aplicación de manera global y es quien establece lineamientos y reglas de negocio. Es el usuario que aprueba o desaprueba cualquier modificación a los catálogos del sistema, resuelve dudas a usuarios finales y de Helpdesk sobre su funcionamiento y los tres tipos de usuarios mencionados anteriormente están bajo sus órdenes. Es deseable que conozca las herramientas de desarrollo aunque no es necesario que las domine. El perfil de este usuario es el de un elemento con alto rango que pertenezca a la subdirección administrativa. Debe estar en contacto directo con la dirección general para tratar los temas relativos al negocio y su mapeo a la aplicación. Este tipo de usuario no debe necesitar de entrenamiento, ya que se supone que debe estar al tanto de todos los temas relacionados con SIAREH desde su concepción hasta su puesta en marcha y operación. Suponiendo que debiera sustituirse se considera que el puesto debe ser captado por alguna persona que pertenezca a la misma subdirección y que debe comprender los temas técnicos sin llegar al detalle en un par de semanas.
- **Usuario.-** Finalmente tenemos al usuario del sistema. Este usuario es el empleado de Capital Humano SA que debe comenzar a llenar sus datos desde que es candidato. Si se le contrata, deberá mantener su información personal y curricular siempre actualizada. Por considerarse que la herramienta fue diseñada y construida con un nivel de amigabilidad muy alto y, por la familiaridad que los usuarios tienen con aplicaciones de internet, se considera que basta con un breve curso de inducción al sistema que no exceda de 2 horas.

4.6 GENERACIÓN DE REPORTES.

Mediante la generación de reportes los operadores pueden realizar consultas a la base de datos para obtener información de interés común para el área de recursos humanos o para alguna otra área específica, la cual hace referencia a los usuarios que están empleados en Capital Humano S.A. como a los candidatos a formar parte de la misma.

Por ejemplo, el encargado del departamento de sistemas requiere de algún recurso humano que pueda tomar el rol de líder de proyecto en un programa de trabajo. Para esto necesita saber algunos datos con los cuales él o las personas encargadas puedan tomar una decisión. Para esto, es necesario que el operador acceda al Sistema de Administración de Recursos Humanos o con una cuenta de operador autorizada por el administrador del Sistema solicitar en el rubro de Reportes de Colaboradores los campos necesarios para su reporte.

Para poder generar este tipo de reportes es necesario acceder al Sistema como operador.

Al momento de ingresar al sistema, se puede entrar a la sección de "Generación de Reportes" a través del menú principal de la página de inicio.

Dentro de la sección "Generación de Reportes", el operador puede seleccionar entre dos rubros para la generación de reportes que son "Reportes de Colaboradores", que contempla toda la base trabajadora o empleados de la empresa en todas las líneas de servicio o departamentos, las cinco áreas y dos subdirecciones.

La otra opción es el rubro de "Reportes de Candidatos", que contempla los usuarios que se han registrado en el sistema en línea pero que todavía no son parte de la empresa.

TESIS CON
FALLA DE ORIGEN

La otra opción es el rubro de "Reportes de Candidatos", que contempla los usuarios que se han registrado en el sistema en línea pero que todavía no son parte de la empresa.

La figura 4.6.1 muestra la entrada a la sección de "Generación de Reportes".



Figura 4.6.1 Sección de Generación de Reportes.

Dependiendo de la opción seleccionada el operador podrá obtener información como es el nombre completo del candidato, sus habilidades generales, sus habilidades técnicas cuando se elija la opción de Reporte de Candidatos. En la figura 2.6.2 se muestra la pantalla para la Generación de Reportes para candidatos.

Reportes para Candidatos. Seleccionamos los Campos deseados:

Si va generar el reporte del candidato, complete los campos siguientes y selecciona la opción deseada.

Nombre del Candidato: _____ Clase del Candidato: _____

Nombre completo del Candidato Habilidades del Candidato Habilidades técnicas

De la siguiente información las opciones deseadas:

Ingresar habilidades y roles desde: Por _____ Candidatos con conocimientos de programación en _____

Candidatos con un diploma en TOCOP Candidatos que han sido libres de proyecto en el último año

Figura 4.6.2 Pantalla Reportes Candidatos

En caso de que la opción seleccionada sea para la generación de reportes de los colaboradores se pueden tener opciones tales como el proyecto de asignación, el rol que están jugando en ese proyecto, el perfil que deben cubrir para determinados roles, la línea o líneas de servicio que llevan a cargo un proyecto determinado, el nombre del colaborador el departamento o línea de servicio a la que esta adscrito entre otros.

La figura 4.6.3 muestra la pantalla de Reportes para los Colaboradores.

TESIS CON
FALLA DE ORIGEN

Reportes para Colaboradores. Selecciones las Llamadas de atención:

Le va a permitir el nombre del colaborador completo los campos y utilizando los siguientes ejemplos:

Nombre Colaborador: _____ Departamento: _____ Clave Colaborador: _____

Nombre completo del colaborador Proyecto asignado No de Colaborador en el Proyecto

Puesto de Colaborador en la empresa Departamento o línea de servicio Pais del Colaborador

Se le van a mostrar solamente los nombres de los siguientes departamentos. Mostrar resultados en los siguientes 3 días.

Asignado al proyecto CLRP Muestre el resultado con paginas de 3 días

Los nombres de programación en PMP Muestre el resultado con paginas de 3 días

Figura 4.6.3 Reportes para Colaboradores

En la pantalla de Reportes para Colaboradores podemos ingresar los datos correspondientes a las personas que colaboran para la empresa, es decir; los empleados. Y generar los reportes que sean de interés para las diversas áreas de la empresa Capital Humano S.A.

Llenando los campos correspondientes como son el Nombre del colaborador y el checkbox del mismo podemos obtener el nombre completo, por otro lado podemos generar el reporte partiendo de la línea de servicio o departamento para generar el reporte. También si conocemos la clave del colaborador se puede generar el reporte.

Ya que se han contemplado las opciones anteriores para generar un reporte detallado contamos con los checkbox que se encuentran en la parte de abajo con opciones determinadas.

Si deseamos generar un reporte que nos muestre el departamento, el puesto, el perfil y las actividades que realiza un empleado del cual sólo conocemos el nombre podemos hacerlo como lo muestra la figura 4.6.4.

The screenshot shows a web browser window with a search bar at the top. Below the search bar, there is a form titled "Reportes para Colaboradores. Seleccione los campos deseados:". The form contains several checkboxes and text input fields:

- Nombre completo del colaborador
- Nombre completo del colaborador
- Puesto del colaborador en la empresa
- Perfil del colaborador
- Departamento
- Perfil del colaborador en el proyecto
- Mostrar empujados con antigüedad de NINGUNO
- Empujados con antigüedad de NINGUNO

At the bottom of the form, there is a button labeled "Generar Reporte".

Figura 4.6.4 Ejemplo de un reporte.

Posteriormente sólo damos clic en el botón generar reporte para obtener nuestro resultado como lo muestra la figura 4.6.5.

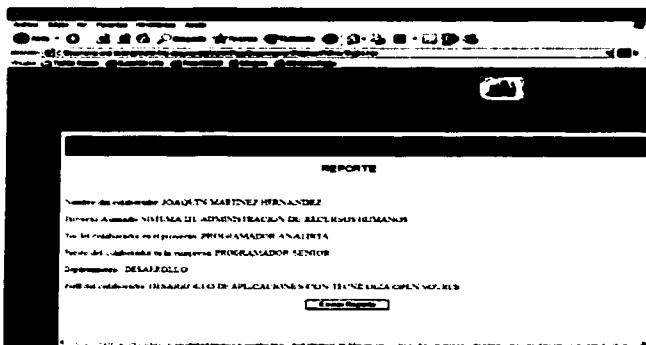


Figura 4.6.6. Resultado del reporte

REPORTE 2

En este reporte necesitamos saber que empleados con una antigüedad de 2 años están asignados a proyectos.

En la figura 4.6.6 se pueden ver los campos seleccionados.

TESIS CON
FALLA DE ORIGEN

Formulario de búsqueda de reportes:

Nombre completo del colaborador Proyecto asignado No del Colaborador en el Proyecto
 Puesto del colaborador en la empresa Departamento o Línea de Servicio País del colaborador

De la siguiente lista seleccione los campos de los reportes que desea. Mostrar resultados solo los reportes característicos.

Asignado al proyecto: Mostrar empleados con asignados en:

Consultados de área/asignado en: Encuentros con asignados en:

Empleados que han sido liberados de proyecto en el último año Empleados que están asignados a proyecto
 Empleados que han recibido capacitación Mostrar cursos de capacitación Empleados con departamentos de apoyo
 Empleados con competencias en idioma: Empleados asociados al departamento:

Figura 4.6.6 Ejemplo 2 de reportes.

En la figura 4.6.7 podemos ver el resultado de nuestra consulta, para poder enviarlo.

REPORTES

3 resultados con atributos de: datos

Nombre del empleado	Asignado
Alfonso Martínez	2 años
Alfonso Zamora	2 años
Alfonso Zamora	2 años
Alfonso Zamora	2 años

Figura 4.6.7 Resultado del reporte 2

REPORTE 3

En este reporte se piden los empleados asignados al proyecto que se encuentran trabajando en el proyecto de generación de la CURP, que puedan viajar y que hayan sido líderes de proyecto en el último año.

En la figura 4.6.8 se pueden ver los campos solicitados.

Nombre Colaborador: _____ Orden alfabético: _____ Línea Colaborador: _____

Nombre completo del colaborador
 Proyecto asignado
 Rol del Colaborador en el Proyecto
 Puesto del colaborador en la empresa
 Departamento o Línea de Servicio
 País del colaborador

De la pantalla seleccionar los campos de las siguientes opciones. Mostrar empleados con las siguientes características.

Asignados al proyecto CURP: Mostrar empleados con asignados de NINGUNO

Características de programación en NINGUNO: Empleados con destino en NINGUNO

Empleados que han sido líderes de proyecto en el último año.
 Subordinados con roles asignados a proyectos.
 Empleados que han recibido capacitación
 Mostrar cursos de capacitación.
 Empleados con disponibilidad de viajar.

Empleados con instrumentos en sistema Ninguno
 Empleados asignados al departamento: NINGUNO

Figura 4.6.8 Ejemplo de reporte 3.

En la Figura 4.6.9 se puede ver el resultado al reporte solicitado.

TESIS CON
 FALLA DE ORIGEN

En la figura 4.6.11 se puede ver el resultado del reporte anterior que puede ser enviado al departamento que la solicitó.

The screenshot shows a web browser window with a report titled "REPORTE". The report content is as follows:

Reporte por los empleados que pertenecen del departamento de Publicidad

Nombre del empleado	Departamento
Valeria Perez	Publicidad
Rafael Nolas	Publicidad
Robert Nolas	Publicidad
Tom Jones	Publicidad

Figura 4.6.11 Resultado del Reporte 4.

REPORTE 5

En el siguiente reporte se requiere saber dentro de la plantilla laboral qué empleados tienen conocimientos de programación en JAVA y/o JSP, así como saber si alguno de ellos ha tomado un diplomado en Seguridad Informática y que posibilidades tiene para ser líder de un proyecto piloto. Así que se seleccionan los campos necesarios para encontrar una persona con el perfil adecuado.

La figura 4.6.12 ilustra la pantalla para este reporte.

TESIS CON
FALLA DE ORIGEN

Nombre del usuario: _____ Correo electrónico: _____ Clave de acceso: _____

Mostrar únicamente del nivel superior Mostrar el resultado de la encuesta Mostrar los datos de los empleados

Empleados que han sido informados de proyectos en el último año Empleados que están asignados a proyectos

Empleados que han recibido capacitación Empleados que están en cursos de capacitación Empleados con disponibilidad de usar

Empleados con conocimientos en idiomas extranjeros Empleados asignados a departamentos: Departamento de Políticas

Figura 4.6.12 Pantalla de Ejemplo del Reporte 6.

El reporte anterior genera el resultado ilustrado en la figura 4.6.13.

REPORTE

Reporte de empleados con conocimientos de programación en Java o .NET, con dominio en seguridad informática y que han sido líderes de proyecto

Proyecto	Programación en Java	Programación en .NET	Lider de Proyecto
Proyecto A	SI	SI	SI
Proyecto B	SI	SI	SI
Proyecto C	SI	SI	SI
Proyecto D	SI	SI	SI
Proyecto E	SI	SI	SI
Proyecto F	SI	SI	SI

Figura 4.6.13 Resultado de el Reporte 6.

REPORTE 6

En este reporte es necesario buscar una persona que no está en la plantilla laboral de la empresa. Se recurre a las personas que se han registrado y han dejado su currícula en la base de datos. El candidato debió ser líder de proyecto en su empleo anterior. Y se busca la persona dentro de las solicitudes que fueron registradas hace 7 días.

En la figura 4.6.14 se puede ver el ejemplo.

Reporte para Candidatos: Seleccione los campos deseados.

A. Va con el nombre del candidato, complete los campos obligatorios y adicione los que se desee.

Nombre del Candidato: _____ Clave del Candidato: _____

Nombre completo del Candidato Habilidades del Candidato Habilidades Formales

De la siguiente selección los campos deseados.

Mostrar solicitudes recibidas desde: Hace 7 días Candidatos con capacitaciones de programación en: HABILIDAD

Candidatos que han sido líderes de proyecto en el último año.

Figura 4.6.14 Reporte 6

En la figura 4.6.15 se puede ser el resultado de este reporte.

TESIS CON
FALLA DE ORIGEN

Candidato	Edad	Clase de programa
Concepcion Rodriguez Vazquez	5076	SI
Marcelina Espinosa Telle	5076	SI
Andrés Vello Lopez	5076	SI
Roberto Muñoz Soto	4680	SI
Roberto Zamudio Yrizar	3999	SI
María C. Castro	3699	SI
Alfonso L. Sarmiento Domínguez	3140	SI
Gregorio Hernández Silva	2100	SI
Hilario Antonio Sánchez Gómez	1119	SI
José María Hernández	4014	SI
Teresa del Jesús López	2099	SI
Adriana Rodríguez Carrasquero	1099	SI
Enrique Muñoz Pizar	2148	SI
Adriana Urtecho Munguía	191	SI

Generar Reporte

Figura 4.6.16 Resultado del reporte 6.

REPORTE 7

El siguiente reporte muestra la forma en la que se buscaría un candidato que dejó su curriculum vitae desde hace 30 días y que tiene conocimientos de programación en PERL. La figura que muestra el ejemplo es la 4.6.16

Reportes para Candidatos. Selecciones los campos deseados.

Para buscar el número de la oferta complete los campos siguientes y seleccione la opción deseada

Nombre del candidato: _____ Clase de Programa: _____

Número completo del CV (último) Subsección del CV (último) Subsección CV (último)

De la manera siguiente se muestran los reportes deseados.

Mostrar solo candidatos que han sido contratados en los últimos 30 días Candidatos con conocimientos de programación en PERL

Candidatos con diploma en MILEAD Candidatos que han sido contratados de provincia en el último año.

Generar Reporte

Figura 4.6.16 Ejemplo de Reporte 7.

Y el resultado se puede apreciar en la figura 4.6.17.

REPORTE DE CANDIDATOS

Reporte de 1 Candidatos con cumplimiento de los requisitos en WEB - 10/01/2014 10:00:00 AM

Candidato	Código	FECHA	FECHA	FECHA	FECHA
Yusuf Pérez					
Yusuf Pérez					
Yusuf Pérez					
Yusuf Pérez					
Yusuf Pérez					

Figura 4.6.17 Resultado de el reporte 7

REPORTE 8

Finalmente en la figura 4.6.18 se muestra el ejemplo de un reporte que requiere saber qué personas dejaron su curriculum vitae desde hace 30 días, que hayan sido líderes de algún proyecto y que tengan conocimientos de Seguridad Informática.

TESIS CON
 FALLA DE ORIGEN

Reporte para Candidatos. Seleccione los campos deseados.

Se va a generar el reporte de candidatos, complete los campos indicados y seleccione el tipo de reporte.

Número del Candidato: Clave del Candidato:

Número y nombre del Candidato Subsección del Candidato Habilitación Técnica

Por lo contrario, entre en los apartados deseados.

Mostrar solicitudes pagadas desde: Més de 30 días Candidatos con Consecuencias de programación en:

Candidatos con aptitud en Seguridad Informática: Candidatos que han sido adscritos al presupuesto en el último año

Figura 4.16.18 Reporte 8.

El resultado del reporte anterior se puede ver en la figura 4.16.19, el cual puede ser enviado al departamento que lo haya solicitado con la finalidad de tomar una decisión.

REPORTE DE CANDIDATOS

El reporte de candidatos que se genera en este apartado se genera en formato de archivo PDF y se puede descargar desde el sistema.

Número	Documento	Clave de perfil
Número de Perfil	cantidad del número	SI
Número de Documento	cantidad del número	SI
Número de Perfil	cantidad del número	SI

Figura 4.16.19 Resultado del reporte 8

ANEXO 1. FORMATOS DE BÚSQUEDA Y ANÁLISIS DE INFORMACIÓN

*"Cuando llegue la inspiración que me
encuentre trabajando."
- Pablo Picasso*

ANEXO 1

CAPITAL HUMANO S.A.

Hoja 1 de 2

SOLICITUD DE INFORMACIÓN		
Puesto solicitado: _____		
Apellido paterno	Apellido Materno	Nombre(s)
Fecha de nacimiento: _____	Edad: _____	
Nacionalidad: _____	Lugar de nacimiento: _____	
Estado civil: soltero(a) ()	casado(a) ()	divorciado(a) ()
Otros: _____		
Sexo: F () M ()		
CURP ó RFC: _____		
Estudios:	Años cursados	Documentos oficiales
Primaria	_____	_____
Secundaria	_____	_____
Preparatoria	_____	_____
Universidad	_____	_____
Otros	_____	_____
No. Pasaporte: _____	No. cartilla militar: _____	
No. Licencia de Manejo: _____		
No. Afiliación IMSS ó ISSSTE: _____		

DOMICILIO	
Calle y número: _____	
Colonia: _____	
Delegación o municipio: _____	
C.P. _____	Ciudad o Estado: _____
País: _____	Teléfono: _____
E-mail: _____	

Formato 2.3.1 Formato de solicitud de ingreso (hoja 1).

TESIS CON
FALLA DE ORIGEN

CAPITAL HUMANO S.A.**EXPERIENCIA LABORAL**

Hoja 2 de 2

¿Trabaja actualmente?

Nombre de la empresa:**Dirección:****Teléfono:****Puesto:****Sueldo:****Nombre del jefe inmediato:****Puesto:****Experiencia:****Teléfono:****Trabajo anterior:****Nombre de la empresa:****Dirección:****Teléfono:****Experiencia:****Puesto:****Trabajo anterior:****Nombre de la empresa:****Dirección:****Teléfono:****Experiencia:****Puesto:****Firma:****Fecha de elaboración:**

Formato 2.3.1 Formato de solicitud de ingreso (hoja 2)

CAPITAL HUMANO S.A.

Hoja 1 de 1

TIPO DE PERSONAL EN PROCESO DE SELECCIÓN	
Candidato:	()
Personal previo:	()
Para entrevista final:	()
Para contratación:	()

DATOS GENERALES DEL ENTREVISTADO		
Apellido paterno	Apellido Materno	Nombre(s)
Fecha de nacimiento: _____	Edad: _____	
CURP ó RFC: _____		
Último grado de estudios: _____		
Certificado obtenido: _____		

PROCESO DE LA ENTREVISTA		
1. () Fecha de la entrevista:		
2. () Teléfono particular:		
3. () Aceptado:		
4. () Entrevista elaborada:	El personal aceptó la entrevista: SI () NO ()	
5. () Cubre papeles oficiales:		
6. () Fecha de exámenes:		
7. () Resultados de exámenes:		
Técnicos	Aprobado ()	No aprobado ()
Personalidad	Aprobado ()	No aprobado ()
Psicotécnicos	Aprobado ()	No aprobado ()
8. () Entrevista final		
9. () Reglamento		
10. () Indicar fecha de actividades		

Elaboró: _____
Fecha: _____

Formato 2.3.2 Procedimiento de selección de personal.

TESIS CON
FALLA DE ORIGEN

CAPITAL HUMANO S.A.

Hoja 1 de 1

DATOS GENERALES DEL ENTREVISTADO

DATOS GENERALES DEL ENTREVISTADO		
Apellido paterno	Apellido Materno	Nombre(s)
Puesto y sueldo autorizados		

AREA SOLICITANTE

Requisitos del área:

Perfil:

DATOS ADICIONALES DEL PERSONAL SELECCIONADO

Documentos oficiales	
Constancias de cursos recibidos	
Experiencia :	
Resultado de evaluaciones:	
Técnicos	
Personalidad	
Psicotécnicos	
Recomendaciones:	

Formato 2.3.3 Este documento se utiliza en el procedimiento de selección del personal e ingresar y es llenado por el departamento de recursos humanos.

**TESIS CON
FALLA DE ORIGEN**

CAPITAL HUMANO S.A.

MOTIVOS DE CONTRATACIÓN		
Estatus del empleado: Alta ()	Baja ()	Cambio ()
Tipo de contratación:	_____	
Es recontratable: Por qué?:	SI ()	NO ()

Es recomendable: Por qué?:	SI ()	NO ()

Puestos que ha ocupado:	_____	

Elaboró: _____

Fecha de elaboración: _____

Formato 2.3.4 Hoja final del proceso de selección del personal.

Toda la información recabada en los formatos durante el proceso de selección, po es capturada al sistema por el operador para obtener un mayor control y manejo de los datos del personal.

**TESIS CON
FALLA DE ORIGEN**

ANEXO 2. MANUA TÉCNICO Y DE USUARIO

"Mathemata mathematicis scribuntur (las matemáticas se escriben para los matemáticos)".

– Nicolás Copérnico

TESIS CON
FALLA DE ORIGEN

MANUAL TÉCNICO.

Respaldo de la base de datos.

Para llevar a cabo el respaldo de la base de datos es necesario ejecutar el comando `pg_dump`, el cual crea un archivo de texto con sentencias SQL que recrean la base de datos tal y como está al momento de ejecutar el comando. El uso básico de este comando es:

```
Pg_dump nombre_base > archivo_salida
```

`Pg_dump` es una aplicación cliente regular de PostgreSQL. Esto significa que puede realizarse este procedimiento de backup desde cualquier host remoto que tenga acceso a la base de datos. Pero recuerde que `pg_dump` no opera con permisos especiales. En particular, debe tener acceso de lectura a todas las tablas que se quieran respaldar, así que en la práctica debe ser superusuario de la base de datos.

Los volcados creados por `pg_dump` son consistentes internamente, es decir, las actualizaciones a la base de datos mientras que el comando `pg_dump` esté corriendo no se verán reflejados en el volcado. `Pg_dump` no bloquea otras operaciones en la base de datos mientras está en ejecución. (Excepto las operaciones que necesiten operar con un bloqueo exclusivo, como `VACUUM FULL`.)

Importante: Cuando el modelo de la base de datos se basa en OIDs (llaves foráneas por ejemplo) debe indicar a `pg_dump` que vuelque también los OIDs. Para hacer esto, utilice la opción `-o` de la línea de comandos. "Objetos grandes" no son volcados por omisión.

A partir de la experiencia sufrida durante el terremoto de 1985 en la Ciudad de México, es recomendable generar un respaldo de la base de datos local y otro fuera de la localidad para evitar pérdidas en caso de un siniestro.

Recuperación del respaldo.

Los archivos de texto generados por `pg_dump` están creados para ser leídos por el programa `psql`. La forma general del comando de restauración de un volcado es:

```
Psql nombre_base < archivo_entrada
```

Donde `archivo_entrada` es el archivo generado por el comando `pg_dump`. El nombre `nombre_base` no será creado con este comando, debe primero ser creado manualmente utilizando la plantilla "template0" antes de ejecutar el comando `psql` (por ejemplo, con `createdb -T template0 nombre_base`). `Psql` soporta opciones similares a `pg_dump` para controlar la localización del servidor de base de datos y los nombres de usuario.

Si los objetos en la base de datos original pertenecen a diferentes usuarios, entonces el volcado instruirá a `psql` para conectarse como cada usuario afectado en turno y luego crear los objetos relevantes. De esta forma la pertenencia original es mantenida. Esto también significa, sin embargo, que todos estos usuarios deben existir, y aún más, que usted debe poder conectarse como cada uno de ellos. Es posible por lo tanto suavizar la configuración de autenticación del cliente. La habilidad del `pg_dump` y `psql` de escribir o leer desde pipes, hace posible el volcado a una base de datos directamente de un servidor a otro, por ejemplo:

```
pg_dump -h host1 nombre_base | psql -h host2 nombre_base
```

Importante: los volcados producidos con `pg_dump` son relativos a la plantilla "template0". Esto significa que los lenguajes, procedimientos, etc. Añadidos a la plantilla "template1" también serán volcados por `pg_dump`. Como resultado, cuando se esté recuperando la base de datos, si usted está utilizando una plantilla actualizada "template1", debe crear una base de datos vacía desde la plantilla "template0".

TESIS CON
FALLA DE ORIGEN

Mantenimiento de la base de datos.

Ocupación del espacio en disco. Las tablas de la base de datos que presentan un crecimiento mayor, es decir, que almacenan la mayor cantidad de registros al utilizar el sistema son:

- Persona
- Boletín
- Mensaje
- Persona_Idioma
- Colaborador_Capacitación
- Rol_Colaborador
- Mensaje_Persona
- Programación
- Colaborador_Línea_Puesto

Existe una fórmula para calcular el crecimiento de una base de datos con base en la cantidad de información introducida en una sesión, y es la siguiente:

$((\text{Tamaño de la sesión}) * (\# \text{ de sesiones/día})) * (\text{días}) = (\text{Tamaño estimado para la base de datos})$

Estas tablas en conjunto crecen a razón de 2 Kbytes por persona que se registre en la base de datos. Si existe un registro de 5 personas al día, el crecimiento en disco será de 10 Kbytes al día, por lo tanto, el crecimiento mensual al ritmo mencionado anteriormente será de aproximadamente 300 Kbytes al mes.

Recuperación de espacio en disco. En la operación normal de PostgreSQL, un UPDATE o un DELETE de un registro no eliminan inmediatamente la tupla (versión del registro). Esto es necesario para mantener un control de concurrencia (múltiples accesos simultáneos): la tupla no debe ser borrada mientras sea potencialmente visible para otras transacciones. Pero finalmente, una tupla antigua o borrada no es de interés para transacción alguna. El espacio que ocupa debe ser recuperado para ser utilizado por nuevas tuplas, para evitar un

crecimiento infinito del requerimiento de espacio en disco. Esto se lleva a cabo ejecutando el comando *VACUUM*.

Por ello, es necesario ejecutar el comando *VACUUM* de forma periódica a la tabla "Persona", de preferencia programando una tarea en el sistema operativo para ello.

Actualización de las estadísticas del planeador de consultas SQL.

El planeador de consultas SQL está basado en información estadística sobre los contenidos de las tablas para generar buenos planes para las consultas. Estas estadísticas son recopiladas por el comando *ANALYZE*, el cual puede ser invocado por sí mismo o como un paso opcional en el comando *VACUUM*. Es importante mantener estadísticas razonablemente precisas o de lo contrario malas elecciones de planeación pueden degradar el rendimiento de la base de datos. Así como la depuración para la recuperación de espacio, actualizaciones frecuentes son mejores para tablas constantemente actualizadas.

Es necesario ejecutar el comando *ANALYZE* de forma periódica en las tablas que presentan la mayor cantidad de crecimiento mencionadas previamente con el fin de mantener el rendimiento del sistema al máximo, reduciendo tiempos de consulta.

MANUAL DEL USUARIO.

Para ingresar al sistema el usuario deberá ingresar el Login y el Password. Una vez escritos el login y el password en el formulario, deberá oprimir el botón de ingresar el cual se encuentra en la parte inferior del formulario (ver Fig. 1.1)

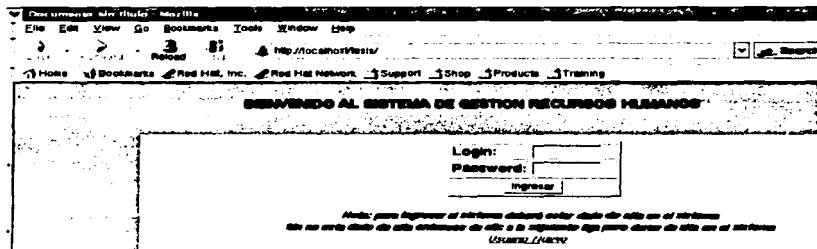


Figura 1.1 Ingresar al sistema de recursos humanos

Es importante mencionar que si el usuario no está dado de alta en el sistema, lo hará de la siguiente manera: deberá dar clic en la liga **Usuario Nuevo**. Le aparecerá la siguiente opción (ver Fig.1.2)

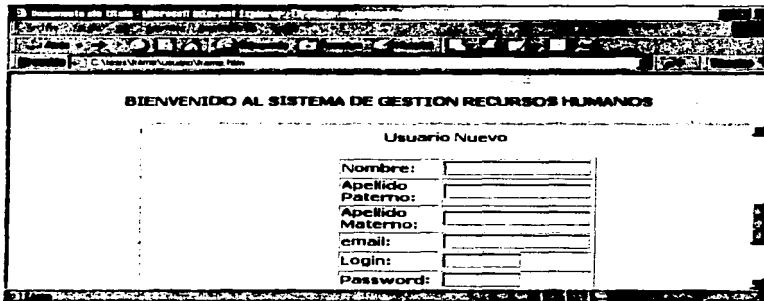


Figura 1.2 Agregar usuario

TESIS CON
FALLA DE ORIGEN

Una vez que haya entrado al sistema, se desplegará su posición global, es decir, la pantalla principal con un menú del lado izquierdo que tendrá diversas opciones dependiendo del tipo de usuario que se trate, para este sistema se tomaron a consideración tres bloques: el primero es el de usuario común; el segundo es del operador, el cual estará a cargo de una persona que sea responsable del Departamento de Recursos Humanos, y por último el administrador del sistema, el cual tendrá a su cargo sólo ciertas responsabilidades, como borrar usuario, desbloquear usuario, etc.

Si al ingresar el login y password es de un usuario verá la siguiente pantalla (ver Fig.1.3)

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

- Navigation Bar:** Archivo, Editar, Ver, Favoritos, Herramientas, Ayuda
- Address Bar:** D:\Yahoo\Des\1\test\Vermensaje_usuario.asp
- Page Title:** eCAPITAL HUMANO S.A. **SIAREH**
- Left Menu (Menú Principal):**
 - [Inicio](#)
 - [Inicio de Sesión](#)
 - [Mantener Datos Personales](#)
 - [Cambio de Contraseña](#)
 - [Cambio de Usuario](#)
 - [Salir de SIAREH](#)
- Main Content Area:**

Hoy es: 2-Jul-2003 y son las: 10:47

Notas Informativas:

Les recordamos que durante el mes de julio pueden pasar a recoger su reparto de utilidades.

NO OLVIDEN que el próximo jueves habrá tocos al pastor y yardas cortés de la taquería de Iván Marquez... GRACIAS IVÁN

NO FALTEN!!!!

Bienvenido Joaquín, tu último acceso a SIAREH fue el 13-Jun-2003 a las 4:27 PM.

Figura 1.3 Menú del usuario

TESIS CON
FALLA DE ORIGEN

Se puede notar que en la parte izquierda, aparece el menú principal donde la primera liga corresponde a *Búsqueda de cursos*, si se selecciona esta opción se visualizará la siguiente pantalla (ver Fig. 1.4)

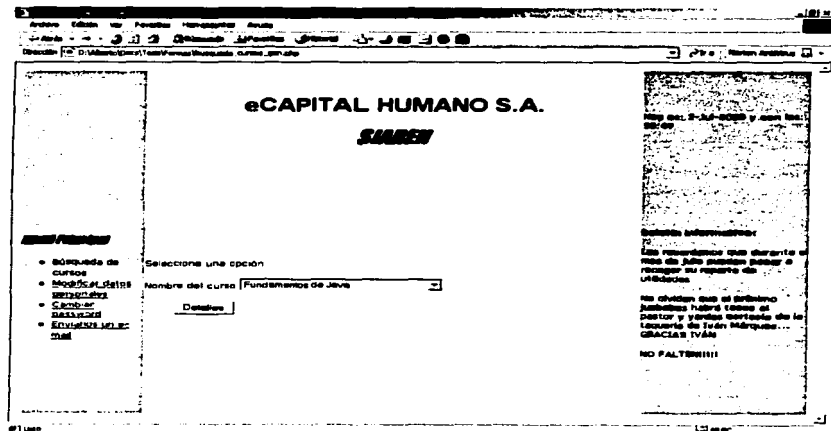


Figura 1.4 Búsqueda de cursos

El usuario podrá ingresar sus datos personales así como también su currícula, de tal forma que el usuario podrá actualizar, eliminar o borrar datos personales, cabe aclarar que depende del usuario la actualización de sus datos personales, así como los de su currícula.

Es importante para el Departamento de Recursos Humanos saber la opinión del usuario y es por esto que el sistema tiene una opción donde podrá enviar quejas y sugerencias (ver Fig.1.5)

TESIS CON
FALLA DE ORIGEN

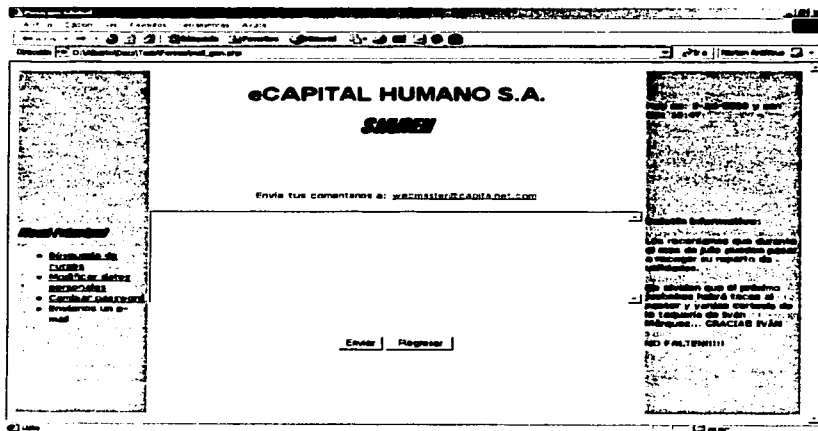


Figura 1.5 Enviar quejas y sugerencias

Esta es la posición global o pantalla principal del operador (ver Fig. 1.6)

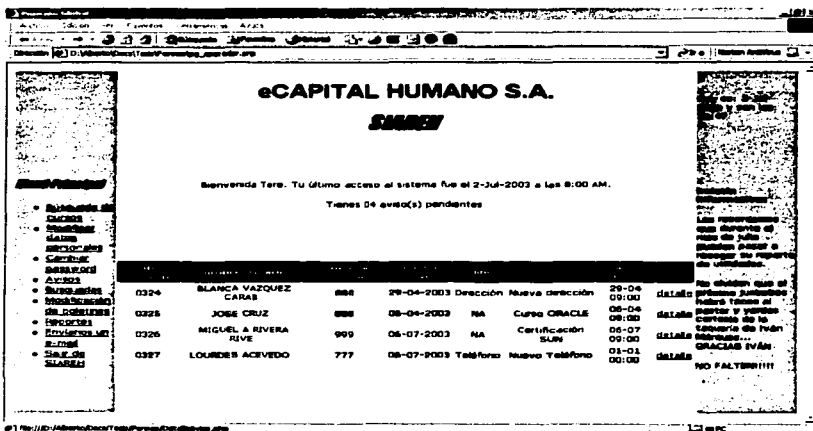


Figura 1.6 Actividades del Operador

El operador podrá hacer búsquedas simples y avanzadas del personal que se encuentra en este momento en la empresa, la búsqueda puede ser por habilidades y si se encuentra en disponibilidad (ver Fig. 1.7).

TESIS CON
FALLA DE ORIGEN

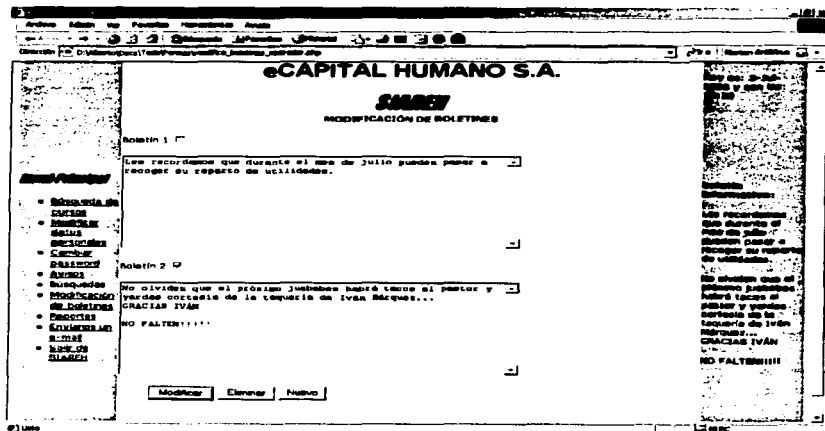


Figura 1.9 Boletines

Una de las tareas que el administrador del sistema puede realizar, es la de desbloquear usuarios (ver Fig. 1.9) donde podemos ver que se muestra una lista de los usuarios bloqueados y en la última columna de la misma tenemos la opción de desbloquear usuario.

TESIS CON
FALLA DE ORIGEN

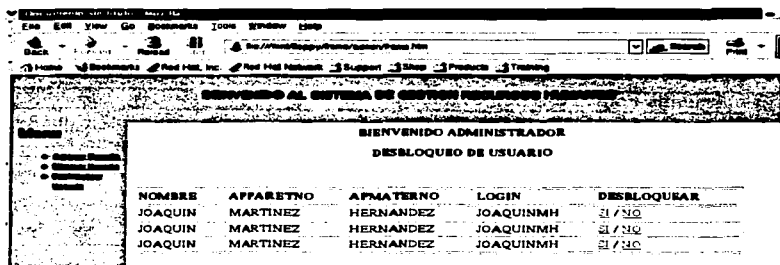


Figura 1.10 Desbloqueo de usuario

Otra de las actividades que el administrador puede hacer, es buscar un usuario para borrarlo o cambiar sus datos (ver Fig. 1.10)

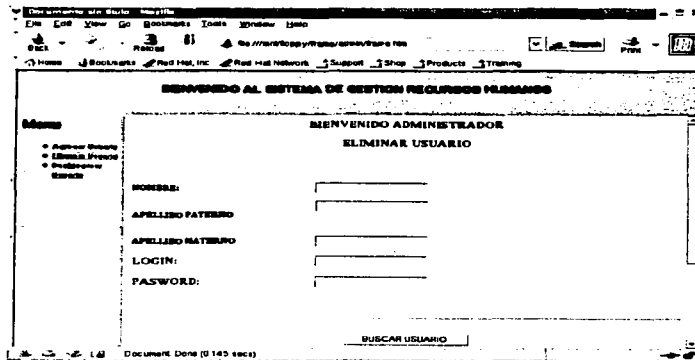


Figura 1.11 Eliminar usuario

Este es el manual de usuario donde podremos consultar alguna duda acerca del sistema. Cualquier duda o sugerencia, enviar un correo a la siguiente dirección: webmaster@capitalnet.com.

TESIS CON
FALLA DE ORIGEN

CONCLUSIONES

*"Con buenas palabras se puede negociar pero
para engrandecerse se requieren buenas obras".
— Lao-tse*

TESIS CON
FALLA DE ORIGEN

CONCLUSIONES.

Todos los que participamos en este trabajo contamos con algunos años de experiencia profesional y hemos comprobado que la informática es una herramienta fundamental para obtener resultados tangibles, consistentes y de gran valor que sirven para mejorar, modernizar y hacer más eficientes y productivos los procesos de negocios de los más diversos giros.

Aplicando el conocimiento obtenido durante nuestros estudios de ingeniería en computación en combinación con la experiencia en el campo laboral, pudimos culminar satisfactoriamente el ciclo del proyecto que nos propusimos como tema de esta tesis desde su concepción hasta su implantación en un ambiente productivo. Trabajamos en equipo durante largas sesiones de análisis aportando ideas, debatiendo puntos de vista e intercambiando experiencias. No fue fácil tomar decisiones con respecto a las herramientas a utilizar para el desarrollo de la aplicación, aunque al final, la mayoría estuvo conforme con cada etapa. Todos y cada uno de los miembros del equipo aprendimos algo: Desde una nueva sintaxis de lenguaje hasta una nueva metodología de análisis y diseño pasando por aspectos técnicos concernientes a sistemas operativos, redes y comunicaciones. El desarrollo del SIAREH siguiendo una disciplina metódica e iterativa nos convence de la enorme utilidad que tiene la ingeniería en computación aplicada a la creación de software robusto. Desafortunadamente, también hemos comprobado que en el campo laboral no siempre se trabaja con la formalidad necesaria, en ocasiones se abusa del empirismo y se pagan severas consecuencias que se pueden traducir en gastos humanos y materiales excesivos. La conclusión de este trabajo nos deja la enorme satisfacción de entregar un producto de utilidad y calidad. Sabemos que dimos nuestro máximo esfuerzo y que valió la pena hacer algunos sacrificios.

El Programa de Apoyo a la Titulación ha sido fundamental para nosotros. Logramos sacudir el miedo que produce enfrentarse al hecho de escribir una tesis

TESIS CON
FALLA DE ORIGEN

en menos de doce sesiones. Las asesorías semanales nos permitieron concebir y proponer un tema de tesis suficientemente aterrizado, aprendimos semana a semana a organizar nuestro tiempo para obtener resultados rápidos y concretos, redactando de manera concisa. Gracias al programa pudimos seguir, de manera efectiva, los pasos para desarrollar una tesis, eliminando de ella lo que no aportaba información importante y dejando aquello que nos permitiera explicar de forma clara y sencilla el tema que escogimos. Aprendimos, también, a estructurar y a organizar la exposición oral derivada de nuestro trabajo escrito.

Todas las ramas de la ingeniería, en lo académico y en lo práctico, requieren forzosamente trabajar en equipo. No es posible hacerlo de otra manera. Lo hemos comprobado durante la carrera universitaria y durante la profesional. Los que participamos en el equipo que desarrolló el SIAREH desconocíamos las personalidades y el estilo de trabajo de cada uno, sin embargo, logramos adaptarnos rápidamente y ejecutamos las tareas asignadas de cada sesión de manera conjunta y responsable.

Haber estudiado la carrera de Ingeniería en Computación ha sido una experiencia única para nosotros como personas por muchas y muy diversas razones. Una de ellas, por el gran esfuerzo que implica recorrer los diez semestres y todas las materias que se deben acreditar. Otra, por el hecho de sacrificar una buena cantidad de fiestas, reuniones familiares, viajes, horas de sueño, entre otros. Pero en esta experiencia también están las razones que nos permitieron seguir en el camino: Nada se puede comparar a la obtención de una buena nota después de un largo y duro semestre, a la satisfacción de vislumbrar en el horizonte que falta cada vez menos, solo unas cuantas materias más para cumplir con todo el plan de estudios. Los sentimientos y las emociones son constantes y, sobre todo, contrastantes. Finalmente, estamos a punto de culminar el trayecto y obtener nuestro título de ingenieros en computación.

BIBLIOGRAFÍA

"No ser útil a nadie equivale a no valer nada".
— René Descartes

TESIS CON
FALLA DE ORIGEN

BIBLIOGRAFÍA.

"Linux 6 edition". Bandel David, Napier Robert,
Prentice Hall.

"Diseño de sistemas de información: Teoría y práctica" (1ª edición en español),
Burch John G, Grudzinski Gary,
Editorial Grupo Noriega Editores.

"Beginning PHP5". Choi Wankyu; Kent Allan; Lea Chris; Prased Grangh; Ullman
Chris; Blank Jon, País: EUA
Wrox. isbn 1801003730

"A programmer's introduction to PHP 4.0". Gilmore, W.J., 2001,
Editorial Apress.

"Fundamentos de bases de datos", 3a edición, Korth H.F., Silberschatz A.,
Sudarshan S, McGraw Hill. (1999.).

"PHP Pocket Reference" (1ª edición), Lerdorf, Rasmus, 2000.
O'Reilly.

"Protocolos de Internet, diseño e implementación en
sistemas UNIX" (1a edición), López Angel, Novo Alejandro,
AlfaOmega/Ra-ma.

"Manual de referencia linux" (2ª edición), Petersen, Richard. 2001.,
McGraw Hill.

"Manual de administración de Linux", Shah, Steve. 2001.,
McGraw Hill.

"Ingeniería de Software Orientada a Objetos con UML, java e Internet",
Weitzenfeld, Alfredo. Thomson Learning, 2003.

TESIS CON
FALLA DE ORIGEN

Fuentes consultadas en Internet:

- http://asexma.cl/contenido/fomento//consultoria_pymes/388html
- <http://www.soluciona.es/SrvIndices?tipo=menu&seccion=Consultor%EDa>
- <http://www.iec.csic.es/criptonomicon/linux>
Seguridad en linux
- <http://www.grulic.org.ar/comos/info-sheet/Infosheet-como-2.html>
Características de linux
-para postgresQL Documentacion
- http://jamesthornton.com/postgres/7_2/
- <http://www.pchardware.org/redes/redes.php#inicio>
- <http://www.postgresql.org/>
- <http://www.webopedia.com/>
- <http://www.php.org/>
- <http://unamosapuntos3.tripod.com/user/rechuma/tesiszully.htm>
- <http://www.xcellenet.com/public/products/afaria/technology.asp>
- <http://www.idc.usb.ve/~teruel/ci4713/clases2001/testUML.html>
- http://www.fro.utn.edu.ar/isi/dds/html_data/docum.htm,
- <http://www.capitalhumano.net>,
- <http://www.mexicoextremo.com.mx/ayuda/librolinux/index.php?nombre=index-1.html>,
- http://www.netapp.com/tech_library/3071.html,
- <http://www.desarrolloweb.com/manuales/6/>,
- <http://www.weberdev.com/>