

TABLA DE CONTENIDO

	Pag.
INTRODUCCION	14
1. MARCO DE REFERENCIA	16
1.1 MARCO CONTEXTUAL	16
1.2 MARCO CONCEPTUAL	21
1.2.1 Sistema Operativo	21
1.2.2 Sistema Operativo Linux	21
1.2.3 GPL(General Public Lincense)	21
1.2.4 Servidor Web	23
1.2.5 Servidor Web Apache	24
1.2.6 Servidor de Nombres de Dominio (DNS)	25
1.2.7 Servidor FTP	25
1.2.8 Servidor de Correo	25
1.2.9 Servidor de Correo Qmail	26
1.2.10 Base de Datos	30
1.2.11 Motor de Bases de Datos postgresql	30

	15
1.2.12 Páginas Web	37
1.2.13 Páginas Dinámicas	37
1.2.14 PHP	39
2. DESARROLLO DE ACTIVIDADES	41
2.1 ADQUISICION DE INFORAMCION Y SOTWARE	41
2.2 INSTALACION Y CONFIGURACION DE LINUX	41
2.3 CONFIGURACION SERVIDOR DNS	45
2.4 CONFIGURACION DEL SERVIDOR FTP WU-FTPD	48
2.5 INSTALACION Y CONFIGURACION DE APACHE	49
2.6 INSTALACION Y CONFIGURACION DE QMAIL	50
2.7 INSTALACION Y CONFIGURACION DE PHP COMO MODULO DE APACHE	54
2.8 INSTALACION Y CONFIGURACION POSTFRESQL	54
2.9 ANALISIS DEL SISTEMA EN EL AREA DE TELEMATICA	57
2.10 DISEÑO LOGICO DE LA BASE DE DATOS	60
2.10.1 Diagrama de Contexto	61
2.10.2 Diagramas Entidad Relación	62
2.10.3 Tablas Telemática	66
2.10.4 Tablas Transmisión	69
2.10.5 Tablas Conmutación	72
2.10.6 Tablas Energía	73
2.10.7 Tabla Independiente de Secciones	73

	16
2.11 IMPLEMENTACION DE LA BASE DE DATOS	74
2.12 DISEÑO DE LAS PAGINAS CON ACCESO A LA BASE DE DATOS	75
2.12.1 Página Principal General	75
2.12.2 Página Principal para cada sección	76
3. CONCLUSIONES	82
4. RECOMENDACIONES	84
BIBLIOGRAFIA	86
ANEXOS	87

LISTA DE TABLAS**Pág.**

Tabla 1. Niveles de Aislamiento y acciones.	34
Tabla 2. Características del Equipo de Trabajo.	42
Tabla 3. Entidades y Atributos Telemática.	
59	
Tabla 4. Relación de Grupos y Usuarios.	74

LISTA DE FIGURAS

	Pág.
Figura 1. Estructura Orgánica de Telecom.	19
Figura 2. Esquema de Comunicación entre el Cliente y el Servidor Web.	24
Figura 3. Arquitectura de Postgresql.	36
Figura 4. Esquema conexión paginas PHP.	40
Figura 5. Arbol de Directorios Servidor FTP.	49
Figura 6. Flujo de Información Telemática.	57
Figura 7. Datos Telemática.	58

LISTA DE ANEXOS**Pág.**

Anexo A. Scripts de Generación de Tablas.	87
Anexo B. Scripts de Asignación de permisos a Usuarios y Grupos.	103
Anexo C. Scripts de Creación de Reglas.	106
Anexo D. Scripts de Parte del Código de las Páginas Web del Proyecto	107
Anexo E. Referencias Internet.	134

RESUMEN

El objetivo de éste proyecto es aprovechar las posibilidades que ofrece Internet para facilitar el manejo de la información del **Grupo Técnico Operativo, Gerencia Departamental de Nariño** de la **Empresa Nacional de Telecomunicaciones TELECOM**. Los servidores Web, FTP, DNS, CORREO, las páginas Web, las Paginas Dinámicas y las Bases de Datos son herramientas que fueron utilizadas para el desarrollo de este proyecto, considerando aspectos de su instalación, configuración y aplicación.

En el proyecto no solo se hace uso de estas herramientas sino que también se trata de explicar los aspectos más relevantes que las caracterizan. De otra parte, en relación con la aplicación, se hace un análisis de la información y su flujo a través de las áreas que conforman el **Grupo Técnico Operativo**, con el propósito de diseñar la Base de Datos que será accedida desde una página Web desde distintos puntos que conforma la intranet de Telecom.

Para el desarrollo de la aplicación se empleó la plataforma Linux, Postgresql como motor de la Base de Datos y PHP como lenguaje de desarrollo. Para la Interfaz de usuario se emplearon herramientas de diseño como Flash, Fireworks y Dreamweaver, para Windows 9X y en la edición de gráficos Gimp para Linux..

ABSTRACT

The goal of this project is to take advantage of the possibilities that offers the Internet to facilitate the manage of the information of the **Grupo Tecncico Operativo of the Gerencia Departamental de Nariño of TELECOM**. The Web, FTP, DNS, MAIL servers, the web pages, the Dynamical Pages and the Databases are tools that were used for the development of this project, considering aspects such as its installation, its configuration and its use.

In this project this tools not only are used, but is tried to explain the most relevant aspects that characterize them. Of another part, in relation with the application, an analysis of the information and its flow is made across the sections that form the **Grupo Técnico Operativo**, with the intention to design a Database that will be accessed from a Web page from different nodes of the Telecom Intranet.

Linux was used for the application development, Postgresql was the Database motor and PHP was selected as development language. For the user interface was used design tools such as Flash, Fireworks and Dreamweaver for Windows 9X, and for the graphics edition was used Gimp for Linux.

INTRODUCCION

La información recopilada en este documento identifica todas las actividades realizadas, dificultades encontradas y sus soluciones, experiencias adquiridas y recomendaciones que se presentaron durante la realización del proyecto “Servidor Internet de prueba bajo sistema operativo Linux, Análisis y Diseño de una Base de Datos para Telecom Pasto, y Diseño de Paginas Web con acceso dinámico a Bases de Datos”.

El documento está dividido en cuatro temas: el primero es el Marco Referencial que a su vez se encuentra dividido en Marco Contextual y Marco Conceptual; el primero se refiere al entorno en el cual se desarrolló el proyecto, identificando la empresa y el área específica de trabajo. El segundo, que permite conocer información sobre las tecnologías (Software) empleadas tanto en el diseño como en la implementación; el sistema Operativo Linux, Servidores: DNS, FTP, Correo y Web, SQL Motores de Bases de Datos, Lenguajes de programación de páginas dinámicas, entre otros. El segundo tema describe todas las actividades en orden cronológico que tuvieron que ser desarrolladas para lograr los objetivos propuestos en el proyecto. La tercera parte contiene las conclusiones finales, después de haber culminado el desarrollo de todas las actividades. Y finalmente en el cuarto tema se presentan unas recomendaciones, que se espera sirvan de guía para futuras modificaciones o

implementaciones de proyectos similares, basadas en las experiencias adquiridas, los problemas que se presentaron y las soluciones que se adoptaron.

1. MARCO DE REFERENCIA

1.1 MARCO CONTEXTUAL

El proyecto se desarrollo dentro de la “Empresa Nacional de Telecomunicaciones” Telecom, Gerencia Departamental de Nariño. Se presenta en seguida una breve reseña sobre la empresa, iniciando con algunas generalidades de Telecom hasta situarse dentro del entorno cercano que es el área de Telemática, conocida también como el área de Servicios de Datos de Telecom en Pasto.

La Empresa Nacional de Telecomunicaciones ha estado asociada a la historia de Colombia desde hace más de 50 años y cada salto tecnológico que ha dado la Empresa ha contribuido de alguna manera al desarrollo del país; existe, por lo tanto, un vínculo entre la población, bien sea de las principales ciudades o de las más apartadas regiones del país, y Telecom. Aunque la primera asociación con Telecom que se hace, es hacia los servicios de telefonía, Telecom representa mucho más que eso.

Telecom es la más grande empresa de Telecomunicaciones del País y pionera del desarrollo de este campo en Latinoamérica. Además de prestar servicios de telefonía de nivel local, regional, nacional o internacional, Telecom ofrece los siguientes Servicios:

- Transmisión por Microondas (para comunicaciones analógicas de voz y video)
- Teleconferencia y Videoconferencia
- Servicios Telefónicos Empresariales
- Servicios de Red Inteligente (que incluye una gama amplia de servicios como: Pago Revertido -9800-, Número Unico Empresarial, Teleopinión, Telecódigo, etc.)
- Servicios de comunicación de datos, tales como:
 - Servicios Satelitales VSAT apropiados para transmisión de datos a baja velocidad.
 - Servicios de red privada virtual (VPN)
 - Redes Corporativas
 - Canales Dedicados (nacionales e Internaciones con protocolos Frame Relay, Clear Chanel, X25, X28)
 - Accesos Dedicados y conmutados a Internet
 - Servicios de Hosting

La Empresa ha hecho cuantiosas inversiones en materia tecnológica, entre las que cabe mencionarse: la Red Nacional de Fibra Óptica, que interconecta las 18 principales ciudades del país; la Red Troncal de Microondas, para minimizar los niveles de congestión de la Red de Transporte; Cables Submarinos, que interconectan a Colombia con la red mundial, buscando en todo caso posicionar a TELECOM como uno de los principales "CARRIERS" internacionales de Sur América, para lo cual ha invertido en el cable Panamericano, que va desde Chile hasta las Islas Vírgenes, en el TPC-5 que une a USA con Japón, PENCAN - 5 que une las Islas Canarias con España y Maya - 1 en el cual participan 17 conectantes internacionales.

En el ámbito internacional **Telecom** mantiene grandes acuerdos con las más importantes empresas de Telecomunicaciones del mundo entre las que están: WorldCom y MCI, AtyT y SPRINT, logrando conexión con más de 250 países.

Telecom, como todas las empresas del sector, está sujeta a la regulación y control de entidades del Estado, como el Ministerio de Comunicaciones en los servicios no regulados por la Ley de Servicios Públicos Domiciliarios (Servicios de Valor agregado y demás), la Comisión de Regulación de Telecomunicaciones (CRT, ente gubernamental encargado de establecer las directrices en materia regulatoria que deben seguir los operadores de los servicios Telefonía Pública Básica Conmutada Local, Local Extendida y de Larga Distancia); y la Superintendencia de Servicios Públicos Domiciliarios, encargada de vigilar y controlar las empresas prestadoras de servicios públicos domiciliarios.

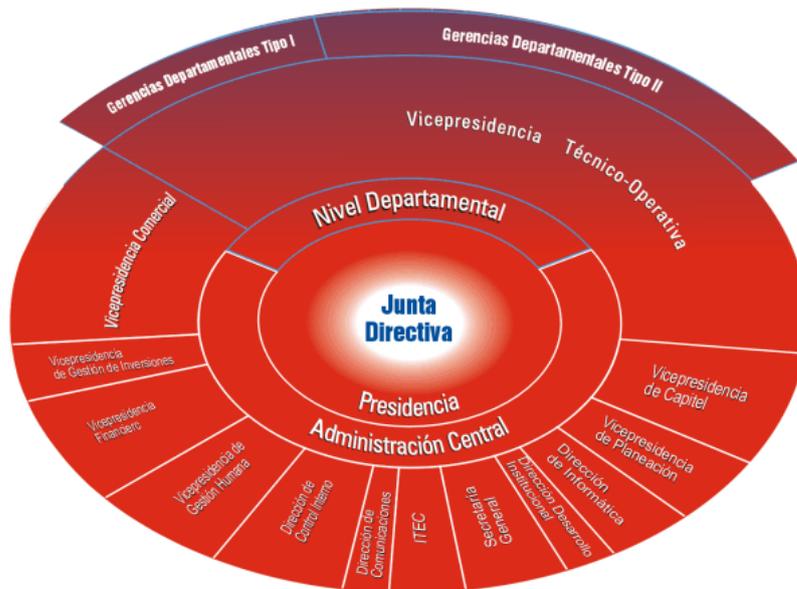
Organizacionalmente **Telecom** presenta una estructura de carácter circular, representada en la Figura 1.

En esta estructura, la máxima instancia es la Junta Directiva de Telecom. Las Gerencias Departamentales, como la de Nariño depende directamente de la Vicepresidencia Técnico Operativa.

Gerencia Departamental de Nariño. Dentro del organigrama aparece dentro del grupo de Gerencias Departamentales de Tipo II; esta clasificación se hace teniendo en cuenta la capacidad de sus instalaciones, el cubrimiento y su producción. Así mismo la clasificación está

relacionada directamente con el presupuesto anual asignado para su funcionamiento y con el presupuesto de inversiones.

Figura 1. Estructura Orgánica de Telecom



Como estrategia organizativa, dentro de la Gerencia Departamental de Nariño se han establecido los siguientes grupos:

- Grupo Técnico Operativo
- Grupo Financiero
- Grupo de Atención al Cliente
- Grupo de Gestión del Servicio
- Grupo de Gestión Humana
- Grupo de Gestión Administrativa

Cada grupo posee un líder, que conjuntamente con el Gerente Departamental conforman el Comité Ejecutivo, máxima instancia departamental, encargada de implementar y proyectar planes para el desarrollo de las telecomunicaciones en la región.

Es de anotar, que el grupo de interés en el desarrollo del proyecto es el Grupo Técnico Operativo, encargado de garantizar el funcionamiento de los sistemas que permiten el desarrollo de las telecomunicaciones a nivel departamental.

El Grupo está conformado por las áreas de: Transmisión, Telemática, Conmutación y Energía.

El presente proyecto se desarrolla en el área de Telemática (o de Servicios de Datos), sobre la que se presente la siguiente reseña:

Área de Datos. El Área de Servicios Telemáticos se encarga de la gestión de los servicios de comunicación de datos, relacionados básicamente con la conexión de Canales Nacionales de Datos (Clear Chanel, Frame Relay o de Canales con Acceso Dedicado a Internet). El área es una de las dependencias con mayor proyección dentro de la Empresa, en razón del crecimiento de las necesidades de comunicaciones de los clientes en el departamento, del avance tecnológico y del fortalecimiento que Telecom ha realizado sobre su infraestructura.

El área de Datos tiene como visión ofrecer un servicio oportuno en el diseño, implementación y monitoreo de las soluciones para la transmisión de datos de sus clientes, apoyados en tecnologías de punta y en un personal debidamente capacitado.

Aunque el proyecto se desarrolló en el área de Telemática, se incluyeron de manera adicional datos que se manejan en las áreas de Transmisión, Conmutación y Energía.

1.2 MARCO CONCEPTUAL

1.2.1 Sistema Operativo. Un Sistema Operativo es un grupo de programas que controlan los recursos del Computador. Un Sistema Operativo funciona como una **Maquina Extendida o Maquina Virtual**, es decir permite que el usuario pueda ejecutar programas de tal manera que no tenga que enterarse de los complejos detalles del Hardware. Desde el punto de vista de administrador de los recursos del Hardware, el Sistema Operativo es el encargado de la asignación ordenada y controlada de los procesadores, memorias y dispositivos de entrada y salida para varios programas que compiten por ellos.

1.2.2 Sistema Operativo Linux. Linux es un Sistema Operativo que a diferencia de otros, ha sido desarrollado por miles de programadores alrededor del mundo. Trabaja bajo otra filosofía de desarrollo de software, la de software libre, que expresa que los programas son una forma de expresión de ideas y que las ideas como la ciencia son propiedad de la humanidad, por esta razón el software libre expone su código fuente de sus programas a quienes desean verlo, modificarlo o copiarlo. Linux pertenece al software con licencias GPL, de sus siglas en ingles “General Public License” Licencia General Pública.

1.2.3 GPL(General Public License). La licencia Pública General pretende garantizar la libertad de compartir y modificar software libre. Este tipo de licencias están diseñadas para asegurarse de tener libertad de distribuir copias de software libre con su código fuente, de

conseguirlo o modificar fragmentos o usarlos para crear nuevos programas. Para proteger sus derechos se necesitan algunas restricciones que se traducen en ciertas obligaciones que se debe cumplir si alguien distribuye copias del software o si lo modifica. La protección de estos derechos se hace con la combinación de dos medidas:

1. Se establece un Copyright
2. Se le asigna una licencia GPL.

Las condiciones para la copia, distribución y modificación se exponen a continuación.

Términos y condiciones para la copia, distribución y modificación

Los Términos y condiciones para la copia, distribución y modificación de cualquier programa o parte de él, son aplicables a aquellos que contengan una nota de Copyright diciendo que puede ser distribuido bajo los términos de la Licencia Publica General.

Se puede copiar y distribuir copias exactas del código fuente en cualquier medio, teniendo en cuenta que debe existir de forma visible en cada una de ellas un anuncio de Copyright y un repudio de garantía. Se puede cobrar un precio por el acto físico de transferir una copia y ofrecer garantía a cambio de honorarios.

Se puede crear un nuevo programa basado en otro, copiarlo y distribuirlo con las siguientes condiciones: Los archivos modificados deben llevar de forma visible un aviso de que estos han sido modificados con su respectiva fecha de modificación. Además el nuevo programa y las partes de que deriva deben ser licenciadas como un todo.

Se puede copiar o distribuir el programa o un trabajo basado en él, según se especifica en el anterior párrafo, como código objeto o en formato ejecutable según los anteriores términos a demás de cumplir las siguientes condiciones: incluir el código fuente completo en formato electrónico o acompañarlo de una oferta por escrito válida durante al menos tres años, de proporcionar una copia en formato electrónico con un costo no mayor que el de realizar físicamente la distribución.

No se puede copiar, modificar, sublicenciar o distribuir el programa excepto como prevé expresamente la licencia GPL.

No es obligación aceptar la licencia GPL, en tal caso, cualquier acción que en ella se redacta estará prohibida por la Ley, de tal manera que si se modifica o distribuye el programa o cualquier trabajo basado en él, esta indicando la aceptación de la Licencia.

AUSENCIA DE GARANTÍA

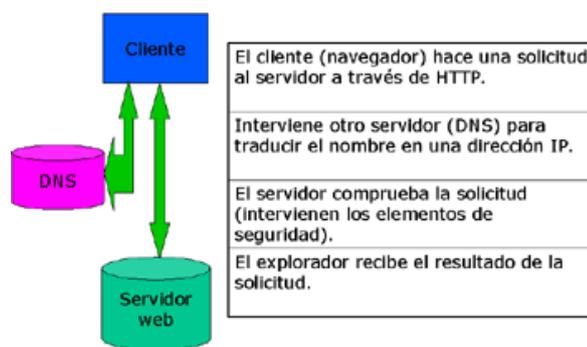
No se ofrece ninguna garantía sobre el programa. Excepto cuando se indique de otra forma por escrito.

1.2.4 Servidores Web. Un servidor Web es un programa de aplicación que satisface las solicitudes HTTP realizadas por los navegadores. Los Servidores Web deben soportar los protocolos estándar de Internet por ejemplo HTTP(Protocolo de Transferencia de Hipertexto) que facilita el intercambio de datos entre el Servidor Web y el navegador, adicionalmente debe

ofrecer soporte a scripts y lenguajes más comunes utilizados en aplicaciones de Internet, como Java, PHP entre otros, finalmente debe contener algunos elementos de seguridad.

Actualmente existen muchos Servidores Web para diferentes Sistemas Operativos, algunos con versiones para dos o más Sistemas entre los que están: Apache que actualmente tiene versiones para Windows, Unix, Linux, Novell entre otros, IIS (Internet Information Server), Xitami Web Server, Sambar Server, etc. La Figura 2 muestra el esquema de comunicación entre el Cliente y el Servidor Web.

Figura 2. Esquema de comunicación entre el cliente y el servidor



1.2.5 Servidor Web Apache. Es uno de los Servidores Web con Licencia GPL más utilizados en el mundo. Se basó en uno de los Servidores más populares el NCSA HTTPD 1.3 creado por Rob McCool de la National Center for Supercomputing Applications, de la Universidad de Illinois en Urbana-Champaign. Apache es un servidor Flexible, rápido y eficiente. Debido a que es software libre, está continuamente actualizándose gracias a miles de desarrolladores por Internet, se basa en el protocolo HTTP 1.1 aunque implementa las últimas actualizaciones. Apache es un servidor modular, es decir, puede ser adaptado a diferentes entornos y necesidades con los módulos de apoyo y con la API, Interfaz de Programa de Aplicación o de Programación de Aplicaciones. Además soporta transacciones seguras

SSL(Secure Socket Layer) y Host Virtuales, que permiten tener más de un sitio WEB en el mismo Host servidor.

1.2.6 Servidor de Nombre de Dominio (DNS). Un servidor DNS es el software que permite acceder a un dominio en Internet. Básicamente su función es atender a las peticiones hechas por los distintos programas que acceden a Internet y resolver la dirección IP asociada al dominio (o nombre) consultado. El sistema DNS es una base de datos distribuida mantenida por miles de servidores DNS, cada una de los cuales es responsable de su “Zona de Internet”. Cuando un programa cliente hace una petición de una dirección Internet, el Servidor DNS procesa la consulta, intentando buscar el dominio en su tabla de registros, sino lo encuentra envía la petición a otro servidor DNS situado en un nivel superior de la jerarquía de nombres de dominios. Esta secuencia de peticiones se repite hasta que se obtiene la dirección IP del computador que corresponde al dominio consultado.

1.2.7 Servidor FTP. Un servidor FTP es un programa que se ejecuta en un computador que da acceso a usuarios remotos a ciertos recursos del equipo, utiliza el Protocolo de Transferencia de Archivos FTP(File Transfer Protocol), por sus siglas en Ingles. FTP es una aplicación TCP/IP que se usa para enviar o recibir archivos a través de Internet

1.2.8 Servidor de Correo. Un servidor de correo es un programa encargado de monitorizar y resolver las peticiones por los protocolos de envío y recepción de mensajes electrónicos en Internet. Entre los más comunes están: POP3 para el envío y SMTP para la recepción de correo electrónico; POP3 Protocolo de Oficina de Correos (Post Office Protocol), por sus siglas en Ingles, es el encargado de recibir los correos y almacenarlos en el cliente si este lo

desea. SMTP (Simple Mail Transfer Protocol) Protocolo Simple de Transmisión de Correo, se encarga del envío de los mensajes.

1.2.9 Servidor de Correo Qmail. Qmail es un servidor de correo compuesto por un grupo de programas que se ejecutan bajo UID/GID (Identificación de Usuario/Identificación de Grupo) diferentes. Es un Agente de Transporte de Correo o MTA (Agent Mail Transfer), para sistemas Unix y utiliza el SMTP para intercambiar mensajes con MTA de otros sistemas. Qmail cuenta con las siguientes características:

Configuración.

- A. Se Adapta automáticamente de Unix a Linux o viceversa
- B. Instalación rápida: No hay una larga lista de decisiones que tomar

Seguridad.

- A. Reducción al mínimo del código que se ejecuta con setuid, impidiendo la identificación de los UID de los usuarios.
- B. Reducción al mínimo del código que se ejecuta como root

Construcción de mensajes

- A. Cumple con los RFC 822 y RFC 1123: Formato estándar Internet para cabeceras de mensajes de correo electrónico y requerimientos para servidores de Internet.
- B. Soporte Completo para grupos de Direcciones
- C. Conversión automática de listas de direcciones en formato antiguo al formato RFC 822

- D. Orden Sendmail para la compatibilidad con los programas clientes de correo de los usuarios, que permite al usuario seguir funcionando con su cliente de correo sin hacer ninguna modificación.
- E. Longitud de línea de la cabecera limitada solo por la memoria disponible
- F. Enmascaramiento de máquina, que se refiere a poder recibir correo a una máquina que no tenga una dirección de correo real.
- G. Enmascaramiento de Usuario

Servicio SMTP

- A. Cumple con RFC 821, RFC 1123, RFC 1651, RFC 1652, y RFC 1854
- B. Control de Relay: Acaba con los reenvíos o retransmisiones que hacen usuarios externos sin autorización
- C. Reconocimiento automático de direcciones IP locales
- D. Caducidad del plazo de envío individual para cada bufer o almacenamiento temporal
- E. Hop Counting: Limitación del número de cabeceras Received que tiene un mensaje, que evita el recibir correo proveniente de varios saltos anteriores

Gestión de Colas

- A. Tratamiento instantáneo de los mensajes que se agregan a la Cola
- B. División del directorio de la Cola: el directorio de cola de correos se divide a su vez en subdirectorios con el objetivo de mejorar el desempeño ya que clasifica los archivos según cierta situación por ejemplo: archivos con errores permanentes en la entrega, archivos de bloqueo, etc.,

- C. Programación cuadrática de reintentos: Cuanto más antiguo es el mensaje, con menor frecuencia se intenta su reenvío
- D. Planificación independiente de reintentos para mensajes
- E. Encolamiento automático seguro: No se pierde el mensaje si el sistema falla
- F. Limpieza automática para colas de correo
- G. Estadísticas de entrega detalladas

Bounces (Devoluciones).

- A. Mensajes de Devolución (bounce) legibles para la máquina y legibles para el usuario
- B. Soporte HCMSSC: Códigos de error según RFC 1893 independientes del idioma
- C. Devoluciones reiteradas enviadas al administrador de correo, permite que el administrador tome decisiones respecto a direcciones que al parecer ya no existen.

Rutados por Dominio.

- A. Cualquier número de nombres para la máquina local
- B. Cualquier número de dominios virtuales, permite tener en un mismo servidor varios dominios que pueden servir por ejemplo para diferenciar secciones de la misma empresa.
- C. Comodines para los dominios: Subdominios enteros pueden gestionarse con una única instrucción de ruteado

Entrega SMTP.

- A. Cumple con RFC 821, RFC 974, RFC 1123
- B. La transmisión a un servidor que no responde se detiene o acelera automáticamente

- C. Encaminamiento artificial, a través de smarthost; el correo es enviado al ISP ideal para conexiones PPP y mailertable para conexiones UUCP que mantiene una tabla de enrutamiento.
- D. Cola SMTP pasiva: Ideal para SLIP/PPP

Reenvío de mensajes y listas de correo.

- A. Compatibilidad con archivos .forward de Sendmail
- B. Compatibilidad con el archivo /etc/aliases de Sendmail, que contiene bases de datos de reenvíos
- C. Comodines para direcciones
- D. Administradores de listas de correo: Desvío automático de devoluciones y mensajes de ausencia por vacaciones
- E. VERP: Automatic Recipient identification for mailing list bounces (Identificación automática del destinatario para devoluciones de listas de correo)
- F. Cabeceras Deliverd-To: Prevención automática de bucles incluso entre máquinas, evita que algún correo se reenvíe, de tal manera que forme un círculo cerrado.

Servicio POP3.

- A. Cumple con RFC 1939
- B. Soporte UIDL: Comando para visualizar los mensajes actuales con su identificación
- C. Soporte TOP: Protocolo de autenticación para correo POP3
- D. Comprobación modular de contraseñas: Permite que utilizar un programa externo para la comprobación de las contraseñas antes de permitir la lectura de los correos

1.2.10 Bases de Datos. Una Base de Datos se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada. Desde un punto de vista informático una Base de Datos es un sistema formado por un conjunto de datos almacenados en disco que permiten el acceso directo a ellos y un conjunto de programas que manipulen esos datos.

El motor es el encargado de administrar y hacer funcional esa base de datos, gestiona el almacenamiento físico, atiende las peticiones de consulta, garantiza la integridad y consistencia de los datos y gestiona la atomicidad transaccional.

1.2.11 Motor de Bases de Datos Postgresql. Postgresql es un motor de Bases de Datos de tipo objeto-relacional diseñado para Linux con licencia GPL y código Abierto, cuenta con un servidor de Bases de Datos, un cliente y varias herramientas adicionales, tiene las mismas o mejores características con las que cuentan motores tan reconocidos como Oracle, Informix, Sybase.

Al igual que muchos proyectos de código abierto, Postgresql es desarrollado por un gran número de programadores que utilizan la Internet como el medio para discutir, acordar y enviar las mejoras que lo han convertido en el líder en su área.

A diferencia del software comercial, Postgresql es mejorado cada 3 ó 5 meses, con nuevas características, con menos errores o más acorde a las normas de SQL92 (Lenguaje de Consultas Estructuradas, Estándar establecido por la ISO y ANSI). El código fuente fundamentalmente está desarrollado en lenguaje C. Postgresql soporta:

Transacciones atómicas. Todos los eventos para realizar una acción deben tomarse como un solo, si alguno de ellos falla, falla toda la transacción y todo queda como estaba antes de realizarse la operación (deshaciendo todos los pasos adelantados).

Triggers o Disparadores. Son sentencias que se disparan o ejecutan después de que se cumple cierta condición.

Foreigns Keys. Una llave foránea es un atributo (puede ser compuesto) de una tabla o relación cuyos valores deben de concordar con los de una llave primaria de otra tabla.

Replicación. Postgresql también soporta conectividad ODBC para ser utilizado con aplicaciones Windows y JDBC para Java, lo que significa que se puede hacer cualquier aplicación en Windows o Java y conectar al servidor de Postgresql. Además de estas características también cuenta con otras particulares como:

Herencia. Una clase o tabla puede estar inherente en otra es decir que una tabla puede heredar los atributos de otra por ejemplo:

```
CREATE TABLE ciudades (  
nombre text,  
poblacion float,  
altitud int )  
);
```

```
CREATE TABLE capitales (  

```

```
departamento char(2)
) INHERITS (ciudades);
```

En este caso, la tabla capitales hereda los atributos de la tabla ciudades, una consulta para encontrar todas las ciudades incluyendo las capitales cuya altitud sea mayor a 500 sería la siguiente:

```
SELECT c.nombre c.altitud
FROM ciudades* c
WHERE c.altitud > 500;
```

Aquí el “*” indica que la consulta debe realizarse sobre ciudades y todas las clases o tablas que estén por debajo de ella en la jerarquía de herencia.

Valores No Atómicos. Uno de los principios del modelo relacional es que los atributos de una tabla son atómicos. PostgreSQL no posee esta restricción; los atributos pueden contener sub-valores que pueden accederse desde el lenguaje de consulta.

Vectores y Arreglos. PostgreSQL permite que los atributos de una tabla sean definidos como vectores multidimensionales de longitud fija o variable, puede crear vectores de cualquiera de los tipos base o de tipos definidos por el usuario ejemplo:

```
CREATE TABLE empleado (
nombre          text,
saltrimestrales int4[],
agenda         text[] []
);
```

La sentencia de arriba creará una tabla llamada empleado con una cadena del tipo “text”, un vector unidimensional del tipo “int4” y un vector bidimensional del tipo “text”.

Control de Concurrencia Multi-Versión MVCC. Es una técnica avanzada para mejorar el servicio de una base de datos en un entorno multiusuario. A diferencia de otros sistemas de bases de datos que usan bloques para el control de concurrencia, PostgreSQL mantiene la consistencia de los datos con un modelo Multi-Versión. Esto significa que mientras se consulta una base de datos, cada transacción ve una imagen de los datos (una versión de la base de datos) como si fuera tiempo atrás, sin tener en cuenta el estado actual de los datos que hay por debajo. Esto evita que la transacción vea datos inconsistentes que pueden ser causados por la actualización de otra transacción concurrente en la misma fila de datos, proporcionando aislamiento transaccional para cada sesión de la base de datos.

Aislamiento Transaccional. El estándar ANSI/ISO SQL define cuatro niveles de aislamiento transaccional en función de tres hechos que deben ser tenidos en cuenta entre transacciones concurrentes. Estos hechos no deseados son:

Lecturas “sucias”. Una transacción lee datos escritos por una transacción no esperada, no cursada.

Lecturas no repetibles. Una transacción vuelve a leer datos que previamente había leído y encuentra que han sido modificados por una transacción cursada.

Lectura “fantasma”. Una transacción vuelve a ejecutar una consulta, devolviendo un conjunto de filas que satisfacen una condición de búsqueda y encuentra que otras filas que satisfacen la condición han sido insertadas por otra transacción cursada.

Los cuatro niveles de aislamiento consignados en la Tabla 1 y sus correspondientes acciones se describen a continuación.

Tabla 1. Niveles de Aislamiento y acciones

	Lectura “Sucia”	Lectura no repetible	Lectura “fantasma”
Lectura cursada	No Posible	Posible	Posible
Lectura Serializable	No. Posible	No. Posible	No. Posible

Nivel de Lectura Cursada. Lectura cursada es el nivel de aislamiento por defecto en PostgreSQL. Cuando una transacción se ejecuta en este nivel, la consulta sólo ve datos cursados antes de que la consulta comenzara y nunca ve datos "sucios" ni los cambios en transacciones concurrentes cursados durante la ejecución de la consulta.

Si una fila devuelta por una consulta mientras se ejecuta una declaración UPDATE (o DELETE, o SELECT FOR UPDATE) está siendo actualizada por una transacción concurrente no cursada, entonces la segunda transacción que intente actualizar esta fila esperará a que la otra transacción se curse o pare. En caso de que pare, la transacción que espera puede proceder a cambiar la fila. En caso de que se curse (y si la fila todavía existe, por ejemplo, no ha sido borrada por la otra transacción), la consulta será reejecutada para esta fila y se comprobará que la nueva fila satisface la condición de búsqueda de la consulta. Si la nueva versión de la fila satisface la condición, será actualizada (o borrada, o marcada para ser actualizada).

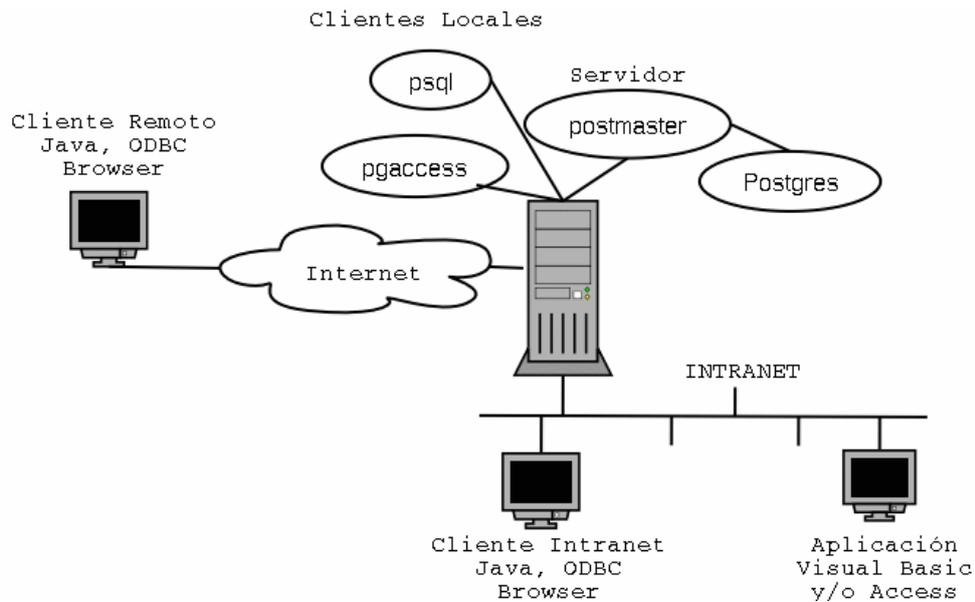
Nivel de Aislamiento Serializable. La serialización proporciona el nivel más alto de aislamiento transaccional. Cuando una transacción está en el nivel serializable, la consulta sólo ve los datos cursados antes de que la transacción comience y nunca ve datos sucios ni los cambios de transacciones concurrentes cursados durante la ejecución de la transacción. Por lo tanto, este nivel emula la ejecución de transacciones en serie, como si las transacciones fueran ejecutadas una detrás de otra, en serie, en lugar de concurrentemente.

Si una fila devuelta por una consulta durante la ejecución de una declaración UPDATE (o DELETE, o SELECT FOR UPDATE) está siendo actualizada por una transacción concurrente no cursada, la segunda transacción que trata de actualizar esta fila esperará a que la otra transacción se curse o pare. En caso de que pare, la transacción que espera puede proceder a cambiar la fila. En el caso de una transacción concurrente se curse, una transacción serializable será parada con el mensaje “ERROR: Can't serialize access due to concurrent update”; porque una transacción serializable no puede modificar filas cambiadas por otras transacciones después de que la transacción serializable haya empezado.

Arquitectura. Postgresql implementa una arquitectura Cliente/Servidor lo que facilita su administración, configuración y mejora su desempeño. Figura 3

En el servidor, Postgresql cuenta con un proceso postmaster, que se encarga de recibir las conexiones y cada vez que recibe una, crea una instancia del motor Postgresql, para procesar todos los requerimientos del cliente.

Figura 3. Arquitectura de Postgresql.



En el cliente, se cuenta con una interfaz de comandos llamada psql, que le permite al usuario ejecutar cualquier sentencia SQL, así como administrar la Base de Datos. Además se cuenta con una interfaz gráfica, pgaccess, que permite realizar las funciones de administración, crear funciones, formas y reportes.

Postgresql cuenta con interfaces de programación para todos los lenguajes: C, C++, Perl, PHP, Python y Java.

Postgresql permite la integración con aplicaciones Windows, ya que soporta ODBC. Con ODBC se pueden desarrollar aplicaciones cliente Windows (VB, FoxPro, Access etc.), que interactúen con un servidor Linux con Postgresql; Existen utilidades que permiten, por

ejemplo, desde una base de datos Access se pueden importar tablas desde un servidor Postgresql.

1.2.12 Paginas Web. Las paginas Web son el alma de la Internet, por medio de ellas se puede obtener todo tipo de información, una pagina Web está diseñada para informar, entretener, trabajar etc.. Para elaborar una pagina Web simplemente es necesario tener a disposición un editor de texto y conocer el HTML (HyperText Markup Language), aunque hoy existen innumerables programas con los cuales el usuario no tiene la necesidad de conocerlo. Entre los programas más conocidos se están: Macromedia Dreamweaver, Macromedia Fireworks, Macromedia Flash. Estos tres programas se complementan para lograr el diseño de paginas Web cada vez más profesionales.

1.2.13 Paginas Dinámicas. Son páginas que cambian su contenido, es decir, páginas que contienen partes vivas que hacen que se vean diferentes cada vez que se accede a ellas. Es importante para una página atraer visitantes y una forma de conseguirlo es evitando que los usuarios siempre vean la misma información.

Una página típica son las de respuestas a los formularios. Es posible crear un script que interprete la información recibida de un formulario y hacer que genere otra página resultado donde se muestren cálculos o simples mensajes informativos que cambien en función de los datos de entrada.

Las páginas dinámicas son creadas en el momento en que son referenciadas por el usuario. Si bien tienen un estilo base, la información desplegada en las mismas es dinámica; son interactivas, ya que permiten que la página a visualizar pueda ser creada con base en la

información creada por el usuario. Las páginas dinámicas permiten interactuar con una base de datos, por lo que son una poderosa herramienta para favorecer los negocios de la empresa. De esta forma la actualización se realiza automáticamente, ya que al acceder a la página se accede a la base de datos con la información actualizada.

Existen muchos lenguajes de programación especializados en el diseño de páginas dinámicas entre los que tenemos ASP, PHP, PERL.

ASP(Active Server Pages). Es un lenguaje derivado de Visual Basic desarrollado por Microsoft. Evidentemente su empleo se realiza sobre plataformas funcionando bajo sistemas Windows NT.

PERL. Practical Extraction and Report Language, algo así como lenguaje práctico de extracción y de informes. Es un lenguaje creado por *Larry Wall* con el objetivo principal de simplificar las tareas de administración de un sistema UNIX; en realidad hoy en día se ha convertido en un lenguaje de propósito general, y una de las principales herramientas de un buen webmaster.

Es un lenguaje que hereda estructuras principalmente 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 máquina nativo del ordenador y sistema operativo en el que se ejecuta.

Aunque desarrollado originalmente en un entorno UNIX, actualmente hay versiones para casi todos los sistemas operativos: DOS (OpenDos) Windows NT, Amiga, MacOS. Los scripts son compatibles entre las diversas plataformas, de forma que es un verdadero lenguaje multiplataforma. Muchos fabricantes lo incluyen en sus versiones de UNIX; también el Linux lo incluye. Es un lenguaje rápido y potente que requiere una etapa de aprendizaje más amplio, esta reservado para personas ya familiarizadas con la programación.

1.2.14 PHP. Lenguaje diseñado en primera instancia para el sistema operativo Linux, aunque hoy también se lo encuentra bajo Windows. PHP es un lenguaje de programación de estilo clásico, es decir es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc.,

PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML (Wireless Markup Language), un lenguaje de programación en línea de XML (Extensible Markup Language). WML permite que partes de texto de las páginas Web sean adaptadas a dispositivos de mano.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. La Figura 4 muestra el esquema de procedimientos en la conexión de un cliente con el Servidor Web con soporte PHP.



Figura 4. Esquema conexión paginas PHP

2. DESARROLLO DE ACTIVIDADES

2.1 ADQUISICION DE INFORMACION Y SOFTWARE

Las primeras actividades realizadas fueron la adquisición de la mayor cantidad de información relacionada con el proyecto, distribuciones de Linux existentes, configuraciones, servidores WEB, FTP, Correo, DNS motores de bases de datos, lenguajes de programación que trabajen bajo plataforma Linux y la obtención del software para cada uno de estos tópicos. En el **Anexo E. Referencias Internet**, se encuentran las direcciones de Internet donde fue obtenido el software.

2.2 INSTALACION Y CONFIGURACION DE LINUX

La siguiente actividad realizada fue la elección e instalación de la distribución del sistema operativo a utilizar. Hoy en día existen muchas distribuciones, como RedHat, Mandrake, Debian, Suse entre otras. Se escogió la distribución de RedHat, porque de ella se encontró más información, tiene desarrollado más software para diferentes utilidades y porque algunos Ingenieros que ya habían trabajado con ella la recomiendan.

El primer paso fue la obtención del código fuente de la última versión **RedHat 7.0**, desde la página Web oficial de RedHat hacia el Disco Duro, lo siguiente fue pasar los instaladores a CDs para facilitar la instalación.

Las especificaciones del equipo donde se instaló se presentan en la Tabla 2.

Procesador	Pentium II de 466 Mhz
Disco Duro	10 Gbytes
Memoria RAM	64 Mbytes
Tarjeta de Red	Sis 10 Base T

Tabla 2. Características del Equipo de Trabajo

Se debe tener en cuenta que el equipo en el cual se va a instalar tenga posibilidad de arranque desde CD, en caso contrario, la instalación debe hacerse después de haber iniciado el equipo ya sea con un disco de arranque o con un sistema pre-instalado que puede ser DOS o Windows 9x. En este caso el equipo cuenta con arranque desde CD.

Existen tres formas de instalar el sistema: instalación gráfica, texto, rescate; la primera como su nombre lo indica, permite hacer una instalación de forma gráfica, con ventanas y objetos, la segunda solamente varía en la forma de presentar las opciones, y la tercera sirve para poder recuperar sistemas que por algún motivo se hayan dañado. Para efectos de estudio se hizo la instalación de las dos primeras formas donde no se encontró ninguna diferencia sustancial.

Para instalar Linux lo primero que se hace después de arrancar el equipo desde los CDs de instalación es determinar las particiones en que se dividirá el disco duro, El Sistema Operativo

Linux puede contener dos o más particiones (root y swap), pero al utilizar solamente estas dos se corre el riesgo de perder toda la información almacenada en root, si sobre ésta ocurre algún daño, así que lo recomendable es tener tantas particiones como tipos de información se vaya a manejar, en este caso se realizaron las siguientes particiones.

Swap o Intercambio. La partición de Swap o de intercambio se usa para soportar la memoria virtual, es decir la cantidad de espacio de disco que se utilizará para la simulación de memoria física en la ejecución de los programas. El Kernel del Sistema Operativo controla las tareas de intercambio de los componentes de un proceso entre la memoria virtual (en el disco duro) y la memoria física. Es recomendable establecer su tamaño como el doble de la memoria **RAM** que contenga el equipo. En este caso se utilizó 128 Mbytes.

Partición Raíz. La partición Raíz es donde reside el directorio raíz (/), punto de partida dentro de la estructura de archivos (File System) que establece y maneja el Sistema Operativo. La partición root contiene todos los archivos de arranque y configuración del sistema, por lo que su tamaño no debe ser muy grande. En este caso se determinaron 500 Mbytes para esta partición.

Partición /home. Aquí se encuentran los directorios home de cada usuario, su tamaño depende del número de usuarios que se desee tener. En este caso se utilizó 1 Gbyte.

Partición /usr. La partición **/usr** es donde residirá gran parte del software del sistema, su tamaño depende de la cantidad de software que se desea instalar. En este caso se utilizó el espacio libre que quedaba.

Luego de establecer las particiones necesarias se procede a formatearlas. También existe la posibilidad de hacer un chequeo físico de cada partición con la opción de marcar bloques dañados.

Dentro del proceso de instalación se continúa con la instalación de los dispositivos Hardware, que se hace automáticamente. En el caso de la tarjeta de red sólo se debe especificar: nombre del equipo y dirección IP, máscara de subred, servidor DNS principal y/o secundario y puerta de enlace. En este caso.

Nombre del Equipo:	telepaso
Dirección IP:	200.21.84.252, dirección real otorgada por Telecom
Máscara de Subred:	255.255.255.0
Servidor DNS Principal:	200.21.64.2, Telecom Cali
Servidor DNS Secundario:	200.21.200.2, Telecom Bogotá
Puerta de Enlace:	200.21.84.1 Enrutador salida a Internet

Finalmente se hace la elección del software a instalar. En este caso se hizo la selección de todo el software disponible.

Luego de tener funcionando Linux RedHat 7.0, se realizaron pruebas sobre programas y aplicaciones, instalándolos, configurándolos y desinstalándolos para observar el rendimiento. En este proceso se notó que muchas veces el equipo se bloqueaba sin razón aparente, es decir que presentaba inestabilidad. Por este motivo se procedió a desinstalar completamente el Sistema Operativo y a probar otra versión de Linux RedHat que para entonces había lanzado:

su nueva distribución, **RedHat 7.1**. Se procedió a conseguir esta nueva versión y se realizaron todas las actividades anteriores. Las pruebas con esta nueva versión resultaron satisfactorias y el sistema demostró una alta confiabilidad, por lo cual se decidió mantener esa versión para continuar con el desarrollo del proyecto.

2.3 CONFIGURACIÓN SERVIDOR DNS

La resolución de nombres en equipos Linux puede ser determinada de dos formas; a través de una tabla estática definida en el archivo **“host”** ubicado en el directorio **“/etc”** y/o a través de la configuración del Servidor DNS. Cabe destacar que se puede especificar los dos tipos dándoles un orden de importancia; este orden se debe especificar en el archivo **“host.conf”** ubicado en el directorio **“/etc”**. Esta última opción fue la implementada para este proyecto. Para esto es necesario el software llamado **BIND (Berkeley Internet Name Domain)**, que viene con la distribución de Linux RedHat.

Archivos de configuración. Un servidor DNS cuenta con archivo de configuración principal llamado **“named.conf”**, que contiene la ubicación de los demás archivos de configuración, conocidos como archivos de **“zona”**. De esta manera el archivo de configuración principal tiene dos tipos de secciones:

Opciones “options”. Indica el directorio donde se encuentran otros archivos de configuración y algunas otras opciones.

Zonas “zone”. Puede existir varias zonas por archivo, estas zonas definen los Dominios (telecom.com.co) y las redes o subredes (200.21.84.0) sobre los que se mantiene información de equipos. Los diferentes tipos de zonas se presentan a continuación:

Zonas de Dominio. Declaran el dominio sobre el cual está el servidor. Estas zonas contienen dos parámetros; **type** y **file**. El primero puede llevar dos valores **master** o **slave**; **master** significa que el Servidor DNS contiene la información principal sobre el dominio, **slave** significa que el Servidor DNS es de respaldo en caso de que el principal falle. El segundo parámetro indica el nombre del archivo que contiene los parámetros específicos de la zona.

Zonas de Redes. En estas zonas se deben expresar todas las subredes que compongan el sistema, al igual que las zonas de dominio; también pueden ser **master** o **slave**.

Zonas de Raíz (“.”). Esta zona de raíz es una zona especial que especifica el archivo que contiene información de los servidores raíz mundiales, que contienen toda la información sobre las redes que existen en Internet. Este archivo ya viene configurado con la distribución del software y por lo general es llamado **“named.ca”** cuyo contenido es el siguiente:

El archivo **named.conf** del Servidor DNS para el proyecto quedo así:

```
options {
    directory "/var/named/";
};
zone "." {
    type hint;
    file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
```

```

        type master;
        file "0.0.127.in-addr.arpa.zone";
};
zone "84.21.200.in-addr.arpa" {
    type master;
    file "84.21.200.in-addr.arpa.zone";
};
zone "localhost" {
    type master;
    file "localhost.zone";
};
zone "telecom.com.co" {
    type master;
    file "telecom.com.co.zone";
};

```

Luego de tener configurado el archivo **named.conf**, se debe crear los dos archivos que tendrán la configuración de las diferentes zonas establecidas anteriormente, estos se deben crear dentro del directorio **named**.

El nombre y contenido de cada archivo se especifica a continuación.

Nombre del archivo: telecom.com.co.zone

Contenido

```

$TTL 86400
@      IN      SOA  telepasto,telecom.com.co  root.telepasto.telecom.com.co (
        2 ; serial
        28800 ; Lapso de tiempo de comunicación de un servidor slave con un Servidor master
        7200 ; Tiempo que debe pasar para reintentar comunicación
        604800 ; Tiempo para cancelar los intentos
        86400 ; Tiempo de validez de la resolución enviada
)
@      IN      NS   telepasto.telecom.com.co
telepasto      IN      A       200.21.84.252
jairo          IN      A       200.21.84.253
patricia       IN      A       200.21.84.250

```

Nombre del archivo: 84.21.200.in-addr.arpa.zone
Contenido

```
$TTL 86400
@      IN      SOA    telepasto,telecom.com.co  root.telepasto.telecom.com.co (
                          2 ; serial
                          28800 ; refresh
                          7200 ; retry
                          604800 ; expire
                          86400 ; ttl
                          )
@      IN      NS     telepasto.telecom.com.co.
250    IN      PTR    patricia.telecom.com.co.
252    IN      PTR    telepasto.telecom.com.co.
253    IN      PTR    jairo.telecom.com.co.
```

Finalmente se debe arrancar el demonio que inicia el servicio, en este caso el demonio “named” que se encuentra en el directorio “/etc/init.d/rc.d/”

```
“# /etc/rc.d/init.d/named start”
```

2.4 CONFIGURACION DEL SERVIDOR FTP WU-FTPD

La versión de Linux RedHat 7.1 ya trae compilado el Servidor FTP por defecto, pero es necesario configurar algunos detalles, los archivos de configuración se encuentran dentro del directorio “/etc”.

Usuarios de Linux No Permitidos. El archivo **ftpusers** contiene una lista de los usuarios de Linux que no tienen acceso al Servidor FTP, entre los que debe estar el usuario root, debido a que al ser el root el administrador, éste tiene la posibilidad de acceder a todo nuestro disco duro lo cual obviamente no es lo conveniente.

Estructura de directorios y permisos. El Servidor FTP está compuesto por el siguiente Arbol de directorios (Figura 5), con origen en un directorio existente previamente seleccionado, observando algunas consideraciones de seguridad (como por ejemplo la no elección del directorio root del sistema).

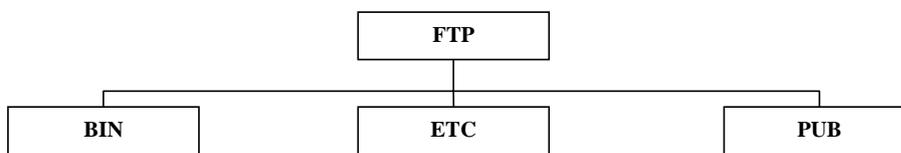


Figura 5. Arbol de Directorios Servidor FTP

El usuario **root** debe ser el propietario del directorio **FTP** y éste debe tener acceso de escritura prohibido, pero sí de ejecución para el resto de usuarios. El directorio **BIN**, también debe ser de propiedad de **root** y sin acceso de escritura a todos los usuarios, este contiene todos los archivos ejecutables. El directorio **ETC** del **root** y con permiso sólo de ejecución contendrá archivos de configuración. **PUB** contiene todos los archivos que se van a compartir con los usuarios; este directorio debe ser de propiedad del usuario **ftp** con permisos de ejecución y lectura.

2.5 INSTALACION Y CONFIGURACION DE APACHE

Lo primero que se hizo fue descomprimir el fuente en un directorio temporal **“/tmp”**, mediante los comandos:

```
“# cd /tmp”
```

```
“# tar -zxvf /tmp/apache_1.3.19.tar.gz”
```

Luego se estableció la configuración que tendrá el Servidor, en este caso se identifica el directorio donde será instalado el **Apache** mediante la cláusula `--prefix=/usr/local/apache`, y se debe activar el módulo para permitir que el Servidor reconozca y ejecute paginas **php**, en este caso `--activate-module=/usr/php/lib/libphp4.a`. El comando completo queda así:

```
“# configure --prefix=/usr/local/apache --activate-module=/usr/php/lib/libphp4.a”
```

El siguiente paso fue la compilación e instalación que se logró con los comandos.

```
“# gmake”
```

```
“# gmake install”
```

2.6 INSTALACION Y CONFIGURACION DE QMAIL

La siguiente actividad fue la instalación y configuración del Servidor de Correo **Qmail**. Antes de continuar con la instalación se hizo necesario conseguir un software adicional como complemento para tener un Servidor de correo completo: **Ucspi-tcp** y **daemontools**. El primero, es un grupo de herramientas para la elaboración de aplicaciones Cliente/Servidor TCP. El segundo es una herramienta para la supervisión y gestión de servicios Unix. Después de haber conseguido este software se procedió a descomprimirlo en el directorio temporal

“/tmp” de la siguiente manera:

```
“# cd /tmp”
```

```
“# tar -zxvf /tmp/qmail-1-03.tar.gz”
```

```
“# tar -zxvf /tmp/ucspi-tcp-0.84.tar.gz”
```

```
“# tar -zxvf /tmp/daemontools-0.61.tar.gz”
```

Al hacer esto, se crearon 3 directorios llamados **qmail-1.03**, **ucspi-tcp-0.84** y **daemontools-061**. A continuación se creó el directorio principal donde se alojará **qmail**, este se debe crear en el directorio **/var**.

```
“# mkdir /var/qmail”
```

A continuación se crearon algunos links simbólicos con el propósito de organizar de mejor manera la instalación.

```
“# ln -s /usr/man /var/qmail/man”
```

```
“# mkdir /etc/qmail”
```

```
“# ln -s /etc/qmail /var/qmail/control”
```

```
“# ln -s /usr/sbin /var/qmail/bin”
```

El siguiente paso, es la creación de los grupos y usuarios propietarios del Servidor, para esto se creó un script de shell que se lo llamó **grpusua.cr** cuyo contenido es el siguiente:

```
pw groupadd nofiles;
pw useradd alias -g nofiles -d /var/qmail/alias -s /nonexistent;
pw useradd qmaild -g nofiles -d /var/qmail -s /nonexistent
pw useradd qmail -g nofiles -d /var/qmail -s /nonexistent
pw useradd qmailp -g nofiles -d /var/qmail -s /nonexistent
pw groupadd qmail
pw useradd qmailq -g qmail -d /var/qmail -s /nonexistent
pw useradd qmailr -g qmail -d /var/qmail -s /nonexistent
pw useradd qmails -g qmail -d /var/qmail -s /nonexistent
```

El comando **“pw groupadd”** adiciona un nuevo grupo, **“pw useradd”** adiciona un nuevo usuario con la opción **“-g”** asigna el grupo al que va a pertenecer ese usuario y la opción **“-d”** le asigna un directorio.

Luego se convierte este archivo en un ejecutable y se lo ejecuta mediante los comandos.

```
“# chmod +x grpusera.cr”
```

```
“# ./grpusera.cr”
```

La siguiente actividad fue la compilación del código fuente así:

```
“# ./config”
```

A continuación se instaló las herramientas que anteriormente se habían descomprimido en el directorio `/tmp`, para eso se pasó al directorio `/tmp/ucspi-tcp-0.84`, para luego realizar la compilación del software.

```
“# cd /tmp/ucspi-tcp-0.84”
```

```
“# make”
```

```
“#make setup check”
```

También se instaló la herramienta **daemontools**, pasando primero al directorio `/tmp/daemontools-0.61` y posteriormente compilando el software.

```
“# cd /tmp/daemontools-061”
```

```
“/# make”
```

```
“/# make setup check”
```

El siguiente paso fue crear el archivo de inicio del Servidor el cual se llamó `rc` y fue creado dentro del directorio `/var/qmail`. El archivo contiene las siguientes líneas:

```
“ exec env – PATH="/var/qmail/bin:$PATH" \”
```

```
“ qmail-start “'cat /var/qmail/control/defaultdelivery'” “
```

Este archivo arranca la ejecución de **Qmail** con un parámetro que determina la forma en que se entregará los mensajes, en este caso el contenido del archivo **defaultdelivery** determina esta forma, lo que se hace a continuación es crear este archivo con el siguiente contenido:

```
“./Maildir/”
```

Finalmente se hizo ejecutable el archivo **rc** con la siguiente orden.

```
“chmod +x /var/qmail/rc”
```

Luego de tener el archivo de inicio, se procedió a crear un script de arranque y parada del servicio, es decir un **demonio**, ya que sin este **demonio** se tendría que ejecutar **Qmail** después de que el sistema haya iniciado, y parar la ejecución antes de cerrar el sistema. Este script de inicio y parada llamando **qmail** esta disponible en la URL que se presenta en el Anexo E **Referencias Internet**. Finalmente se procedió a copiar el demonio en el directorio **/etc/rc.d/init.d**, donde se encuentra todos los demonios de Linux, posteriormente se creó algunos links necesarios para la puesta en marcha del Servidor.

```
“# ln -s /etc/rc.d/init.d/qmail /etc/rc.d/rc0.d/K30qmail ”  
“# ln -s /etc/rc.d/init.d/qmail /etc/rc.d/rc1.d/K30qmail ”  
“# ln -s /etc/rc.d/init.d/qmail /etc/rc.d/rc2.d/S80qmail “  
“# ln -s /etc/rc.d/init.d/qmail /etc/rc.d/rc4.d/S80qmail “  
“# ln -s /etc/rc.d/init.d/qmail /etc/rc.d/rc5.d/S80qmail “  
“# ln -s /etc/rc.d/init.d/qmail /etc/rc.d/rc6.d/K80qmail “
```

El comando **“ln -s”** crea un enlace de un archivo o un directorio en otra ruta del disco, en este caso se crean varios enlaces del archivo de arranque a otros directorios con un nombre diferente.

2.7 INSTALACION Y CONFIGURACION DE PHP COMO MODULO DE APACHE

Se descomprimió en el directorio `/usr/local/php`, luego se compiló como modulo de Apache y con soporte para Postgresql. Esto se logra con el siguiente comando:

```
"# ./configure --with-postgresql=/usr/local/pgsql \ --with-  
apache=/usr/local/apache --enable-track-vars "  
"# gmake "  
"# gmake install "  
"# cp /usr/src/php/php.ini-dist /usr/local/lib/php.ini "
```

2.8 INSTALACION Y CONFIGURACIÓN POSTGRESQL

Luego de la obtención del código fuente del servidor de bases de datos Postgresql, realizado en la primera actividad, se precedió a su respectiva instalación y configuración que se resume en los siguientes pasos:

Crear el usuario administrador de la base de datos Postgres con permisos de ejecución, este usuario es el responsable también de la ejecución del Postmaster que es el demonio en espera de peticiones de servicios.

Luego se debe establecer algunas variables de entorno necesarias para el uso del servidor. En el directorio "home" del usuario "postgres" se edita el archivo ".bashrc" que es el archivo de

Comentario [JS1]: Es el administrador?

Comentario [JS2]: Qué contiene este archivo? O cuál es su propósito?

configuración del shell “bash” que por defecto es el utilizado por cada usuario, [cuyo contenido](#) debe depender del shell a utiliza, en este caso el shell #bash”:-

```
PATH=$PATH:/usr/local/pgsql/bin
MANPATH=$MANPATH:/usr/local/pgsql/man
PGLIB=/usr/local/pgsql/lib
PGDATA=/usr/local/pgsql/data
export PATH MANPATH PGLIB PGDATA
```

La primera línea crea un camino para los ejecutables de Postgresql, la segunda crea un camino para las páginas man (manual) que viene con la distribución del software, la tercera direcciona las librerías necesarias para la ejecución del motor, la siguiente establece un directorio en el cual se crearán las bases de los usuarios, y la última línea hace que las anteriores tengan efecto.

El siguiente paso es descomprimir y compilar el código fuente, en la carpeta /usr/local/pgsql; para esto es necesario establecer algunas características bajo las cuales va a funcionar el motor y son:

Prefix: Directorio donde se va a instalar.

With-xxx: Tecnologías soportadas, xxx puede ser ODBC, Perl etc.

El comando con estas características finalmente queda así:

```
"# ./configure --prefix=/usr/local/pgsql --with-perl --with-odbc --enable-locale "
```

Comentario [JS3]:

El comando **configure** investiga cómo está configurado el sistema y define constantes de compilación para añadir o no módulos según sea necesario; en este caso adiciona los módulos para trabajar con interfaz Perl y con conectividad ODBC.

Posteriormente se hizo la respectiva compilación con los siguientes comandos:

```
"# gmake all >& make.log &"
```

```
"# tail -f make.log"
```

Finalmente se hizo la instalación.

```
"# gmake install >& make.install.log &"
```

```
"# tail -f make.install.log"
```

El comando **gmake** es una versión del comando **make**, que realiza las tareas de compilación para obtener los archivos ejecutables a partir del código fuente, tareas que deben estar consignadas en un archivo que por defecto lleva el nombre de makefile, aunque se puede especificar otro.

El comando **tail -f** simplemente permite visualizar las últimas 10 del contenido de un archivo; en este caso se visualiza la salida que va produciendo el **gmake** y que se registra en el archivo **make.install.log**

2.9 ANALISIS DEL SISTEMA EN EL AREA DE TELEMATICA

Actualmente existe un diseño para la Base de Datos del Grupo Técnico Operativo realizado por el Ingeniero Juan Carlos Lince, pero teniendo en cuenta que en algunas secciones se han producido cambios tanto en la información como en su manejo, se decidió rediseñar la Base de Datos que tendrá el nombre de **TELEPASTO**, y para ello se hizo un nuevo análisis de algunas secciones, que se refleja a continuación.

Telemática. Telemática es la sección encargada de la instalación, configuración y control de todos los Clientes que han contratado con Telecom uno o más Canales Dedicados, los cuales constan de un Enlace y una o más Soluciones de Último Kilometro. Al realizar el respectivo análisis de la información requerida por cada Cliente se puede establecer el flujo de ésta a través de todos los elementos que intervienen en la conexión de un canal, flujo que se representa en la Figura 6.

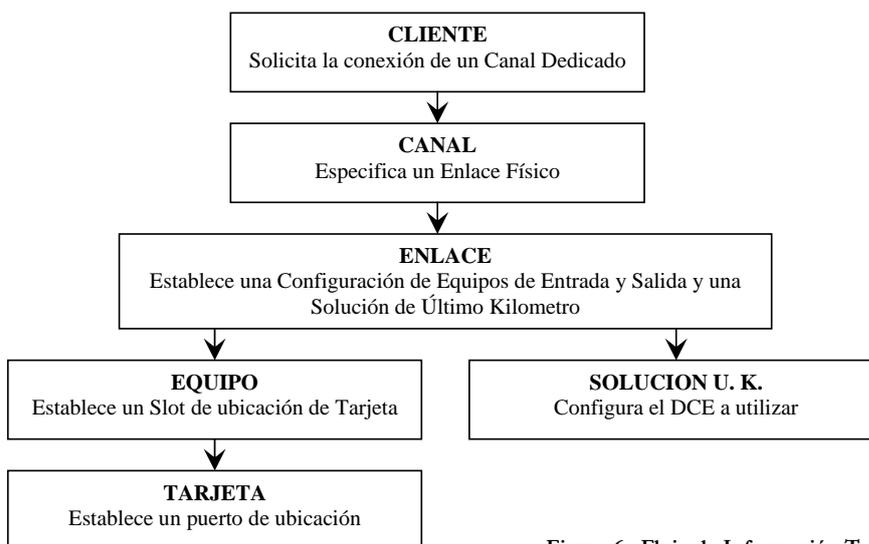


Figura 6. Flujo de Información Telemática

Después de haber determinado el flujo de la información entre los elementos determinados para el área, se especificaron los datos involucrados, que se representan en la Figura 7.

Ya en esta etapa, se identificó plenamente para cada elemento, la información recibida, y la información que ellos proporcionan, convirtiéndose en **Entidades**. La identificación de las entidades y sus atributos se detalla en la Tabla 3.

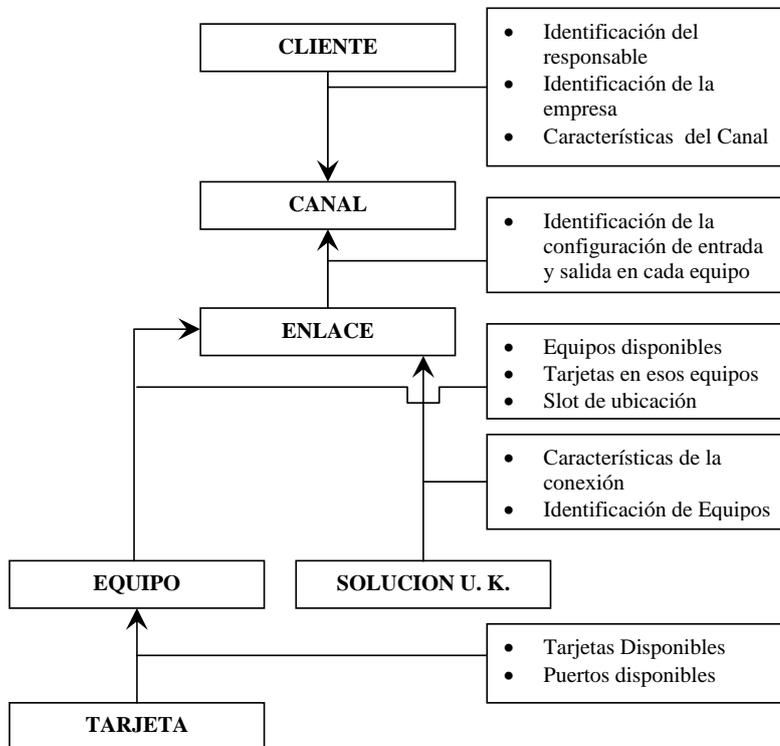


Figura 7. Datos Telemática

Tabla 3. Entidades y Atributos Telemática

CLIENTES
<ul style="list-style-type: none"> • Identificación del Responsable <ul style="list-style-type: none"> • Nombre y Apellidos • Identificación de la Empresa <ul style="list-style-type: none"> • Nit • Razon Social • Dirección • Telefono
CANALES
<ul style="list-style-type: none"> • Empresa • Características del Canal <ul style="list-style-type: none"> • Tipo de Canal • Velocidad • Origen • Destino
ENLACES
<ul style="list-style-type: none"> • Identificación del Canal • Identificación del enlace de Entrada y Salida <ul style="list-style-type: none"> • Equipo • Slot • Tarjeta • Puerto
EQUIPOS
<ul style="list-style-type: none"> • Identificación del Equipo • Tipo de Equipo • Numero de Slots
TARJETAS
<ul style="list-style-type: none"> • Identificación de la tarjeta • Tipo de Tarjeta • Puertos

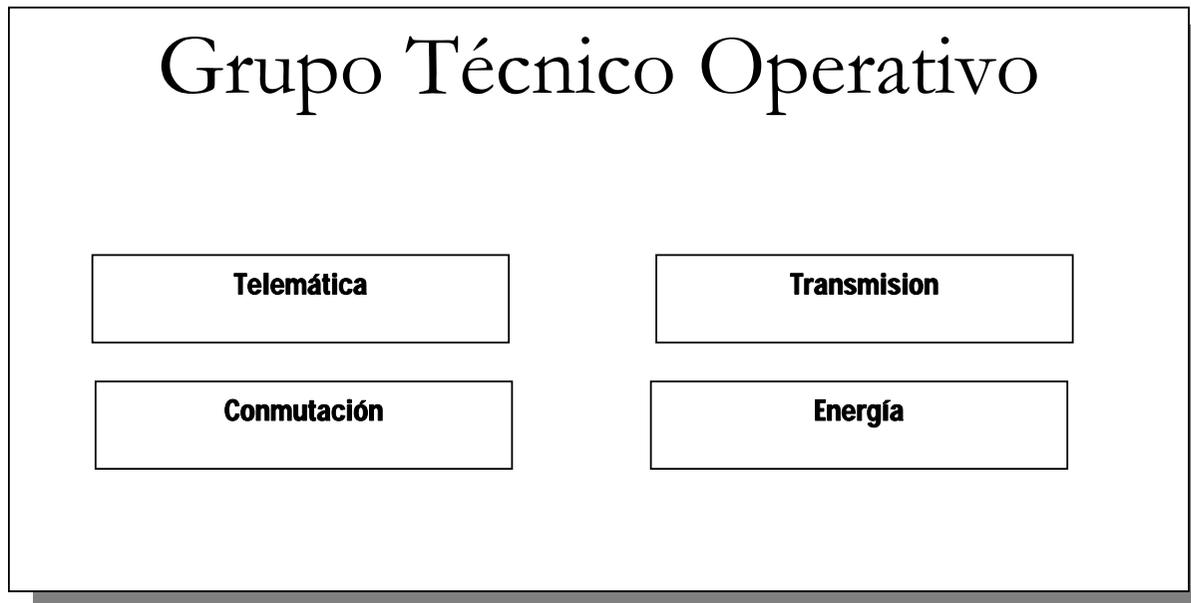
SOLUCIONES ULTIMO KILOMETRO
<ul style="list-style-type: none">• Enlace• Características del Equipo de Comunicación de Datos DCE<ul style="list-style-type: none">• Identificación• Tipo

Finalmente después de analizar el manejo de la información del área de **Telemática**, se dispuso a realizar un diseño de la Base de Datos en el cual se incluyen, aparte de las Entidades especificadas anteriormente, algunas Entidades lógicas que permiten organizar y facilitar el desempeño del área. Estas entidades serán especificadas con claridad en el diseño de la Base de Datos.

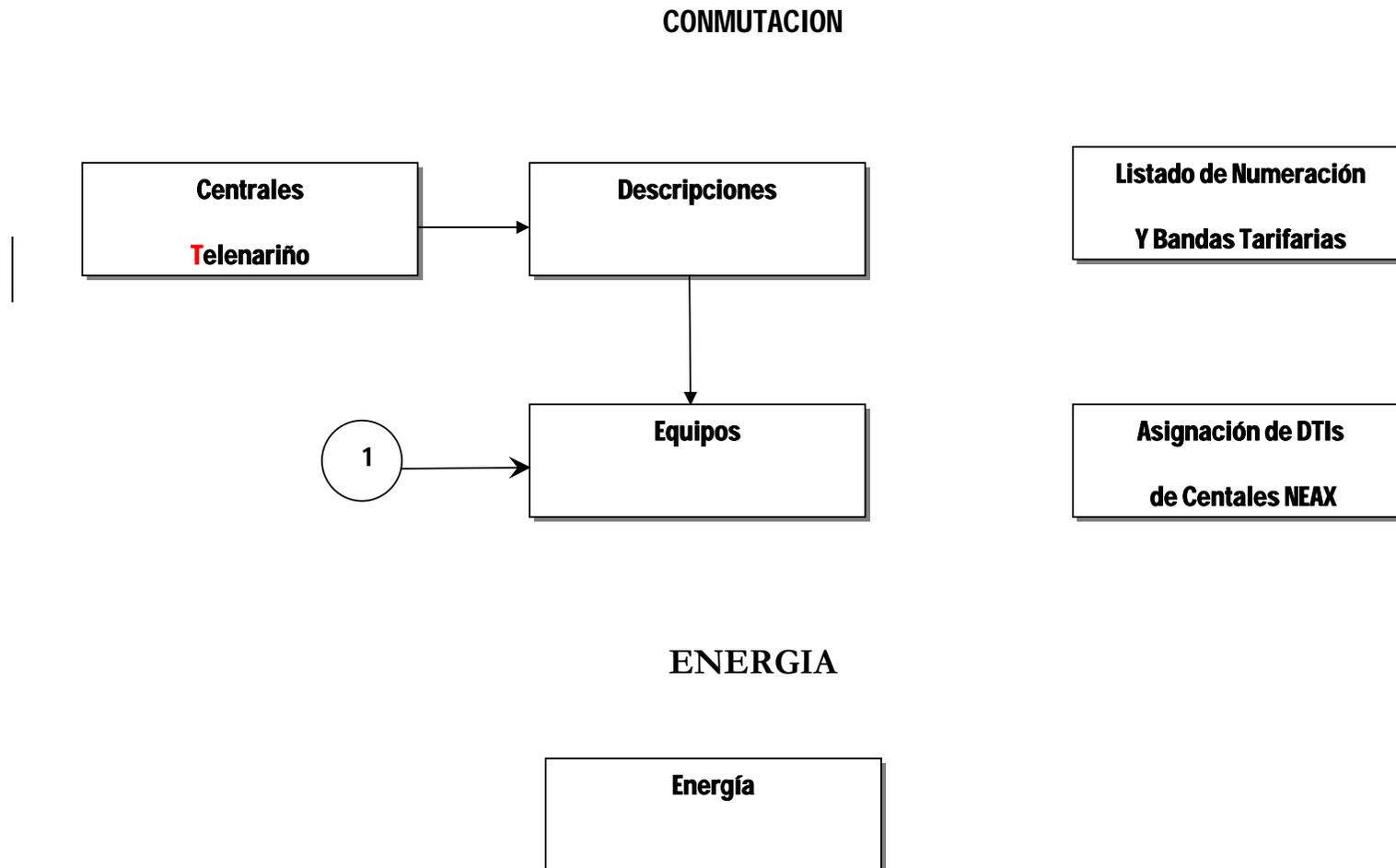
2.10 DISEÑO LOGICO DE LA BASE DE DATOS

Del anterior análisis sobre el área de **Telemática**, y algunas modificaciones a los diseños realizados por el Ingeniero Juan Carlos Lince, el diseño lógico final de la Base de Datos del área Técnico Operativa de la Empresa Nacional de Telecomunicaciones (**Telecom**), Gerencia del departamento de Nariño se presenta a continuación.

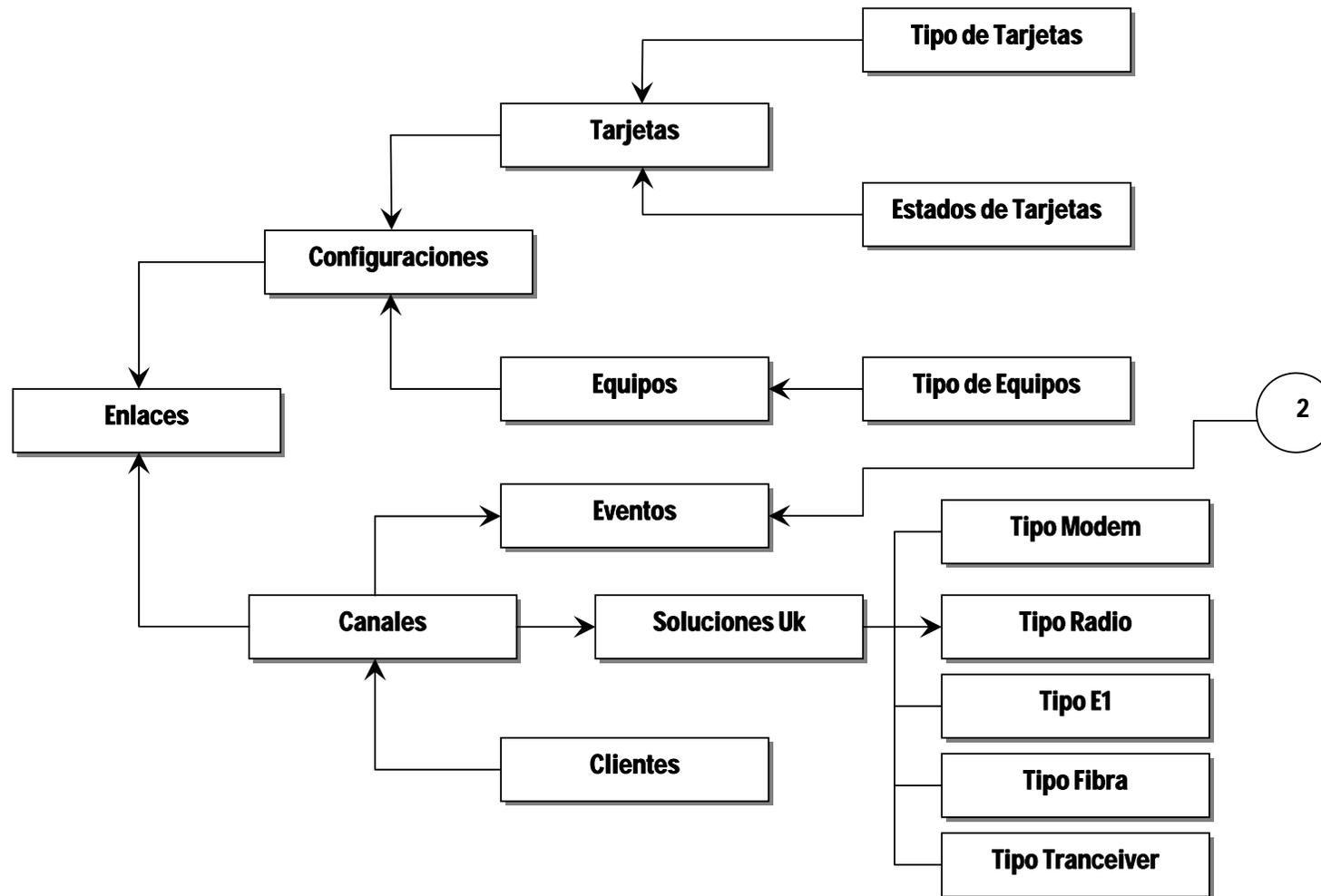
2.10.1 Diagrama de Contexto



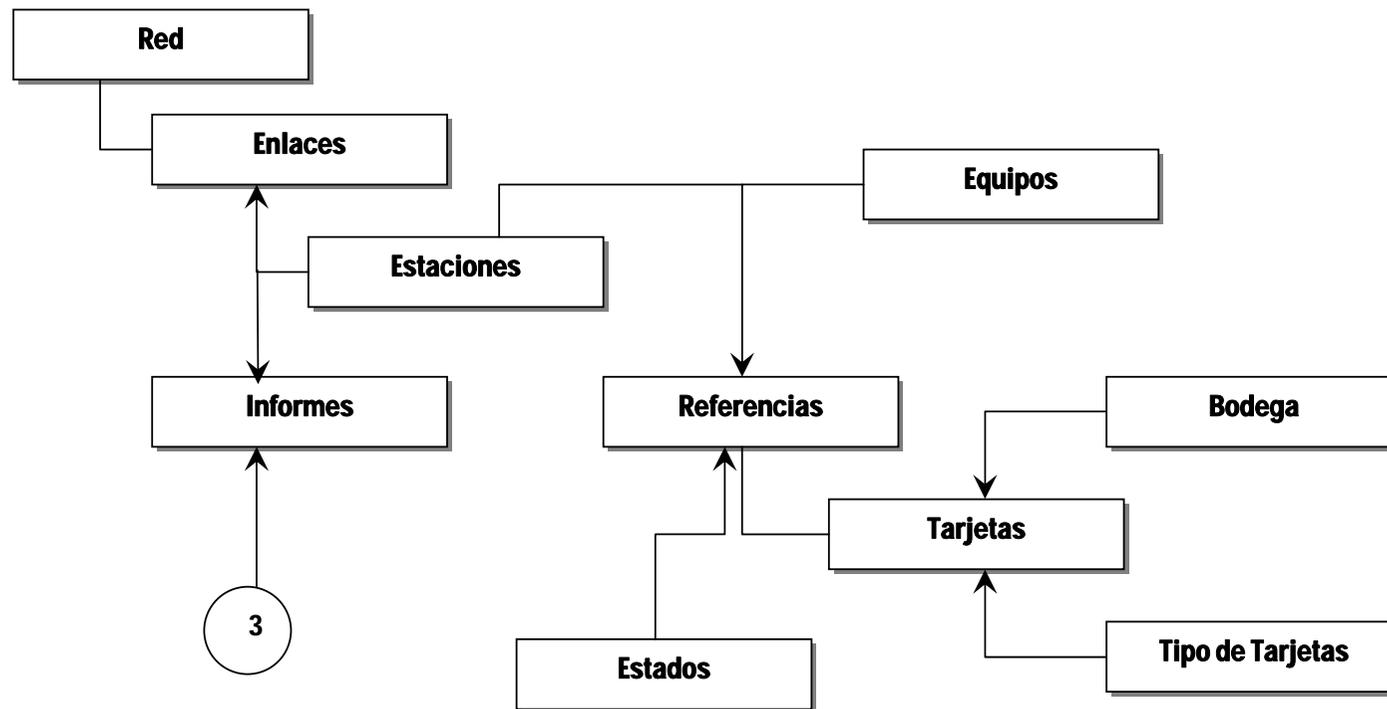
2.10.2 Diagramas Entidad Relación



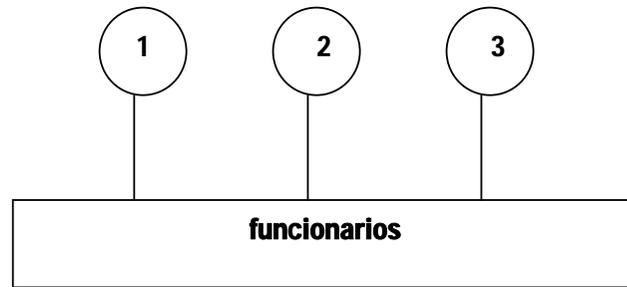
TELEMÁTICA



TRANSMISION



FUNCIONARIOS



2.10.3 Tablas Telematica

Cientes: clientes_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_cli	Numeric(5)	No	Pk
Razón Social	Razon_cli	Varchar(30)	No	
Nit Cliente	Nit_cli	Varchar(10)	No	
Nombre Responsable	Nom_cli	Varchar(30)	No	
Apellido Responsable	Ape_cli	Varchar((30)	No	
Dirección Cliente	Dir_cli	Varchar(30)	No	
Teléfono Cliente	Tel_cli	Varchar(20)	Si	

Tipo de Equipos: tipoEquipos_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_tip	Numeric(5)	No	Pk
Tipo de Equipo	Tipo_equ	Varchar(20)	No	

Equipos: equipos_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_equ	Numeric(5)	No	Pk
Código del Tipo	Cod_tip	Numeric(5)	No	Fk
Serial del Equipo	Serial	Varchar(20)	No	
Numero de slots	Num_slots	Numeric(5)	No	

Tipo de Tarjetas: tipo_tarjetas_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_tip	Numeric(5)	No	Pk
Tipo de Tarjeta	Tipo_tar	Varchar(20)	No	

Estados de las Tarjetas: estados_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_est	Numeric(5)	No	Pk
Estado	Estado	Varchar(20)	No	

Tarjetas: tarjetas_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_tar	Numeric(5)	No	Pk
Código del Tipo	Cod_tip	Numeric(5)	No	Fk
Código del Estado	Cod_est	Numeric(5)	No	Fk
Serial de la Tarjeta	Serial	Varchar(20)	No	
Tipo de Interfaz	Interfaz	Varchar(20)	No	
Numero de Puertos	Num_ptos	Numeric(5)	No	

Configuraciones de los Equipos-Tarjetas: configuraciones_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_conf	Numeric(5)	No	Pk
Código del Equipo	Cod_equ	Numeric(5)	No	Fk
Código de la Tarjeta	Cod_tar	Numeric(5)	No	Fk
Slot	slot	Numeric(5)	No	
Puerto	puerto	Numeric(5)	No	

Canales: canales_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_can	Numeric(5)	No	Pk
Código del Cliente	Cod_cli	Numeric(5)	No	Fk
Tipo de Canal	Tipo_can	Varchar(20)	No	
Velocidad del Canal	Velocidad	Varchar(20)	No	
Origen	Origen	Varchar(20)	No	
Destino	Destino	Varchar(20)	No	

Enlaces: enlaces_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_enl	Numeric(5)	No	Pk
Código del Canal	Cod_can	Numeric(5)	No	Fk
Configuración Entrada	Conf_ent	Numeric(5)	No	Fk
Configuración Salida	Conf_sal	Numeric(5)	No	Fk
Nombre del Enlace	Enlace	Varchar(20)	No	

Soluciones Ultimo Kilometro: solucion_uk				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_uk	Numeric(5)	No	Pk
Código del Canal	Cod_can	Numeric(5)	No	Fk
Serial del DCE en Telemática	Serial_te	Varchar(20)	No	
Serial del DCE en Cliente	Serial_cli	Varchar(20)	No	
Marca del DCE	Marca	Varchar(20)	No	
Tipo de DCE	Tipo_dce	Varchar(2)	No	

Tipo de DCE Modem: tipo_modem				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_mod	Numeric(5)	No	Pk
Código Ultimo Kilómetro	Cod_uk	Numeric(5)	No	Fk
Impedancia del Par	Impedancia	Varchar(10)	No	
Calibre del par	Calibre	Varchar(10)	No	
No. de los pares (Array)	Num_par	Int4[]	No	
No. Interconexiones (Array)	Num_int	Int4[]	No	

Tipo de DCE Radio: tipo_radio				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_rad	Numeric(5)	No	Pk
Código Ultimo Kilómetro	Cod_uk	Numeric(5)	No	Fk
Frecuencia de Transmisión	F_tx	Varchar(10)	No	
Frecuencia de Recepción	F_rx	Varchar(10)	No	
Banda	Banda	Varchar(20)	No	
Potencia de Transmisión	P_tx	Numeric(5)	No	
Antena	Antena	Varchar(20)	No	

Tipo de DCE Fibra Optica: tipo_fo				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_rad	Numeric(5)	No	Pk
Código Ultimo Kilómetro	Cod_uk	Numeric(5)	No	Fk
Tipo de Fibra	Tipo	Varchar(20)	No	

Tipo de DCE Tranceiver: tipo_tran				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_rad	Numeric(5)	No	Pk
Código Ultimo Kilómetro	Cod_uk	Numeric(5)	No	Fk
Interfaz de Entrada	Interfaz_ent	Varchar(20)	No	
Interfaz de Salida	Interfaz_sal	Varchar(20)	No	

Tipo de DCE E1: tipo_e1				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_rad	Numeric(5)	No	Pk
Código Ultimo Kilómetro	Cod_uk	Numeric(5)	No	Fk
Tipo de Cable	Cable	Varchar(20)	No	
E1	E1	Varchar(20)	No	

Eventos en Telematica: eventos_tel				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_even	Numeric(5)	No	Pk
Código Canal	Cod_can	Numeric(5)	No	Fk
Código del Funcionario	Cod_fun	Numeric(5)	No	Fk
Descripción del Evento	Evento	Varchar(40)	No	
Fecha	Fecha_even	Timestamp	No	
Detalle del Evento	Detalle	Text	No	

2.10.4 Tablas Transmisión

Estaciones: estaciones_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_est	Numeric(5)	No	Pk
Estación	Estacion	Varchar(30)	No	
Ubicación Longitud (Array)	Long_est	Varchar(3)[3]	No	
Ubicación Laltitud (Array)	Latit_est	Varchar(3)[3]	No	
Departamento	Depto_est	Varchar((30)	No	
Municipio	Municipio_est	Varchar(30)	No	
Marca de la Antena	Marca_antena	Varchar(20)	No	
Modelo de la Antena	Modelo_antena	Varchar(20)	No	
Tipo de Antena	Tipo_antena	Varchar(20)	No	
Frecuencia de Tx	Frecuencia	Varchar(10)	No	

Enlaces: enlaces_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_enl12	Numeric(5)	No	Pk
Código Estación A	Cod_enl1	Numeric(5)	No	Fk
Código Estación B	Cod_enl2	Numeric(5)	No	Fk

Red de Enlaces: red_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_red	Numeric(5)	No	Pk
Código Enlace	Cod_enl12	Numeric(5)	No	Fk
Multiacceso	Multiac	Varchar(30)	No	
Topología	Topologia	Varchar(2)	No	
Red	Red	Varchar(20)	No	
Tecnología	Tecnologia	Varchar(20)	No	
Marca del Equipo	Marca_equ	Varchar(10)	No	
Modelo del Equipo	Modelo_equ	Varchar(20)	No	
Tipo de Equipo	Tipo_equ	Varchar(10)	No	
Capacidad del canal	Capacidad	Varchar(5)	No	
Velocidad	Velocidad	Varchar(5)	No	
Potencia	Potencia	Varchar(5)	No	
Frecuencia Superior	Frecuencia	Varchar(5)	No	
Emisión	Emision	Varchar(10)	No	
Ancho de banda	Banda	Varchar(5)	No	
Distancia	Distancia	Varchar(10)	No	

Equipos: equipos_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_equ	Numeric(5)	No	Pk
Código del Estado	Cod_est	Numeric(5)	No	Fk
Serial	Referencia	Varchar(30)	No	
Numero de paneles	Paneles	Numeric(5)	No	

Tipo de Tarjetas: tipo_tarjetas_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_tip	Numeric(5)	No	Pk
Tipo de Tarjeta	Tipo_tar	Varchar(20)	No	

Estados de Equipos: estados_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_esta	Numeric(5)	No	Pk
Estado	Estado	Varchar(20)	No	

Tarjetas: tarjetas_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_tar	Numeric(5)	No	Pk
Código del Tipo	Cod_tip	Numeric(5)	No	Fk
Serial de la Tarjeta	Referencia	Varchar(30)	No	

Referencia de Tarjetas: referencia_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_ref	Numeric(5)	No	Pk
Código del Equipo	Cod_equ	Numeric(5)	No	Fk
Código de la Tarjeta	Cod_tar	Numeric(5)	No	Fk
Código del estado	Cod_esta	Numeric(5)	No	Fk

Tarjetas en Bodega: bodega_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_bod	Numeric(5)	No	Pk
Código de la Tarjeta	Cod_tar	Numeric(5)	No	Fk
Código de la Tarjeta	Cod_tar	Numeric(5)	No	Fk
Velocidad del Canal	Velocidad	Varchar(20)	No	
Origen	Origen	Varchar(20)	No	
Destino	Destino	Varchar(20)	No	

Informe de Comisión: informe_tra				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_inf	Numeric(5)	No	Pk
Código del Funcionario	Cod_fun	Numeric(5)	No	Fk
Código de la Estación	Cod_est	Numeric(5)	No	Fk
Fecha del informe	Fecha_inf	Timestamp	No	
Clase de Equipo	Clase_equ	Varchar(10)	No	
Pruebas realizadas	Pruebas	Text	No	
Detalles	Observacione	Text	No	
Modulos (Array)	Modulos	Varchar(3) [6]	No	
Items	Item	Varchar(30) [9]	No	
Especificaciones	Especifica	Varchar(30) [9]	No	
Mediciones	Medicion	Float[9]	No	
Valores	Valor	Float[9]	No	
No. De Bateria	Num_bat	Numeric [12]	No	
Densidad Batería	D_bat	Float[12]	No	
Voltaje de Batería	V_bat	Float[12]	No	
Sistema de Acumulación	Sistema	Varchar(30)	No	

2.10.5 Tablas Conmutación

Centrales: centrales_con				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_cen	Numeric(5)	No	Pk
Nombre Central	Nom_cen	Varchar(30)	No	
Versión de Software	Software_cen	Varchar(12)	No	
Ubicación	Lugar_cen	Varchar(30)	No	

Descripciones: descripciones_con				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_des	Numeric(5)	No	Pk
Código de la Central	Cod_cen	Numeric(5)	No	Fk
Descripción	Des_cen	Varchar(30)	No	
Precio total	Precio	Float	No	

Equipos: equipos_con				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_equ	Numeric(5)	No	Pk
Código de la Descripción	Cod_des	Numeric(5)	No	Fk
Código del Funcionario	Cod_fun	Numeric(5)	No	Fk
Descripción	Des_equ	Varchar(20)	No	
Cantidad	Can_equ	Numeic(5)	No	
Valor del Equipo	Precio	Float	No	
Subtotal	Subtotal	Float	No	

Listado de Numeración y Bandas Tarifarias: numeracion_bandas_con				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_nyb	Numeric(5)	No	Pk
Ciudad Destino	Ciu_des	Varchar(30)	No	
Departamento	Depto_nyb	Varchar(30)	No	
Indicativo	Ind_nyb	Varchar(10)	No	
Serie Local	Serie_l	Varchar(50)	No	
C1	C1_nyb	Varchar(3)	No	

Asignación de DTIs de Cnetrales NEAX: dti_neax_con				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_dti	Numeric(5)	No	Pk
CP	Cp_dti	Varchar(4)	No	
Spce	Space_dti	Varchar(4)	No	
Hw	Hw_dti	Varchar(4)	No	
Shw	Shw_dti	Varchar(4)	No	
Ltfa01	Ltfa01_dti	Varchar(10)	No	
Dti	Dti_dti	Varchar(4)	No	
Destino	Destino_dti	Varchar(30)	No	
Ruta	Ruta_dti	Varchar(40)	No	
Cable	Cable_dti	Varchar(30)	No	
Tb	Tb_dti	Varchar(5)	No	

2.10.6 Tablas Energia

Energía: energia_ene				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_ene	Numeric(5)	No	Pk
Locilidad	Local_ene	Varchar(30)	No	
Subestación	Subestacion_ene	Varchar(20)	No	
Motor Generador 1	Motog1_ene	Varchar(40)	No	
Motor Generador 2	Motog2_ene	Varchar(40)	No	
Rectificador	Recti_ene	Varchar(40)	No	
Aire A	Airea_ene	Varchar(30)	No	
Nombre Equipos	Equipos_ene	Varchar(30)	No	

2.10.7 Tabla Independiente de secciones

Funcionarios: funcionarios				
Descripción	Campo	Tipo	Nulo	Llave
Código Consecutivo	Cod_fun	Numeric(5)	No	Pk
Nombres	Nom_fun	Varchar(30)	No	
Apellidos	ape_fun	Varchar(30)	No	
Municipio de Residencia	Mun_fun	Varchar(30)	No	
Sección de Trabajo	Seccion_fun	Varchar(40)	No	
Area de Trabajo	Area_fun	Varchar(40)	No	
Cargo	Cargo_fun	Varchar(40)	No	
Telefono	Tel_fun	Varchar(12)	No	
Usuario de la Base de datos	Usuario	Varchar(16)	Si	
Grupo del Usuario	Grupo	Varchar(30)	Si	

2.11 IMPLEMENTACION DE LA BASE DE DATOS

El motor de Base de Datos Postgresql cuenta con su propios usuarios y grupos, los cuales pueden ser creados por el administrador del motor, que en este caso será el administrador general de la Base de Datos, por esta razón se decidió que los usuarios y grupos que la manejarán sean de este tipo y no parte de la Base de Datos como tal con el objetivo de incrementar su seguridad, debido a que en este caso, los grupos y usuarios serán tratados con el mismo nivel de seguridad que el motor otorga a su sistema y el mismo nivel que **Linux** otorga a sus componentes.

Para este fin se crearon tres tipos de usuarios para cada sección del **Grupo Técnico Operativo**, los cuales están incluidos dentro de tres tipos Grupos, dependiendo del grado de responsabilidad asignado al manejo de la información; estos tres tipos de usuarios son: **Administradores, Operativos e Invitados**. Teniendo en cuenta lo anterior, se definieron los grupos y sus usuarios descritos en la Tabla 4.

Tabla 4. Relación de Grupos y Usuarios

GRUPO	USUARIOS	PERMISOS
Administradores	<ul style="list-style-type: none"> • Administradores Telemática • Administradores Transmisión • Administradores Conmutación • Administradores Energía 	<ul style="list-style-type: none"> • Lectura para todas la tablas de la Base da Datos • Escritura para todas las tablas de su sección
Operativos	<ul style="list-style-type: none"> • Operativos Telemática • Operativos Transmisión • Operativos Conmutación • Operativos Energía 	<ul style="list-style-type: none"> • Lectura para todas la tablas de la Base de Datos • Escritura para algunas tablas de su sección
Invitados	<ul style="list-style-type: none"> • Invitados 	<ul style="list-style-type: none"> • Lectura para todas las tablas de la Base de datos

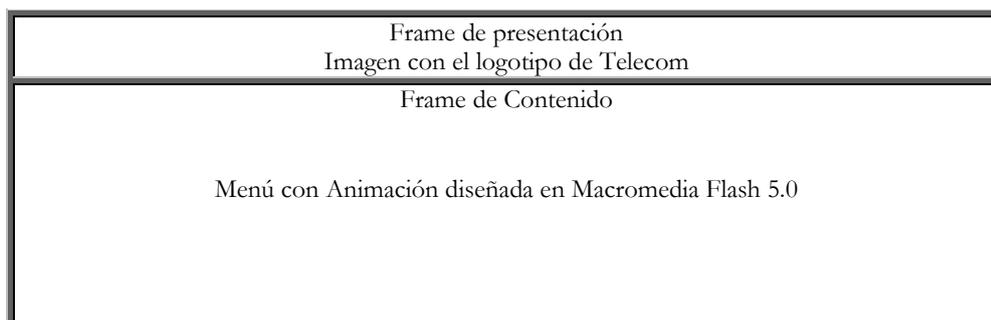
Finalmente se procedió a desarrollar los Scripts necesarios para la creación de las tablas, permisos y reglas determinados en el diseño de la Base de Datos, ver Anexo A. **“Scripts de Generación de Tablas”**.

2.12 DISEÑO DE LAS PAGINAS CON ACCESO A LA BASE DE DATOS

Con la Base de datos ya diseñada y funcionando, se procedió a realizar el diseño de la paginas Web que permitan acceder (insertar, borrar, actualizar, consultar) a la misma.

Se diseñaron prototipos para la pagina principal general y las paginas principales de cada sección, estos prototipos se presentan a continuación.

2.12.1 Pagina Principal General.

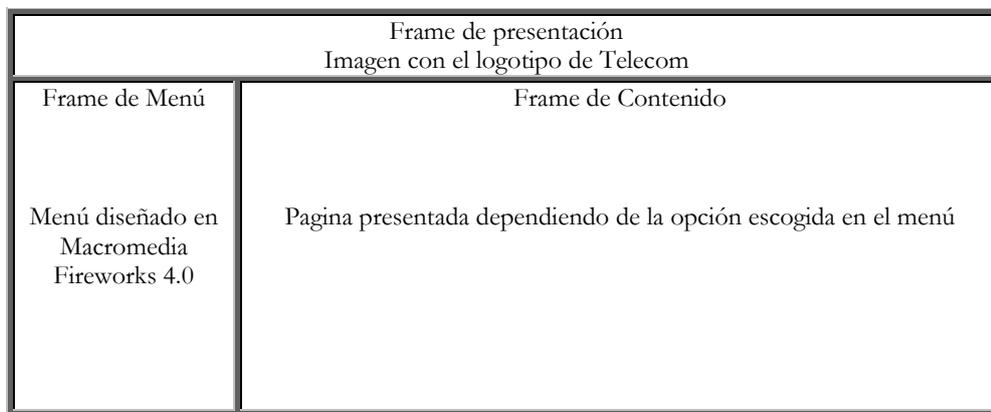


Estructura del Menú.

- Telemática
- Energía

- Transmisión
- Conmutación

2.12.2 Pagina Principal para cada sección.



Estructura del Menú Telemática

- **Cientes**
 - Modificar
 - Consultar
- **Canales**
 - Modificar
 - Consultar
- **Enlaces**
 - Adicionar
 - Modificar
 - Consultar
- **Equipos**
 - Tipos
 - Referencias
 - Consultar
- **Tarjetas**
 - Tipos
 - Estados

- Referencias
- Consultar
- **Soluciones UK**
 - Modificar
 - Consultar

- **Configuraciones**
 - Modificar
 - Consultar
- **Eventos**
- **Funcionarios**
 - Modificar
 - Consultar

Estructura del Menú Transmisión

- **Estaciones**
 - Modificar
 - Consultar
- **Enlaces**
 - Adicionar
 - Modificar
 - Consultar
- **Red**
 - Modificar
 - Consultar
- **Equipos**
 - Modificar
 - Consultar
- **Tarjetas**
 - Tipos
 - Estados
 - Tarjetas
 - Referencias
 - Consultar
- **Informes**
- **Funcionarios**
 - Modificar
 - Consultar

Estructura del Menú Conmutación

- **Centrales**
 - Modificar
 - Consultar
- **Descripciones**
 - Modificar
 - Consultar
- **Equipos**
 - Modificar
 - Consultar
- **Numeración y Bandas**
 - Modificar
 - Consultar
- **Asignación DTIs**
 - Modificar
 - Consultar
- **Funcionarios**
 - Modificar
 - Consultar

Estructura del Menú Energía

- **Energía**
 - Modificar
 - Consultar
- **Funcionarios**
 - Modificar
 - Consultar

Cabe anotar que los gráficos utilizados para todas las paginas fueron obtenidos de paginas pertenecientes a Internet e Intranet de Telecom, editadas con **GIMP**, un programa de edición de gráficos similar a Adobe Photoshop pero para Linux con Licencia GPL. Las paginas de presentación y menús fueron diseñadas en Software Macromedia,. Las paginas de contenido fueron realizadas con código **PHP** en un editor de texto.

A continuación se presenta las imágenes de las páginas Web Inicial, autenticación y la página principal de Telematica. En el Anexo D se especifica el código de generación para la pagina **Cientes de Telematica** de la sección **Telemática**.

IMAGEN PANTALLA INICIAL

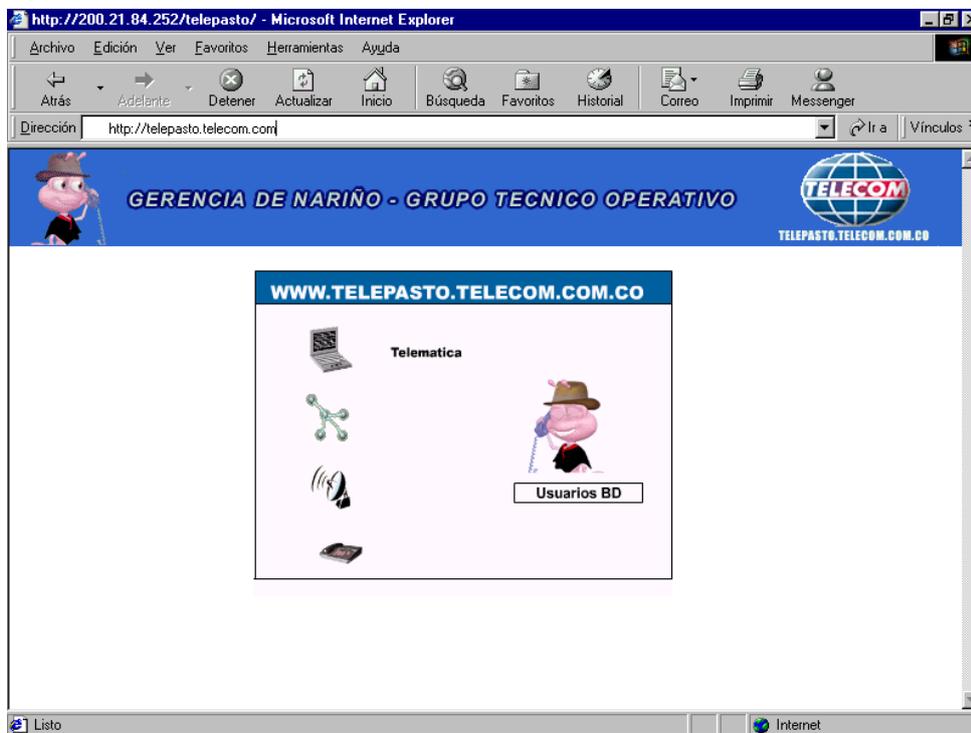


IMAGEN PANTALLA AUTENTIFICACION

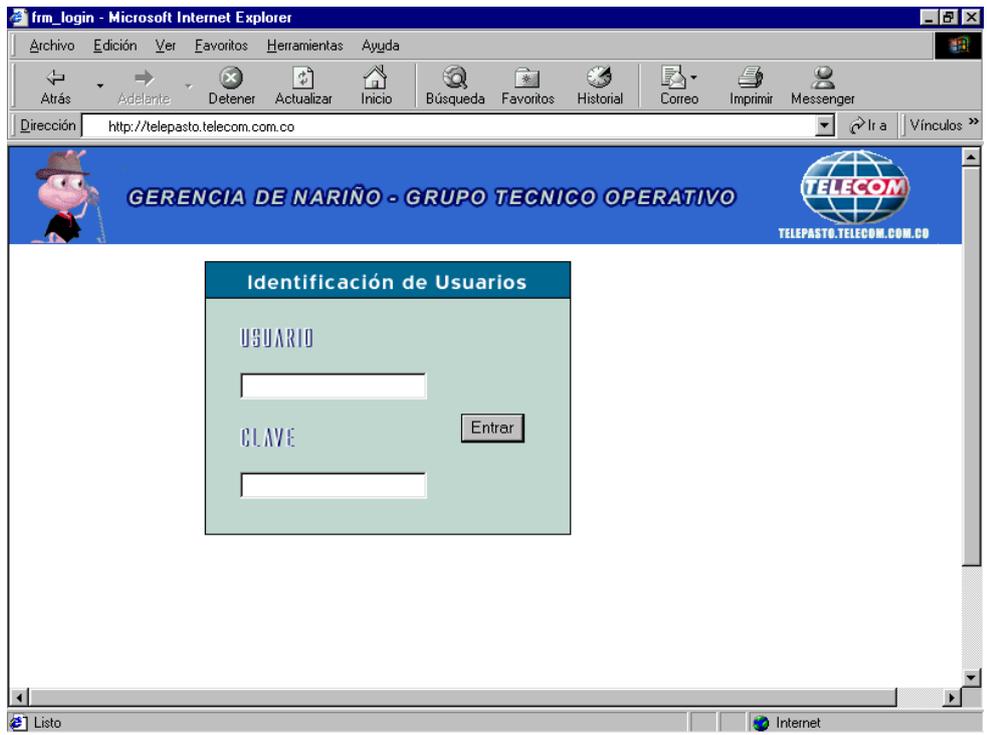


IMAGEN PANTALLA PRINCIPAL TELEMATICA

The screenshot shows a Microsoft Internet Explorer browser window displaying the main interface of the Telemática web application. The browser's address bar shows the URL `http://telepasto.telecom.com.co`. The application header features a blue banner with a cartoon character on the left, the text "GERENCIA DE NARIÑO - GRUPO TECNICO OPERATIVO" in the center, and the TELECOM logo on the right. Below the banner, a vertical navigation menu on the left lists various options: Clientes, Canales, Enlaces, Equipos, Tarjetas, Soluciones Wk, Configuraciones, Eventos, and Funcionarios. The main content area is titled "Clientes Telemática" and contains a form with several input fields labeled "Razon Social", "Nit", "Nombres", "Apellidos", "Direccion", and "Telefono". Below the form are buttons for "Adicionar", "Eliminar", "Actualizar", and "Buscar". A "VOLVER" link is also present. The status bar at the bottom of the browser shows "Listo" and "Internet".

3. CONCLUSIONES

SOBRE LA PASANTIA

- Una pasantía es una de las mejores formas de desarrollar el trabajo de grado, porque en ella además de aplicar y aumentar los conocimientos adquiridos en transcurso de la carrera, también se obtiene una gran experiencia de situaciones reales que pueden presentarse a futuro en la vida laboral.
- Una pasantía aporta a la empresa el fruto del esfuerzo del estudiante, los logros obtenidos por él y la compromete para continuar en el desarrollo de este tipo actividades.

SOBRE EL PROYECTO

- Los proyectos de investigación, desarrollan en el estudiante la capacidad de autoformación, responsabilidad y compromiso hacia las metas fijadas.
- Los proyectos de investigación otorgan a las empresas la oportunidad de crecer y acercarse a nuevas tecnologías que facilitan y mejoran la prestación de servicios.

SOBRE LAS HERRAMIENTAS UTILIZADAS

- Linux resulto ser un Sistema Operativo muy potente, que además de permitir desarrollar Software de aplicación, también disminuye en gran porcentaje los costos del mismo, convirtiéndolo en una opción muy conveniente.
- Postgresql es un motor de Base de Datos que no sólo se ajustó a las necesidades del proyecto, sino que también presentó características propias que facilitaron su integración con las demás aplicaciones software desarrolladas.
- PHP como herramienta de trabajo fue el uno de los elementos importantes que posibilitó la culminación del proyecto planteado. Aunque existen otros lenguajes de desarrollo, PHP presenta una característica que lo hicieron apropiado para este proyecto. Cabe destacar el hecho de que ya se encuentra integrado con muchas mas utilidades de desarrollo.

4. RECOMENDACIONES

SOBRE EL DESARROLLO

- Al hacer la instalación de todas las herramientas utilizadas es conveniente que se haga en primer lugar desde el CD de instalación de Linux Red Hat 7.1 (de manera automática), con el objetivo de revisar la configuración que ésta establece para luego volver a instalarlos manualmente y así comprender mejor su funcionamiento y poder personalizar su configuración.
- Se debe tener en cuenta a la hora de desarrollar un pagina de Internet, que es indispensable conocer los recursos existentes en el ambiente del usuario final, como son: Sistema Operativo, Equipos, Navegadores, ya que el resultado final no es siempre el mismo en todos los equipos.
- Estudiar profundamente, el Servidor de Correo **Qmail**, ya que éste cuenta con infinidad de características que pueden ser un gran tema de investigación para el desarrollo de otra pasantía, complementándolo con proveer éste servicio a usuarios de Internet Conmutado de Telecom.

- Estudiar el desempeño de otros Motores de Bases de Datos, en especial Oracle ya que como se sabe en uno de los motores de Bases de Datos más potentes que existen en la actualidad, además de que ya existen versiones para Linux con licencia GPL.

PARA LA EMPRESA

- Dotar a los pasantes con un equipo de trabajo con mejores características ya que éste puede ser en ciertos casos una limitante para mejorar los proyectos propuestos.
- Seguir con la incorporación de estudiantes de las diferentes Universidades a Telecom, como pasantes.
- Incentivar aún más por parte de Telecom al estudiante que ingresa a hacer su pasantía de tal forma que éste sea recompensado de la mejor forma posible.

PARA LA UNIVERSIDAD

- Formalizar entre las empresas y la Universidad de Nariño la incorporación de Estudiantes semestrales, y divulgar más esta posibilidad, para que esta no sea solo privilegio de unos cuantos.

Anexo A. Scripts Generación de Tablas

```

<?
pg_connect("host=localhost user=postgres dbname=telepaso");

//TABLAS TRANSMISION

//CREA LA TABLA informes
$result2 = pg_exec("DROP TABLE informe_tra");
$result4 = pg_exec("CREATE TABLE informe_tra (
    cod_inf      NUMERIC(5)      PRIMARY KEY,
    cod_fun      NUMERIC(5)      NOT NULL,
    cod_est      NUMERIC(5)      NOT NULL,
    fecha_inf    TIMESTAMP      NOT NULL,
    clase_equi   VARCHAR(10)     NOT NULL,
    pruebas      TEXT           NOT NULL,
    observacion  TEXT           NOT NULL,
    modulos      VARCHAR(3)[6]   NOT NULL,
    item         VARCHAR(30)[9]  NOT NULL,
    especifica   VARCHAR(30)[9]  NOT NULL,
    medicion     FLOAT[9]        NOT NULL,
    valor        FLOAT[9]        NOT NULL,
    num_bat      NUMERIC(2)[12]  NOT NULL,
    d_bat        FLOAT[12]       NOT NULL,
    v_bat        FLOAT[12]       NOT NULL,
    sistema      VARCHAR(30)     NOT NULL,
    FOREIGN KEY (cod_fun) references funcionarios,
    FOREIGN KEY (cod_est) references estaciones_tra");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA informe");
}
else
{
    echo ("TABLA informe CREADA CORRECTAMENTE");
}

```

```

//CREA LA TABLA estaciones
$result1 = pg_exec("DROP TABLE estaciones_tra");
$result1 = pg_exec("CREATE TABLE estaciones_tra (
    cod_est          NUMERIC(5)      PRIMARY KEY,
    estacion         VARCHAR(30)     NOT NULL,
    long_est         VARCHAR(3)[3]   NOT NULL,
    latit_est        VARCHAR(3)[3]   NOT NULL,
    depto_est        VARCHAR(30)     NOT NULL,
    municipio_est    VARCHAR(30)     NOT NULL,
    marca_antena     VARCHAR(20)     NOT NULL,
    modelo_antena    VARCHAR(20)     NOT NULL,
    tipo_antena      VARCHAR(20)     NOT NULL,
    frecuencia       VARCHAR(10)     NOT NULL);

if (!$result1)
{
    echo ("ERROR CREANDO LA TABLA estaciones");
}
else
{
    echo ("TABLA estaciones CREADA CORRECTAMENTE");
}

//CREA LA TABLA enlace
$result2 = pg_exec("DROP TABLE enlaces_tra");
$result4 = pg_exec("CREATE TABLE enlaces_tra (
    cod_enl12       NUMERIC(5)      PRIMARY KEY,
    cod_enl1        NUMERIC(5)      NOT NULL,
    cod_enl2        NUMERIC(5)      NOT NULL,
    FOREIGN KEY (cod_enl1) references estaciones_tra,
    FOREIGN KEY (cod_enl2) references estaciones_tra");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA enlaces transmision");
}
else
{
    echo ("TABLA enlaces transmision CREADA CORRECTAMENTE");
}

//CREA LA TABLA red transmision
$result2 = pg_exec("DROP TABLE red_tra");
$result4 = pg_exec("CREATE TABLE red_tra (
    cod_red         NUMERIC(5)      PRIMARY KEY,
    cod_enl12       NUMERIC(5)      NOT NULL,
    multiac         VARCHAR(30)     NOT NULL,
    topologia       VARCHAR(2)      NOT NULL,

```

```

red          VARCHAR(20)    NOT NULL,
tecnologia  VARCHAR(20)    NOT NULL,
marca_equ   VARCHAR(10)    NOT NULL,
modelo_equ  VARCHAR(20)    NOT NULL,
tipo_equ    VARCHAR(10)    NOT NULL,
capacidad   VARCHAR(5)     NOT NULL,
velocidad   VARCHAR(5)     NOT NULL,
potencia    VARCHAR(5)     NOT NULL,
frecuencia  VARCHAR(5)     NOT NULL,
emision     VARCHAR(10)    NOT NULL,
banda       VARCHAR(5)     NOT NULL,
distancia   VARCHAR(10)    NOT NULL,
FOREIGN KEY (cod_en12) references enlaces_tra");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA red transmision");
}
else
{
    echo ("TABLA red transmision CREADA CORRECTAMENTE");
}

//CREA LA TABLA equipos transmision
$result2 = pg_exec("DROP TABLE equipos_tra");
$result3 = pg_exec("CREATE TABLE equipos_tra (
    cod_equ    NUMERIC(5)    PRIMARY KEY,
    cod_est    NUMERIC(5)    NOT NULL,
    referencia VARCHAR(30)   NOT NULL,
    paneles    NUMERIC(5)    NOT NULL,
    FOREIGN KEY (cod_est) references estaciones_tra");

if (!$result3)
{
    echo ("ERROR CREANDO LA TABLA equipos transmision");
}
else
{
    echo ("TABLA equipos transmision CREADA CORRECTAMENTE");
}

//CREA LA TABLA tipos de tarjetas transmision
$result2 = pg_exec("DROP TABLE tipos_tarjetas_tra");
$result3 = pg_exec("CREATE TABLE tipos_tarjetas_tra (
    cod_tip NUMERIC(5)    PRIMARY KEY,
    tipo_tar VARCHAR(20) NOT NULL");

if (!$result3)

```

```
{
  echo ("ERROR CREANDO LA TABLA tipos tarjetas transmision");
}
else
{
  echo ("TABLA tipos tarjetas transmision CREADA CORRECTAMENTE");
}

//CREA LA TABLA tarjetas transmision
$result2 = pg_exec("DROP TABLE tarjetas_tra");
$result3 = pg_exec("CREATE TABLE tarjetas_tra (
  cod_tar      NUMERIC(5)      PRIMARY KEY,
  cod_tip      NUMERIC(5)      NOT NULL,
  referencia   VARCHAR(30)     NOT NULL,
  FOREIGN KEY (cod_tip) references tipos_tarjetas_tra");

if (!$result3)
{
  echo ("ERROR CREANDO LA TABLA tarjetas transmision");
}
else
{
  echo ("TABLA tarjetas transmision CREADA CORRECTAMENTE");
}

//CREA LA TABLA REFERENCIA TARJETAS
$result2 = pg_exec("DROP TABLE referencia_tra");
$result3 = pg_exec("CREATE TABLE referencia_tra (
  cod_ref      NUMERIC(5)      PRIMARY KEY,
  cod_tar      NUMERIC(5)      NOT NULL,
  cod_esta     NUMERIC(5)      NOT NULL,
  cod_equ      NUMERIC(5)      NOT NULL,
  FOREIGN KEY (cod_tar) references tarjetas_tra,
  FOREIGN KEY (cod_esta) references estados_tra,
  FOREIGN KEY (cod_equ) references equipos_tra");

if (!$result3)
{
  echo ("ERROR CREANDO LA TABLA referencia tarjetas");
}
else
{
  echo ("TABLA referencia tarjetas CREADA CORRECTAMENTE");
}

//CREA LA TABLA estados transmision
$result2 = pg_exec("DROP TABLE estados_tra");
```

```

$result4 = pg_exec("CREATE TABLE estados_tra (
    cod_esta    NUMERIC(5)    PRIMARY KEY,
    estado     VARCHAR(30)   NOT NULL);
if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA estados transmision");
}
else
{
    echo ("TABLA estados transmision CREADA CORRECTAMENTE");
}

//CREA LA TABLA bodega
$result2 = pg_exec("DROP TABLE bodega_tra");
$result4 = pg_exec("CREATE TABLE bodega_tra (
    codtar_bod    NUMERIC(5)    PRIMARY KEY,
    cod_tar       NUMERIC(5)    NOT NULL,
    cod_esta      NUMERIC(5)    NOT NULL,
    FOREIGN KEY (cod_tar) references tarjetas_tra,
    FOREIGN KEY (cod_esta) references estados_tra");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA bodega");
}
else
{
    echo ("TABLA bodega CREADA CORRECTAMENTE");
}

//CREACION DE SECUENCIAS PARA TRANSMISION
$result2 = pg_exec("DROP SEQUENCE sec_codest_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codest_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codenl_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codenl_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codred_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codred_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codequ_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codequ_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codtar_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codtar_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codtiptar_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codtiptar_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codref_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codref_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codesta_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codesta_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codbod_tra");

```

```

$result3 = pg_exec("CREATE SEQUENCE sec_codbod_tra");
$result2 = pg_exec("DROP SEQUENCE sec_codinf_tra");
$result3 = pg_exec("CREATE SEQUENCE sec_codinf_tra");

//TABLAS TELEMATICA

//CREA LA TABLA clientes telematica
$result1 = pg_exec("DROP TABLE clientes_tel");
$result1 = pg_exec("CREATE TABLE clientes_tel (
    cod_cli      NUMERIC(5)      PRIMARY KEY,
    razon_cli   VARCHAR(30)     NOT NULL,
    nit_cli     VARCHAR(10)     NOT NULL,
    nom_cli     VARCHAR(30)     NOT NULL,
    ape_cli     VARCHAR(30)     NOT NULL,
    dir_cli     VARCHAR(30)     NOT NULL,
    tel_cli     VARCHAR(20));

if (!$result1)
{
    echo ("ERROR CREANDO LA TABLA clientes de telematica");
}
else
{
    echo ("TABLA clientes de telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA tipos equipos telematica
$result2 = pg_exec("DROP TABLE tipos_equipos_tel");
$result4 = pg_exec("CREATE TABLE tipos_equipos_tel (
    cod_tip     NUMERIC(5)      PRIMARY KEY,
    tipo_equ   VARCHAR(20)     NOT NULL);

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA tipos equipos telematica");
}
else
{
    echo ("TABLA tipos equipos telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA equipos telematica
$result2 = pg_exec("DROP TABLE equipos_tel");
$result4 = pg_exec("CREATE TABLE equipos_tel (
    cod_equi   NUMERIC(5)      PRIMARY KEY,
    cod_tip    NUMERIC(5)      NOT NULL,
    serial     VARCHAR(20)     NOT NULL,
    num_slots  NUMERIC(5)      NOT NULL,

```

```
FOREIGN KEY (cod_tip) references tipos_equipos_tel");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA equipos telematica");
}
else
{
    echo ("TABLA equipos telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA tipos tarjetas telematica
$result2 = pg_exec("DROP TABLE tipos_tarjetas_tel");
$result4 = pg_exec("CREATE TABLE tipos_tarjetas_tel (
    cod_tip    NUMERIC(5)    PRIMARY KEY,
    tipo_tar   VARCHAR(20)   NOT NULL);");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA tipos tarjetas telematica");
}
else
{
    echo ("TABLA tipos tarjetas telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA ESTADOS TARJETAS TELEMATICA
$result2 = pg_exec("DROP TABLE estados_tel");
$result4 = pg_exec("CREATE TABLE estados_tel (
    cod_est    NUMERIC(5)    PRIMARY KEY,
    estado     VARCHAR(20)   NOT NULL);");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA estados telematica");
}
else
{
    echo ("TABLA estados telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA tarjetas telematica
$result2 = pg_exec("DROP TABLE tarjetas_tel");
$result4 = pg_exec("CREATE TABLE tarjetas_tel (
    cod_tar    NUMERIC(5)    PRIMARY KEY,
    cod_tip    NUMERIC(5)    NOT NULL,
    cod_est    NUMERIC(5)    NOT NULL,
    serial     VARCHAR(20)   NOT NULL,
```

```

        interfaz      VARCHAR(20)      NOT NULL,
        num_ptos     NUMERIC(5)        NOT NULL,
        FOREIGN KEY (cod_tip) references tipos_tarjetas_tel,
        FOREIGN KEY (cod_est) references estados_tel");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA tarjetas telematica");
}
else
{
    echo ("TABLA tarjetas telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA configuraciones telematica
$result2 = pg_exec("DROP TABLE configuraciones_tel");
$result3 = pg_exec("CREATE TABLE configuraciones_tel (
    cod_conf      NUMERIC(5)          PRIMARY KEY,
    cod_equi      NUMERIC(5)          NOT NULL,
    cod_tar       NUMERIC(5)          NOT NULL,
    slot          NUMERIC(5)          NOT NULL,
    puerto        NUMERIC(5)          NOT NULL,
    FOREIGN KEY (cod_tar) references tarjetas_tel,
    FOREIGN KEY (cod_equi) references equipos_tel");

if (!$result3)
{
    echo ("ERROR CREANDO LA TABLA configuraciones telematica");
}
else
{
    echo ("TABLA configuraciones telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA canales telematica
$result2 = pg_exec("DROP TABLE canales_tel");
$result3 = pg_exec("CREATE TABLE canales_tel (
    cod_can      NUMERIC(5)          PRIMARY KEY,
    cod_cli      NUMERIC(5)          NOT NULL,
    tipo_can     VARCHAR(20)         NOT NULL,
    velocidad    VARCHAR(20)         NOT NULL,
    origen       VARCHAR(20)         NOT NULL,
    destino      VARCHAR(20)         NOT NULL,
    FOREIGN KEY (cod_cli) references clientes_tel");

if (!$result3)
{
    echo ("ERROR CREANDO LA TABLA canales telematica");
}

```

```
}
else
{
    echo ("TABLA canales telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA enlaces telematica
$result2 = pg_exec("DROP TABLE enlaces_tel");
$result3 = pg_exec("CREATE TABLE enlaces_tel (
    cod_enl      NUMERIC(5)      PRIMARY KEY,
    enlace       VARCHAR(20)     NOT NULL,
    cod_can      NUMERIC(5)     NOT NULL,
    conf_ent     NUMERIC(5)     NOT NULL,
    conf_sal     NUMERIC(5)     NOT NULL,
    FOREIGN KEY (cod_can) references canales_tel,
    FOREIGN KEY (conf_ent) references configuraciones_tel,
    FOREIGN KEY (conf_sal) references configuraciones_tel");

if (!$result3)
{
    echo ("ERROR CREANDO LA TABLA enlaces telematica");
}
else
{
    echo ("TABLA enlaces telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA eventos telematica
$result2 = pg_exec("DROP TABLE eventos_tel");
$result4 = pg_exec("CREATE TABLE eventos_tel (
    cod_even     NUMERIC(5)     PRIMARY KEY,
    cod_can      NUMERIC(5)     NOT NULL,
    cod_fun      NUMERIC(5)     NOT NULL,
    evento       VARCHAR(40)    NOT NULL,
    fecha_even   TIMESTAMP      NOT NULL,
    observacion  TEXT,
    FOREIGN KEY (cod_can) references canales_tel,
    FOREIGN KEY (cod_fun) references funcionarios");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA eventos telematica");
}
else
{
    echo ("TABLA eventos telematica CREADA CORRECTAMENTE");
}
}
```

```

//CREA LA TABLA funcionarios
$result2 = pg_exec("DROP TABLE funcionarios");
$result4 = pg_exec("CREATE TABLE funcionarios (
    cod_fun      NUMERIC(9)      PRIMARY KEY,
    nom_fun      VARCHAR(30)     NOT NULL,
    ape_fun      VARCHAR(30)     NOT NULL,
    mun_fun      VARCHAR(30)     NOT NULL,
    seccion_fun  VARCHAR(40)     NOT NULL,
    area_fun     VARCHAR(40)     NOT NULL,
    cargo_fun    VARCHAR(40)     NOT NULL,
    tel_fun      VARCHAR(12)     NOT NULL,
    grupo        VARCHAR(30),
    usuario      VARCHAR(16)");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA funcionarios");
}
else
{
    echo ("TABLA funcionarios CREADA CORRECTAMENTE");
}

//CREACION DE SECUENCIAS
$result2 = pg_exec("DROP SEQUENCE sec_codcli");
$result3 = pg_exec("CREATE SEQUENCE sec_codcli");
$result2 = pg_exec("DROP SEQUENCE sec_codequi");
$result3 = pg_exec("CREATE SEQUENCE sec_codequi");
$result2 = pg_exec("DROP SEQUENCE sec_codtar");
$result3 = pg_exec("CREATE SEQUENCE sec_codtar");
$result2 = pg_exec("DROP SEQUENCE sec_codconf");
$result3 = pg_exec("CREATE SEQUENCE sec_codconf");
$result2 = pg_exec("DROP SEQUENCE sec_codcan");
$result3 = pg_exec("CREATE SEQUENCE sec_codcan");
$result2 = pg_exec("DROP SEQUENCE sec_codenl");
$result3 = pg_exec("CREATE SEQUENCE sec_codenl");
$result2 = pg_exec("DROP SEQUENCE sec_codeven");
$result3 = pg_exec("CREATE SEQUENCE sec_codeven");
$result2 = pg_exec("DROP SEQUENCE sec_codfun");
$result3 = pg_exec("CREATE SEQUENCE sec_codfun");
$result2 = pg_exec("DROP SEQUENCE sec_codest");
$result3 = pg_exec("CREATE SEQUENCE sec_codest");
$result2 = pg_exec("DROP SEQUENCE sec_codtiptar");
$result3 = pg_exec("CREATE SEQUENCE sec_codtiptar");
$result2 = pg_exec("DROP SEQUENCE sec_codtipequ");
$result3 = pg_exec("CREATE SEQUENCE sec_codtipequ");

```

```

//CREA LA TABLA solucion_uk telematica
$result1 = pg_exec("DROP TABLE solucion_uk");
$result1 = pg_exec("CREATE TABLE solucion_uk (
    cod_uk      NUMERIC(5)      PRIMARY KEY,
    cod_can     NUMERIC(5)      NOT NULL,
    serial_tel  VARCHAR(20)     NOT NULL,
    serial_cli  VARCHAR(20)     NOT NULL,
    marca       VARCHAR(20)     NOT NULL,
    tipo_dce    VARCHAR(2)      NOT NULL,
    FOREIGN KEY (cod_can) references canales_tel");

if (!$result1)
{
    echo ("ERROR CREANDO LA TABLA solucion uk de telematica");
}
else
{
    echo ("TABLA solucion uk de telematica CREADA CORRECTAMENTE");
}

//CREA LA TABLA tipo_modem
$result2 = pg_exec("DROP TABLE tipo_modem");
$result4 = pg_exec("CREATE TABLE tipo_modem (
    cod_mod     NUMERIC(5)      PRIMARY KEY,
    cod_uk      NUMERIC(5)      NOT NULL,
    impedancia  VARCHAR(10)     NOT NULL,
    calibre     VARCHAR(10)     NOT NULL,
    propietario VARCHAR(30)     NOT NULL,
    num_par     INT4[]          NOT NULL,
    num_int     INT4[]          NOT NULL,
    FOREIGN KEY (cod_uk) references solucion_uk");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA tipo modem");
}
else
{
    echo ("TABLA tipo modem CREADA CORRECTAMENTE");
}

//CREA LA TABLA tipo_radio
$result2 = pg_exec("DROP TABLE tipo_radio");
$result4 = pg_exec("CREATE TABLE tipo_radio (
    cod_rad     NUMERIC(5)      PRIMARY KEY,
    cod_uk      NUMERIC(5)      NOT NULL,
    f_tx        VARCHAR(10)     NOT NULL,
    f_rx        VARCHAR(10)     NOT NULL,

```

```
        banda        VARCHAR(20)        NOT NULL,
        p_tx         NUMERIC(5)         NOT NULL,
        antena       VARCHAR(20)        NOT NULL,
        FOREIGN KEY (cod_uk) references solucion_uk");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA tipo radio");
}
else
{
    echo ("TABLA tipo radio CREADA CORRECTAMENTE");
}

//CREA LA TABLA tipo_fo
$result2 = pg_exec("DROP TABLE tipo_fo");
$result4 = pg_exec("CREATE TABLE tipo_fo (
    cod_fo        NUMERIC(5)        PRIMARY KEY,
    cod_uk        NUMERIC(5)        NOT NULL,
    tipo          VARCHAR(20)        NOT NULL,
    FOREIGN KEY (cod_uk) references solucion_uk");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA tipo fibra");
}
else
{
    echo ("TABLA tipo fibra CREADA CORRECTAMENTE");
}

//CREA LA TABLA tipo_e1
$result2 = pg_exec("DROP TABLE tipo_e1");
$result4 = pg_exec("CREATE TABLE tipo_e1 (
    cod_e1        NUMERIC(5)        PRIMARY KEY,
    cod_uk        NUMERIC(5)        NOT NULL,
    cable         VARCHAR(20)        NOT NULL,
    e1            VARCHAR(20)        NOT NULL,
    FOREIGN KEY (cod_uk) references solucion_uk");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA e1");
}
else
{
    echo ("TABLA e1 CREADA CORRECTAMENTE");
}
}
```

```

//CREA LA TABLA tranceiver
$result2 = pg_exec("DROP TABLE tipo_tran");
$result4 = pg_exec("CREATE TABLE tipo_tran (
    cod_tran    NUMERIC(5)    PRIMARY KEY,
    cod_uk     NUMERIC(5)    NOT NULL,
    interfaz_ent  VARCHAR(20)  NOT NULL,
    interfaz_sal  VARCHAR(20)  NOT NULL,
    FOREIGN KEY (cod_uk) references solucion_uk");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA tipo tranceiver");
}
else
{
    echo ("TABLA tipo tranceiver CREADA CORRECTAMENTE");
}

//CREACION DE SECUENCIAS
$result2 = pg_exec("DROP SEQUENCE sec_coduk");
$result3 = pg_exec("CREATE SEQUENCE sec_coduk");
$result2 = pg_exec("DROP SEQUENCE sec_codmod");
$result3 = pg_exec("CREATE SEQUENCE sec_codmod");
$result2 = pg_exec("DROP SEQUENCE sec_codrad");
$result3 = pg_exec("CREATE SEQUENCE sec_codrad");
$result2 = pg_exec("DROP SEQUENCE sec_code1");
$result3 = pg_exec("CREATE SEQUENCE sec_code1");
$result2 = pg_exec("DROP SEQUENCE sec_codtran");
$result3 = pg_exec("CREATE SEQUENCE sec_codtran");
$result2 = pg_exec("DROP SEQUENCE sec_codfo");
$result3 = pg_exec("CREATE SEQUENCE sec_codfo");

//CREA LA TABLA Energia
$result1 = pg_exec("DROP TABLE energia_ene");
$result1 = pg_exec("CREATE TABLE energia_ene (
    cod_ene        NUMERIC(5)    PRIMARY KEY,
    local_ene      VARCHAR(30)   NOT NULL,
    subestacion_ene  VARCHAR(20)  NOT NULL,
    motog1_ene     VARCHAR(40)   NOT NULL,
    motog2_ene     VARCHAR(40)   NOT NULL,
    recti_ene      VARCHAR(40)   NOT NULL,
    airea_ene      VARCHAR(30)   NOT NULL,
    equipos_ene    VARCHAR(30)   NOT NULL");

if (!$result1)
{
    echo ("ERROR CREANDO LA TABLA energia");
}

```

```
}
else
{
    echo ("TABLA energia CREADA CORRECTAMENTE");
}

//CREACION DE SECUENCIAS
$result2 = pg_exec("DROP SEQUENCE sec_cod_ene");
$result2 = pg_exec("CREATE SEQUENCE sec_cod_ene");

//CREA LA TABLA centrales
$result1 = pg_exec("DROP TABLE centrales_con");
$result1 = pg_exec("CREATE TABLE centrales_con (
    cod_cen    NUMERIC(5)    PRIMARY KEY,
    nom_cen    VARCHAR(30)   NOT NULL,
    software_cen VARCHAR(12) NOT NULL,
    lugar_cen  VARCHAR(30)   NOT NULL);");

if (!$result1)
{
    echo ("ERROR CREANDO LA TABLA centrales");
}
else
{
    echo ("TABLA centrales CREADA CORRECTAMENTE");
}

//CREA LA TABLA descripciones
$result2 = pg_exec("DROP TABLE descripciones_con");
$result4 = pg_exec("CREATE TABLE descripciones_con (
    cod_des    INTEGER      PRIMARY KEY,
    cod_cen    NUMERIC(5)    NOT NULL,
    des_cen    VARCHAR(40)   NOT NULL,
    precio_cen FLOAT        NOT NULL,
    FOREIGN KEY (cod_cen) references centrales_con);");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA descripciones");
}
else
{
    echo ("TABLA descripciones CREADA CORRECTAMENTE");
}

//CREA LA TABLA equipos
$result2 = pg_exec("DROP TABLE equipos_con");
$result4 = pg_exec("CREATE TABLE equipos_con (
```

```

cod_equi    NUMERIC(5)    PRIMARY KEY,
cod_des    NUMERIC(5)    NOT NULL,
cod_fun    NUMERIC(5)    NOT NULL,
des_equi    VARCHAR(20)  NOT NULL,
can_equi    NUMERIC(5)    NOT NULL,
precio     FLOAT         NOT NULL,
subtotal   FLOAT         NOT NULL,
FOREIGN KEY (cod_des) references descripciones_con,
FOREIGN KEY (cod_fun) references funcionarios");

if (!$result4)
{
    echo ("ERROR CREANDO LA TABLA equipos");
}
else
{
    echo ("TABLA equipos CREADA CORRECTAMENTE");
}

//CREA LA TABLA listado numeracion y bandas tarifarias
$result2 = pg_exec("DROP TABLE numeracion_bandas_con");
$result3 = pg_exec("CREATE TABLE numeracion_bandas_con (
    cod_nyb    NUMERIC(5)    PRIMARY KEY,
    ciu_des    VARCHAR(30)  NOT NULL,
    depto_nyb  VARCHAR(10)  NOT NULL,
    ind_nyb    VARCHAR(10)  NOT NULL,
    serie_1    VARCHAR(50)  NOT NULL,
    c1_nyb     VARCHAR(3)   NOT NULL)");

if (!$result3)
{
    echo ("ERROR CREANDO LA TABLA numeracion y bandas tarifarias");
}
else
{
    echo ("TABLA numeracion y bandas tarifarias CREADA CORRECTAMENTE");
}

//CREA LA TABLA Asignacion de DTI de centrales NEAX
$result2 = pg_exec("DROP TABLE dti_neax_con");
$result3 = pg_exec("CREATE TABLE dti_neax_con (
    cod_dti    NUMERIC(5)    PRIMARY KEY,
    cp_dti     VARCHAR(4)    NOT NULL,
    space_dti  VARCHAR(4)    NOT NULL,
    hw_dti     VARCHAR(4)    NOT NULL,
    shw_dti    VARCHAR(4)    NOT NULL,
    ltfa01_dti VARCHAR(10)   NOT NULL,
    dti_dti    VARCHAR(4)    NOT NULL,

```

```
destino_dti    VARCHAR(30)    NOT NULL,
ruta_dti      VARCHAR(40)    NOT NULL,
cable_dti     VARCHAR(30)    NOT NULL,
tb_dti        VARCHAR(5)     NOT NULL));

if (!$result3)
{
    echo ("ERROR CREANDO LA TABLA dti neax");
}
else
{
    echo ("TABLA dti neax CREADA CORRECTAMENTE");
}

//CREACION DE SECUENCIAS
$result2 = pg_exec("DROP SEQUENCE sec_codcen_con");
$result3 = pg_exec("CREATE SEQUENCE sec_codcen_con");
$result2 = pg_exec("DROP SEQUENCE sec_coddes_con");
$result3 = pg_exec("CREATE SEQUENCE sec_coddes_con");
$result2 = pg_exec("DROP SEQUENCE sec_codequi_con");
$result3 = pg_exec("CREATE SEQUENCE sec_codequi_con");
$result2 = pg_exec("DROP SEQUENCE sec_codnyb_con");
$result3 = pg_exec("CREATE SEQUENCE sec_codnyb_con");
$result2 = pg_exec("DROP SEQUENCE sec_coddti_con");
$result3 = pg_exec("CREATE SEQUENCE sec_coddti_con");

pg_close();

?>
```

Anexo B. Scripts de Definición de Permisos

```

<?
pg_connect("host=localhost user=postgres dbname=telepasto");

//CREACION DE PERMISOS PARA TRANSMISION
$result2 = pg_exec("GRANT ALL ON
estaciones_tra,enlaces_tra,red_tra,equipos_tra,tipos_tarjetas_tra,tarjetas_tra,referencia_tra,estados_
tra,bodega_tra,informe_tra,baterias_tra,sec_codest_tra,sec_codenl_tra,sec_codred_tra,sec_codequ
_tra,sec_codtar_tra,sec_codtiptar_tra,sec_codref_tra,sec_codesta_tra,sec_codbod_tra,sec_codinf_t
ra TO GROUP administradores_tra");

$result2 = pg_exec("GRANT SELECT ON
estaciones_tra,enlaces_tra,red_tra,equipos_tra,tipos_tarjetas_tra,tarjetas_tra,referencia_tra,estados_
tra,bodega_tra,informe_tra,baterias_tra,sec_codest_tra,sec_codenl_tra,sec_codred_tra,sec_codequ
_tra,sec_codtar_tra,sec_codtiptar_tra,sec_codref_tra,sec_codesta_tra,sec_codbod_tra,sec_codinf_t
ra TO GROUP operadores_tra");

$result2 = pg_exec("GRANT ALL ON
estaciones_tra,enlaces_tra,red_tra,referencia_tra,estados_tra,bodega_tra,informe_tra,baterias_tra,se
c_codest_tra,sec_codenl_tra,sec_codred_tra,sec_codref_tra,sec_codesta_tra,sec_codbod_tra,sec_c
odinf_tra TO GROUP operadores_tra");

$result2 = pg_exec("GRANT SELECT ON
estaciones_tra,enlaces_tra,red_tra,equipos_tra,tipos_tarjetas_tra,tarjetas_tra,referencia_tra,estados_
tra,bodega_tra,informe_tra,baterias_tra,sec_codest_tra,sec_codenl_tra,sec_codred_tra,sec_codequ
_tra,sec_codtar_tra,sec_codtiptar_tra,sec_codref_tra,sec_codesta_tra,sec_codbod_tra,sec_codinf_t
ra TO GROUP publicos");

//CREACION DE PERMISOS PARA TELEMATICA

$result2 = pg_exec("GRANT ALL ON
clientes_tel,equipos_tel,tarjetas_tel,configuraciones_tel,canales_tel,enlaces_tel,eventos_tel,funciona
rios,estados_tel,tipos_tarjetas_tel,tipos_equipos_tel,sec_codest,sec_codtiptar,sec_codtipequ,sec_co
dcli,sec_codequi,sec_codtar,sec_codconf,sec_codcan,sec_codenl,sec_codeven,sec_codfun TO
GROUP administradores_tel");

```

```
$result2 = pg_exec("GRANT ALL ON
clientes_tel,canales_tel,eventos_tel,funcionarios,sec_codcli,sec_codcan,sec_codeven,sec_codfun
TO GROUP operadores_tel");

$result2 = pg_exec("GRANT SELECT ON
equipos_tel,tarjetas_tel,configuraciones_tel,enlaces_tel,estados_tel,tipos_tarjetas_tel,tipos_equipos
_tel,sec_codest,sec_codtiptar,sec_codtipecu,sec_codequi,sec_codtar,sec_codconf,sec_codenl TO
GROUP operadores_tel");

$result2 = pg_exec("GRANT SELECT ON
clientes_tel,equipos_tel,tarjetas_tel,configuraciones_tel,canales_tel,enlaces_tel,eventos_tel,funciona
rios,estados_tel,tipos_tarjetas_tel,tipos_equipos_tel,sec_codest,sec_codtiptar,sec_codtipecu,sec_co
dcli,sec_codequi,sec_codtar,sec_codconf,sec_codcan,sec_codenl,sec_codeven,sec_codfun TO
GROUP publicos");

$result2 = pg_exec("GRANT ALL ON funcionarios,sec_codfun TO GROUP
administradores_con");

$result2 = pg_exec("GRANT ALL ON funcionarios,sec_codfun TO GROUP
administradores_ene");

$result2 = pg_exec("GRANT ALL ON funcionarios,sec_codfun TO GROUP
administradores_tra");

$result2 = pg_exec("GRANT ALL ON funcionarios,sec_codfun TO GROUP operadores_con");

$result2 = pg_exec("GRANT ALL ON funcionarios,sec_codfun TO GROUP operadores_ene");

$result2 = pg_exec("GRANT ALL ON funcionarios,sec_codfun TO GROUP operadores_tra");

$result2 = pg_exec("GRANT SELECT ON funcionarios,sec_codfun TO GROUP publicos");

$result2 = pg_exec("GRANT ALL ON
solucion_uk,tipo_modem,tipo_radio,tipo_fo,tipo_e1,tipo_tran,sec_coduk,sec_codmod,sec_codrad
,sec_code1,sec_codtran,sec_codfo TO GROUP administradores_tel");

$result2 = pg_exec("GRANT ALL ON
solucion_uk,tipo_modem,tipo_radio,tipo_fo,tipo_e1,tipo_tran,sec_coduk,sec_codmod,sec_codrad
,sec_code1,sec_codtran,sec_codfo TO GROUP operadores_tel");

$result2 = pg_exec("GRANT SELECT ON
solucion_uk,tipo_modem,tipo_radio,tipo_fo,tipo_e1,tipo_tran,sec_coduk,sec_codmod,sec_codrad
,sec_code1,sec_codtran,sec_codfo TO GROUP publicos");

//CREACION DE PERMISOS PARA ENERGIA
$result2 = pg_exec("GRANT ALL ON energia_ene,sec_cod_ene TO GROUP
administradores_ene");
```

```
$result2 = pg_exec("GRANT ALL ON energia_ene,sec_cod_ene TO GROUP operadores_ene");  
$result2 = pg_exec("GRANT SELECT ON energia_ene,sec_cod_ene TO GROUP publicos");  
  
//CREACION DE PERMISOS PARA CONMUTACION  
  
$result2 = pg_exec("GRANT ALL ON  
centrales_con,descripciones_con,equipos_con,numeracion_bandas_con,dti_neax_con,sec_codcen  
_con,sec_coddes_con,sec_codequi_con,sec_codnyb_con,sec_coddti_con TO GROUP  
administradores_con");  
  
$result2 = pg_exec("GRANT ALL ON  
centrales_con,descripciones_con,numeracion_bandas_con,sec_codcen_con,sec_coddes_con,sec_c  
odnyb_con TO GROUP operadores_con");  
  
$result2 = pg_exec("GRANT SELECT ON  
equipos_con,dti_neax_con,sec_codequi_con,sec_coddti_con TO GROUP operadores_con");  
  
$result2 = pg_exec("GRANT SELECT ON  
centrales_con,descripciones_con,equipos_con,numeracion_bandas_con,dti_neax_con,sec_codcen  
_con,sec_coddes_con,sec_codequi_con,sec_codnyb_con,sec_coddti_con TO GROUP publicos");  
  
pg_close();  
  
?>
```

Anexo C. Scripts de Creación de Reglas

```
<?
pg_connect("host=localhost user=postgres dbname=telepasto");

//CREACION DE REGLAS
$result2 = pg_exec("CREATE RULE tipomodem AS ON DELETE TO tipo_modem DO
DELETE FROM solucion_uk WHERE OLD.cod_uk=cod_uk");
$result2 = pg_exec("CREATE RULE tiporadio AS ON DELETE TO tipo_radio DO DELETE
FROM solucion_uk WHERE OLD.cod_uk=cod_uk");
$result2 = pg_exec("CREATE RULE tipoe1 AS ON DELETE TO tipo_e1 DO DELETE
FROM solucion_uk WHERE OLD.cod_uk=cod_uk");
$result2 = pg_exec("CREATE RULE tipofo AS ON DELETE TO tipo_fo DO DELETE
FROM solucion_uk WHERE OLD.cod_uk=cod_uk");
$result2 = pg_exec("CREATE RULE tipotran AS ON DELETE TO tipo_tran DO DELETE
FROM solucion_uk WHERE OLD.cod_uk=cod_uk");

?>
```

Anexo D. Scripts de Parte del Código de las Páginas Web del Proyecto

- El siguiente Script pertenece al código de la página Web “**frm_clientes.php**”, diseñada para Insertar, Eliminar, Actualizar Clientes de Telemática.

```
<html>
<head>
<title>Clientes Telemática</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body text="#000000" bgcolor="f7f7f7">
<SCRIPT LANGUAGE="JavaScript">
    function javaclick()
    {
        document.form2.encontro1.disabled=false;
    }
    function confirma()
    {
        document.form1.acepta.value=confirm("Realmente desea Eliminar este
registro?");
    }
</SCRIPT>
<?php
global
$txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo,$link,$usuario,$password,$acepta;

//FUNCION VALIDAR
function validar()
{
    global $txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo;
    $sw1=true;
    $ptel=$txttel;
    $pnit=$txtnit;
    settype($txttel,"integer");
    settype($txtnit,"integer");
    settype($txttel,"string");
```

```

settype($txtnit,"string");
if ($txtnit=="" or $txtrazon=="" or $txtnom=="" or $txtape=="" or $txtmdir=="")
{
    ?>
    <SCRIPT LANGUAGE="JavaScript">
        alert("Todos los campos con ** son Necesarios");
    </SCRIPT>
    <?
    $sw1=false;
}
else
{
    if (($txttel==$ptel) and ($txtnit==$pnit))
    {
        $sw1=true;
    }
    else
    {
        ?>
        <SCRIPT LANGUAGE="JavaScript">
            alert("Los campos nit y telefono solo deben tener numeros");
        </SCRIPT>
        <?
        $sw1=false;
    }
}
$txttrazon=strtoupper($txtrazon);
$txtnom=strtoupper($txtnom);
$txtape=strtoupper($txtape);
$txtmdir=strtoupper($txtmdir);
return $sw1;
}

//FUNCION LIMPIAR
function limpiar()
{
    global $txtnit,$txtrazon,$txtnom,$txtape,$txtmdir,$txttel,$codigo;
    $txtrazon="";
    $txtnit="";
    $txtnom="";
    $txtape="";
    $txtmdir="";
    $txttel="";
    $codigo="";
}

//FUNCION ADICIONAR
function adicionar()

```

```

{
    global $txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo;
    $sw=repetidos();
    if ($sw==false)
    {
        $result = pg_exec("INSERT INTO clientes_tel
(cod_cli,nit_cli,razon_cli,nom_cli,ape_cli,dir_cli,tel_cli) VALUES
(NEXTVAL('sec_codcli'),' $txtnit',' $txtrazon',' $txtnom',' $txtape',' $txtdir',' $txttel)");
        if ($result)
        {
            ?>
            <SCRIPT LANGUAGE="JavaScript">
            alert("Registro Insertado");
            </SCRIPT>
            <?
        }
        else
        {
            echo pg_errormessage();
        }
    }
    else
    {
        ?>
        <SCRIPT LANGUAGE="JavaScript">
        alert("Cliente Ya Existe");
        </SCRIPT>
        <?
    }
} //100

//FUNCION MOSTRAR
function mostrar()
{
    global $txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo,$op;
    if ($txtrazon=="")
    {
        ?>
        <SCRIPT LANGUAGE="JavaScript">
        alert("Debe ingresar Razon Social");
        </SCRIPT>
        <?
        limpiar();
        $op=true;
        forma();
    }
    else
    {

```

```

    $txtrazon=strtoupper($txtrazon);
    $result=pg_exec("SELECT * FROM clientes_tel WHERE razon_cli like
'%$txtrazon%'");
    if ($result)
    {
        $num=pg_NumRows($result);
        if ($num < 1)
        {
            ?>
            <SCRIPT LANGUAGE="JavaScript">
                alert("Registro No Encontrado");
            </SCRIPT>
            <?
                limpiar();
                $op=true;
                forma();
            }
        }
        else
        {
            if ($num > 1)
            {
                ?>
                <div id="Layer10" style="position:absolute; left:10px; top:20px; width:300px;
height:30px; z-index:13">
                    <div align="Left">
                        <p><b><font face="long cool woman" size="7"
color="927e57">Clientes Telematica</font></b></p>
                    </div>
                    <div>
                        <form id="form2" name="form2" method="post"
action="frm_clientes.php">
                            <div id="Layer10" style="position:absolute; left:10px; top:80px; width:300px;
height:30px; z-index:13">
                                <FONT FACE='alien encounters' SIZE='2'
COLOR='0a1070'>CLIENTES REGISTRADOS</FONT>
                            </div>
                            <div id="Layer1" style="position:absolute; left:10px; top:60px; width:600px;
height:10px; z-index:1">
                                <hr>
                            </div>
                            <div align=center style="position:absolute; overflow:scroll; left:10px;
top:110px; width:600px; height:170px;z-index:12">
                                <? $i=0;
                                    echo "<TABLE FRAME=BELOW WIDTH=400
ALIGN=LEFT CELLBORDER=0 COLSPAN=4 CELLPADDING=0
CELLSPACING=0>";
                                    echo "<TR ALIGN=LEFT>";
                                    echo "<TH BGCOLOR=12b7ff WIDTH=15>";

```

```

        echo "<TH BGCOLOR=12b7ff WIDTH=180><FONT
face='ARIAL NARROW ITALIC' SIZE='2'>Razon Social</FONT>";
        echo "<TH BGCOLOR=12b7ff WIDTH=50><FONT
face='ARIAL NARROW ITALIC' SIZE='2'>Nit</FONT>";
        echo "<TH BGCOLOR=12b7ff WIDTH=150><FONT
face='ARIAL NARROW ITALIC' SIZE='2'>Responsable</FONT>";
        while($i<$num)
        {
            $txtrazon=pg_result($result, $i, "razon_cli");
            $txtnit=pg_result($result, $i, "nit_cli");
            $codigo=pg_result($result, $i, "cod_cli");
            $txtnom=pg_result($result, $i, "nom_cli");
            $txtape=pg_result($result, $i, "ape_cli");
        echo "<TR>";
        echo "<TD BGCOLOR=ffffff><input type='radio' name='codigo'
value=$codigo onclick='javaclick()'>";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'> $txtrazon";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'> $txtnit ";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'> $txtnom $txtape";
            $i++;
        }
        ?>
    </table>
</div>
<div align=right style="position:absolute; left:430px; top:80px">
    <input type="submit" name="encontro" id="encontro1"
value="Aceptar " disabled>
</div>
<div align=right style="position:absolute; left:520px; top:80px">
    <input type="submit" name="cancelar" value="Cancelar">
</div>
</form>
<?
}
else
{
    $txtrazon=pg_result($result, 0, "razon_cli");
    $txtnit=pg_result($result, 0, "nit_cli");
    $txtnom=pg_result($result, 0, "nom_cli");
    $txtape=pg_result($result, 0, "ape_cli");
    $txtdir=pg_result($result, 0, "dir_cli");
    $xttel=pg_result($result, 0, "tel_cli");
    $codigo=pg_result($result, 0, "cod_cli");
    forma();
}

```

```

    }
  }
}
return $codigo;
}

//FUNCION BUSCAR
function buscar()
{
  global $txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo,$sw;
  $result=pg_exec("SELECT * FROM clientes_tel WHERE cod_cli=$codigo");
  $txtrazon=pg_result($result, 0, "razon_cli");
  $txtnit=pg_result($result, 0, "nit_cli");
  $txtnom=pg_result($result, 0, "nom_cli");
  $txtape=pg_result($result, 0, "ape_cli");
  $txtdir=pg_result($result, 0, "dir_cli");
  $txttel=pg_result($result, 0, "tel_cli");
  $sw=false;
}

//FUNCION REPETIDOS
function repetidos()
{
  global $txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo;
  $result=pg_exec("SELECT * FROM clientes_tel");
  $sw=false;
  if ($result)
  {
    $num=pg_NumRows($result);
    $i=0;
    while($i<$num)
    {
      $codigo=pg_result($result, $i, "cod_cli");
      $antnit=pg_result($result, $i, "nit_cli");
      if($txtnit==$antnit)
      {
        $i=$num;
        $sw=true;
      }
      $i++;
    }
  }
  return $sw;
}

//FUNCION ELIMINAR
function eliminar()

```

```

{
    global $txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo,$acepta;
    if($acepta)
    {
        $result=pg_exec("SELECT * FROM canales_tel WHERE cod_cli=$codigo");
        $num=pg_NumRows($result);
        if($num>0)
        {
            ?>
            <SCRIPT LANGUAGE="JavaScript">
                alert("Registro No puede ser Eliminado, Existe una referencia a este
Cliente en la tabla Canales");
            </SCRIPT>
            <?
        }
        else
        {
            $result=pg_exec("DELETE FROM clientes_tel WHERE
cod_cli=$codigo");
            ?>
            <SCRIPT LANGUAGE="JavaScript">
                alert("Registro Eliminado");
            </SCRIPT>
            <?
        }
    }
}

//FUNCION ACTUALIZAR
function actualizar()
{
    global $txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo;
    $result=pg_exec("UPDATE clientes_tel SET razon_cli='$txtrazon', nit_cli='$txtnit',
nom_cli='$txtnom', ape_cli='$txtape', dir_cli='$txtdir', tel_cli='$txttel' WHERE
cod_cli='$codigo'");
    ?>
    <SCRIPT LANGUAGE="JavaScript">
        alert("Registro Actualizada");
    </SCRIPT>
    <?
}

//FUNCION FORMA
function forma()
{
    global $txtnit,$txtrazon,$txtnom,$txtape,$txtdir,$txttel,$codigo,$op,$acepta;
    ?>

```

```

    <div id="Layer4" style="position:absolute; left:40px; top:80px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="Impact" size="2">Razon
Social </font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:110px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="Impact"
size="2">Nit</font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:140px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="Impact"
size="2">Nombres</font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:170px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="Impact"
size="2">Apellidos</font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:200px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="Impact"
size="2">Direccion</font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:230px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="Impact"
size="2">Telefono</font></div>
    </div>
    <div id="Layer2" style="position:absolute; left:10px; top:75px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:105px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:135px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:165px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:195px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:225px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer10" style="position:absolute; left:10px; top:20px; width:300px;
height:30px; z-index:13">
    <div align="Left">
    <p><b><font face="long cool woman" size="7" color="927e57">Clientes
Telematica</font></b></p>
    </div>

```

```

</div>
<div id="Layer1" style="position:absolute; left:10px; top:60px; width:600px;
height:80px; z-index:1">
<hr>
</div>
<div align=rigth style="position:absolute; left:550px; top:50px">
<A HREF=frm_general.htm><FONT FACE='alien encounters' SIZE='2'
COLOR='0a1070'>VOLVER</FONT></A>
</div>
<form id="form1" name="form1" method="post" action="frm_clientes.php">
<input type="hidden" name="codigo" value="<?php echo $codigo; ?>">
<input type="hidden" name="acepta" value="">
<div id="Layer7" style="position:absolute; left:128px; top:75px; width:162px;
height:25px; z-index:8">
<input type="text" name="txtrazon" size="30" value="<?php echo
$txtrazon;?>" maxlength="30">
</div>
<div id="Layer7" style="position:absolute; left:128px; top:105px; width:114px;
height:25px; z-index:9">
<input type="text" name="txtnit" value="<?php echo $txtnit;?>" size="20"
maxlength="30">
</div>
<div id="Layer7" style="position:absolute; left:128px; top:135px; width:161px;
height:25px; z-index:10">
<input type="text" name="txtnom" size="30" maxlength="30"
value="<?php echo $txtnom;?>">
</div>
<div id="Layer7" style="position:absolute; left:128px; top:165px; width:162px;
height:25px; z-index:11">
<input type="text" name="txtape" value="<?php echo $txtape;?>"
size="30" maxlength="30">
</div>
<div id="Layer7" style="position:absolute; left:128px; top:195px; width:163px;
height:25px; z-index:12">
<input type="text" name="txtdir" value="<?php echo $txtdir;?>" size="30"
maxlength="30">
</div>
<div id="Layer7" style="position:absolute; left:128px; top:225px; width:112px;
height:25px; z-index:13">
<input type="text" name="txttel" value="<?php echo $txttel;?>" size="20"
maxlength="30">
</div><?
if ($op==true)
{ ?>
<div id="Layer3" style="position:absolute; left:10px; top:260px;
width:76px; height:29px; z-index:14">
<input type="submit" name="adicionar" value=" Adicionar ">
</div>

```

```

                <div id="Layer9" style="position:absolute; left:110px; top:260px;
width:76px; height:29px; z-index:15">
                <input type="submit" name="eliminar" value=" Eliminar "
disabled>
                </div>
                <div id="Layer9" style="position:absolute; left:210px; top:260px;
width:80px; height:29px; z-index:16">
                <input type="submit" name="actualizar" value="Actualizar"
disabled>
                </div>
                <div id="Layer9" style="position:absolute; left:310px; top:260px;
width:76px; height:29px; z-index:17">
                <input type="submit" name="buscar" value=" Buscar ">
                </div> <?
            }
            else
            {
                $op=true;?>
                <div id="Layer3" style="position:absolute; left:10px; top:260px;
width:76px; height:29px; z-index:14">
                <input type="submit" name="adicionar" value=" Adicionar "
disabled>
                </div>
                <div id="Layer9" style="position:absolute; left:110px; top:260px;
width:72px; height:29px; z-index:15">
                <input type="submit" name="eliminar" value=" Eliminar "
onclick="confirma()">
                </div>
                <div id="Layer9" style="position:absolute; left:210px; top:260px;
width:80px; height:28px; z-index:16">
                <input type="submit" name="actualizar" value="Actualizar">
                </div>
                <div id="Layer9" style="position:absolute; left:310px; top:260px;
width:76px; height:28px; z-index:17">
                <input type="submit" name="buscar" value=" Buscar " disabled>
                </div>
                <?
            } ?>
        </form>
        <?
    }

// CUERPO DEL PROGRAMA
$link=pg_connect("host=telepasto.telecom.com.co user=$usuario password=$password
dbname=telepasto");
if($link)
{

```

```
if ($adicionar)
{
    $sw1=validar();
    if ($sw1==true)
    {
        adicionar();
    }
    limpiar();
    $op=true;
    forma();
}
else
{
    if ($buscar)
    {
        mostrar();
    }
    else
    {
        if ($actualizar)
        {
            $sw=validar();
            if ($sw==true)
            {
                actualizar();
            }
            limpiar();
            $op=true;
            forma();
        }
        else
        {
            if ($eliminar)
            {
                eliminar();
                limpiar();
                $op=true;
                forma();
            }
            else
            {
                if($encontro)
                {
                    buscar();
                    forma();
                }
                else
                {
```

```
        if($cancelar)
        {
            $op=true;
            forma();
        }
        else
        {
            $op=true;
            forma();
        }
    }
}
}
}
}
}
}
}
}
}
}
else
{
    echo $php_errormsg;
}
pg_close($link);
?>
<!-- CIERRA EL CUERPO !-->
</body>
</html>
```

- El siguiente Script pertenece al código de la página Web **“frm_consulta_cli.php”**, diseñada para Consultar Clientes de Telemática.

```
<html>
<head>
<title>Consulta Clientes Telemática</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body text="#000000" bgcolor="f7f7f7">
<?php
$link=pg_connect("host=telepasto.telecom.com.co user=$usuario password=$password
dbname=telepasto");
if ($link)
{
$result=pg_exec("SELECT * FROM clientes_tel");
if ($result)
{
```

```

?>
<div id="Layer10" style="position:absolute; left:10px; top:20px; width:300px;
height:30px; z-index:13">
  <div align="Left">
    <p><b><font face="long cool woman" size="7" color="927e57">Clientes
Telematica</font></b></p>
  </div>
</div>
<div id="Layer10" style="position:absolute; left:10px; top:80px; width:300px;
height:30px; z-index:13">
  <FONT FACE='alien encounters' SIZE='2' COLOR='0a1070'>CLIENTES
REGISTRADOS</FONT>
</div>
<div id="Layer1" style="position:absolute; left:10px; top:60px; width:600px;
height:10px; z-index:1">
  <hr>
</div>
<div align=right style="position:absolute; left:550px; top:50px">
  <A HREF=frm_general.htm><FONT FACE='alien encounters' SIZE='2'
COLOR='0a1070'>VOLVER</FONT></A>
</div>
<div align=center style="position:absolute; overflow:scroll; left:10px; top:110px;
width:600px; height:170px;z-index:12">
  <TABLE FRAME=BELOW WIDTH=580 ALIGN=LEFT CELLBORDER=0
COLSPAN=5 CELLPADDING=0 CELLSPACING=0>
  <TR ALIGN=LEFT>
  <TH BGCOLOR=12b7ff WIDTH=180><FONT face='ARIAL NARROW
ITALIC' SIZE='2'>Razon Social</FONT>
  <TH BGCOLOR=12b7ff WIDTH=50><FONT face='ARIAL NARROW ITALIC'
SIZE='2'>Nit</FONT>
  <TH BGCOLOR=12b7ff WIDTH=150><FONT face='ARIAL NARROW
ITALIC' SIZE='2'>Responsable</FONT>
  <TH BGCOLOR=12b7ff WIDTH=150><FONT face='ARIAL NARROW
ITALIC' SIZE='2'>Direccion</FONT>
  <TH BGCOLOR=12b7ff WIDTH=35><FONT face='ARIAL NARROW ITALIC'
SIZE='2'>Telefono </FONT>
  <?
  $num=pg_NumRows($result);
  if ($num < 1)
  {
    ?>
    <TR>
    <TD> Ningun Cliente encontrado
    <?
  }
else
{
  $i=0;

```

```

while($i<$num)
{
    $txtrazon=pg_result($result, $i, "razon_cli");
    $txtnit=pg_result($result, $i, "nit_cli");
    $txtnom=pg_result($result, $i, "nom_cli");
    $txtape=pg_result($result, $i, "ape_cli");
    $txtdir=pg_result($result, $i, "dir_cli");
    $txttel=pg_result($result, $i, "tel_cli");
    echo "<TR>";
    echo "<TD BGCOLOR=ffffff ><font face='ARIAL' size='1'
color='0f03a9'> $txtrazon";
    echo "<TD BGCOLOR=ffffff ><font face='ARIAL' size='1'
color='0f03a9'> $txtnit ";
    echo "<TD BGCOLOR=ffffff ><font face='ARIAL' size='1'
color='0f03a9'> $txtnom $txtape";
    echo "<TD BGCOLOR=ffffff ><font face='ARIAL' size='1'
color='0f03a9'> $txtdir";
    echo "<TD BGCOLOR=ffffff ><font face='ARIAL' size='1'
color='0f03a9'> $txttel";
    $i++;
}
?>
</TABLE>
</div>
<?
}
}
}
else
{
    echo $php_errormsg;
}
pg_close($link);
?>
<!-- CIERRA EL CUERPO !-->
</body>
</html>

```

- El siguiente Script pertenece al código de la página Web **“frm_funcionarios.php”**, diseñada para Insertar, Eliminar, Actualizar los Funcionarios de Telecom

```

<html>
<head>
<title>Funcionarios Telecom</title>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body text="#000000" bgcolor="f7f7f7">
<SCRIPT LANGUAGE="JavaScript">
    function javaclick()
    {
        document.form2.encontro1.disabled=false;
    }
</SCRIPT>
<?php
global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo;

//FUNCION VALIDAR
function validar()
{
    global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo;
    $sw1=true;
    $ptel=$txttel;
    settype($txttel,"integer");
    settype($txttel,"string");
    if ($ciudad=="" or $txtnom=="" or $txtape=="" or $seccion=="" or $area=="" or
$cargo=="")
    {
        ?>
        <SCRIPT LANGUAGE="JavaScript">
            alert("Todos los campos con ** son Necesarios");
        </SCRIPT>
        <?
        $sw1=false;
    }
    else
    {
        if ($txttel==$ptel)
        {
            $sw1=true;
        }
        else
        {
            ?>
            <SCRIPT LANGUAGE="JavaScript">
                alert("El Telefono solo deben tener numeros");
            </SCRIPT>
            <?
            $sw1=false;
        }
    }
    $ciudad=strtoupper($ciudad);
    $txtnom=strtoupper($txtnom);
}

```

```

$txtape=strtoupper($txtape);
$seccion=strtoupper($seccion);
$cargo=strtoupper($cargo);
$area=strtoupper($area);
$ciudad=strtoupper($ciudad);
$txtnom=trim($txtnom);
$txtape=trim($txtape);
$seccion=trim($seccion);
$cargo=trim($cargo);
$area=trim($area);
return $sw1;
}

//FUNCION LIMPIAR
function limpiar()
{
    global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo;
    $txtnom="";
    $txtape="";
    $ciudad="";
    $seccion="";
    $area="";
    $txttel="";
    $cargo="";
    $codigo="";
}

//FUNCION ADICIONAR
function adicionar()
{
    global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo;
    $sw=repetidos();
    if ($sw==false)
    {
        $result = pg_exec("INSERT INTO funcionarios
(cod_fun,nom_fun,ape_fun,mun_fun,seccion_fun,area_fun,cargo_fun,tel_fun) VALUES
(NEXTVAL('sec_codfun'),' $txtnom',' $txtape',' $ciudad',' $seccion',' $area',' $cargo',' $txttel)");
        if ($result)
        {
            ?>
            <SCRIPT LANGUAGE="JavaScript">
            alert("Registro Insertado");
            </SCRIPT>
            <?
        }
        else
        {
            echo pg_errormessage();
        }
    }
}

```

```

    }
  }
  else
  {
    ?>
    <SCRIPT LANGUAGE="JavaScript">
      alert("Funcionario Ya Registrado");
    </SCRIPT>
    <?
  }
}

//FUNCION MOSTRAR
function mostrar()
{
  global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo,$op;
  if ($txtnom=="")
  {
    ?>
    <SCRIPT LANGUAGE="JavaScript">
      alert("Debe ingresar un Nombre o Apellido");
    </SCRIPT>
    <?
    limpiar();
    $op=true;
    forma();
  }
  else
  {
    $txtnom=strtoupper($txtnom);
    $txtape=strtoupper($txtape);
    $txtnom=trim($txtnom);
    $txtape=trim($txtape);
    $result=pg_exec("SELECT * FROM funcionarios WHERE nom_fun like
'%$txtnom%'");
    if ($result)
    {
      $num=pg_NumRows($result);
      if ($num < 1)
      {
        ?>
        <SCRIPT LANGUAGE="JavaScript">
          alert("Registro No Encontrado");
        </SCRIPT>
        <?
        limpiar();
        $op=true;
      }
    }
  }
}

```

```

        forma();
    }
    else
    {
        if ($num > 1)
        {
            ?>
            <div id="Layer10" style="position:absolute; left:10px; top:20px; width:300px;
height:30px; z-index:13">
                <div align="Left">
                    <p><b><font face="long cool woman" size="7"
color="927e57">Funcionarios Telecom</font></b></p>
                    </div>
                </div>
                <form id="form2" name="form2" method="post"
action="frm_funcionarios.php">
                    <div id="Layer10" style="position:absolute; left:10px; top:80px; width:300px;
height:30px; z-index:13">
                        <FONT FACE='alien encounters' SIZE='2'
COLOR='0a1070'>FUNCIONARIOS REGISTRADOS</FONT>
                    </div>
                    <div id="Layer1" style="position:absolute; left:10px; top:60px; width:600px;
height:10px; z-index:1">
                        <hr>
                    </div>
                    <div align=center style="position:absolute; overflow:scroll; left:10px;
top:110px; width:600px; height:170px;z-index:12">
                        <? $i=0;
                        echo "<TABLE FRAME=BELOW WIDTH=600 ALIGN=LEFT
CELLBORDER=0 COLSPAN=5 CELLPADDING=0 CELLSPACING=0>";
                        echo "<TR ALIGN=LEFT>";
                        echo "<TH BGCOLOR=12b7ff WIDTH=15>";
                        echo "<TH BGCOLOR=12b7ff WIDTH=150><font face='ARIAL
NARROW ITALIC' size='2'>Nombres";
                        echo "<TH BGCOLOR=12b7ff WIDTH=150><font face='ARIAL
NARROW ITALIC' size='2'>Apellidos";
                        echo "<TH BGCOLOR=12b7ff WIDTH=65><font
face='ARIAL NARROW ITALIC' size='2'>Ciudad";
                        echo "<TH BGCOLOR=12b7ff WIDTH=90><font
face='ARIAL NARROW ITALIC' size='2'>Seccion";
                        echo "<TH BGCOLOR=12b7ff WIDTH=130><font
face='ARIAL NARROW ITALIC' size='2'>Area";
                        while($i<$num)
                        {
                            $txtnom=pg_result($result, $i, "nom_fun");
                            $txtape=pg_result($result, $i, "ape_fun");
                            $codigo=pg_result($result, $i, "cod_fun");
                            $ciudad=pg_result($result, $i, "mun_fun");
                            $seccion=pg_result($result, $i, "seccion_fun");

```

```

        $area=pg_result($result, $i, "area_fun");
    echo "<TR>";
    echo "<TD BGCOLOR=ffffff WIDTH=15><input type='radio'
name='codigo' value=$codigo onclick='javaclick()>";
    echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'> $txtnom";
    echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'> $txtape";
    echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'> $ciudad";
    echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'> $seccion";
    echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'> $area";
        $i++;
    }
    ?>
</table>
</div>
<div align=righth style="position:absolute; left:430px; top:80px">
    <input type="submit" name="encontro" id="encontro1"
value="Aceptar " disabled>
</div>
<div align=righth style="position:absolute; left:520px; top:80px">
    <input type="submit" name="cancelar" value="Cancelar">
</div>
</form>
<?
}
else
{
    $txtnom=pg_result($result, 0, "nom_fun");
    $txtape=pg_result($result, 0, "ape_fun");
    $ciudad=pg_result($result, 0, "mun_fun");
    $seccion=pg_result($result, 0, "seccion_fun");
    $area=pg_result($result, 0, "area_fun");
    $txttel=pg_result($result, 0, "tel_fun");
    $carga=pg_result($result, 0, "carga_fun");
    $codigo=pg_result($result, 0, "cod_fun");
    forma();
}
}
}
}
return $codigo;
}

```

```
//FUNCION BUSCAR
```

```

function buscar()
{
    global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo,$sw;
    $result=pg_exec("SELECT * FROM funcionarios WHERE cod_fun=$codigo");
    $txtnom=pg_result($result, 0, "nom_fun");
    $txtape=pg_result($result, 0, "ape_fun");
    $ciudad=pg_result($result, 0, "mun_fun");
    $seccion=pg_result($result, 0, "seccion_fun");
    $area=pg_result($result, 0, "area_fun");
    $cargo=pg_result($result, 0, "cargo_fun");
    $txttel=pg_result($result, 0, "tel_fun");
    $sw=false;
}

//FUNCION REPETIDOS
function repetidos()
{
    global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo;
    $result=pg_exec("SELECT * FROM funcionarios");
    $sw=false;
    if ($result)
    {
        $num=pg_NumRows($result);
        $i=0;
        while($i<$num)
        {
            $codigo=pg_result($result, $i, "cod_fun");
            $antnom=pg_result($result, $i, "nom_fun");
            $antape=pg_result($result, $i, "ape_fun");
            $antciudad=pg_result($result, $i, "mun_fun");
            if($txtnom.$txtape.$ciudad==$antnom.$antape.$antciudad)
            {
                $i=$num;
                $sw=true;
            }
            $i++;
        }
    }
    return $sw;
}

//FUNCION ELIMINAR
function eliminar()
{
    global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo;
    $result=pg_exec("DELETE FROM funcionarios WHERE cod_fun=$codigo");
    if($result)

```

```

    {
        ?>
        <SCRIPT LANGUAGE="JavaScript">
            alert("Registro Eliminado");
        </SCRIPT>
        <?
    }
else
    {
        ?>
        <SCRIPT LANGUAGE="JavaScript">
            alert($php_errormsg);
        </SCRIPT>
        <?
    }
}

//FUNCION ACTUALIZAR
function actualizar()
{
    global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo;
    $result=pg_exec("UPDATE funcionarios SET seccion_fun='$seccion',
area_fun='$area', mun_fun='$ciudad', cargo_fun='$cargo', tel_fun='$txttel' WHERE
cod_fun='$codigo'");
    ?>
    <SCRIPT LANGUAGE="JavaScript">
        alert("Registro Actualizada");
    </SCRIPT>
    <?
}

//FUNCION FORMA
function forma()
{
    global $ciudad,$seccion,$txtnom,$txtape,$area,$txttel,$codigo,$cargo,$op;
    ?>
    <div id="Layer4" style="position:absolute; left:40px; top:80px; width:75px;
height:20px; z-index:14">
        <div align="left"><font color="#000000" face="IMPACT"
size="2">Nombres </font></div>
        </div>
    <div id="Layer4" style="position:absolute; left:40px; top:110px; width:75px;
height:20px; z-index:14">
        <div align="left"><font color="#000000" face="IMPACT"
size="2">Apellidos</font></div>
        </div>

```

```

    <div id="Layer4" style="position:absolute; left:40px; top:140px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="IMPACT"
size="2">Ciudad</font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:170px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="IMPACT"
size="2">Seccion</font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:200px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="IMPACT"
size="2">Area</font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:230px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="IMPACT"
size="2">Cargo</font></div>
    </div>
    <div id="Layer4" style="position:absolute; left:40px; top:260px; width:75px;
height:20px; z-index:14">
    <div align="left"><font color="#000000" face="IMPACT"
size="2">Telefono</font></div>
    </div>
    <div id="Layer2" style="position:absolute; left:10px; top:75px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:105px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:135px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:165px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:195px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:225px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer2" style="position:absolute; left:10px; top:255px; width:105px;
height:25px; z-index:4"></div>
    <div id="Layer10" style="position:absolute; left:10px; top:20px; width:300px;
height:30px; z-index:13">
    <div align="Left">
    <p><b><font face="long cool woman" size="7"
color="927e57">Funcionarios Telecom</font></b></p>
    </div>
    </div>
    <div id="Layer1" style="position:absolute; left:10px; top:60px; width:600px;
height:80px; z-index:1">

```

```

<hr>
</div>
<div align=right style="position:absolute; left:550px; top:50px">
  <A HREF=frm_general.htm><FONT FACE='alien encounters' SIZE='2'
COLOR='0a1070'>VOLVER</FONT></A>
</div>
<form method="post" action="frm_funcionarios.php">
  <input type="hidden" name="codigo" value="<?php echo $codigo; ?>">
  <div id="Layer7" style="position:absolute; left:128px; top:75px; width:162px;
height:25px; z-index:8">
    <input type="text" name="txtnom" size="30" value="<?php echo $txtnom; ?>"
maxlength="30">
  </div>
  <div id="Layer7" style="position:absolute; left:128px; top:105px; width:114px;
height:25px; z-index:8">
    <input type="text" name="txtape" value="<?php echo $txtape; ?>" size="30"
maxlength="30">
  </div>
  <div id="Layer7" style="position:absolute; left:128px; top:135px; width:161px;
height:25px; z-index:8">
    <input type="text" name="ciudad" size="30" maxlength="30" value="<?php echo
$ciudad; ?>">
  </div>
  <div id="Layer7" style="position:absolute; left:128px; top:165px; width:162px;
height:25px; z-index:8">
    <input type="text" name="seccion" value="<?php echo $seccion; ?>" size="40"
maxlength="40">
  </div>
  <div id="Layer7" style="position:absolute; left:128px; top:195px; width:163px;
height:25px; z-index:8">
    <input type="text" name="area" value="<?php echo $area; ?>" size="40"
maxlength="40">
  </div>
  <div id="Layer7" style="position:absolute; left:128px; top:225px; width:112px;
height:25px; z-index:8">
    <input type="text" name="cargo" value="<?php echo $cargo; ?>" size="30"
maxlength="30">
  </div>
  <div id="Layer7" style="position:absolute; left:128px; top:255px; width:112px;
height:25px; z-index:8">
    <input type="text" name="txttel" value="<?php echo $txttel; ?>" size="12"
maxlength="12">
  </div>
  <?
    if ($op==true)
    { ?>
      <div id="Layer3" style="position:absolute; left:10px; top:290px;
width:76px; height:29px; z-index:10">

```

```

        <input type="submit" name="adicionar" value="Adicionar">
      </div>
      <div id="Layer9" style="position:absolute; left:100px; top:290px;
width:72px; height:29px; z-index:11">
        <input type="submit" name="eliminar" value="Eliminar" disabled>
      </div>
      <div id="Layer9" style="position:absolute; left:265px; top:290px;
width:76px; height:28px; z-index:11">
        <input type="submit" name="buscar" value="Buscar">
      </div>
      <div id="Layer9" style="position:absolute; left:175px; top:290px;
width:80px; height:28px; z-index:11">
        <input type="submit" name="actualizar" value="Actualizar" disabled>
      </div> <?
    }
  else
  {
    $op=true;?>
    <div id="Layer3" style="position:absolute; left:10px; top:290px;
width:76px; height:29px; z-index:10">
      <input type="submit" name="adicionar" value="Adicionar"
disabled>
    </div>
    <div id="Layer9" style="position:absolute; left:100px; top:290px;
width:72px; height:29px; z-index:11">
      <input type="submit" name="eliminar" value="Eliminar">
    </div>
    <div id="Layer9" style="position:absolute; left:265px; top:290px;
width:76px; height:28px; z-index:11">
      <input type="submit" name="buscar" value="Buscar" disabled>
    </div>
    <div id="Layer9" style="position:absolute; left:175px; top:290px;
width:80px; height:28px; z-index:11">
      <input type="submit" name="actualizar" value="Actualizar">
    </div>
    <?
  } ?>
</form>
<?
}

// CUERPO DEL PROGRAMA
$link=pg_connect("host=telepasto.telecom.com.co user=$usuario password=$password
dbname=telepasto");
if($link)
{

```

```
if ($adicionar)
{
    $sw1=validar();
    if ($sw1==true)
    {
        adicionar();
    }
    limpiar();
    $op=true;
    forma();
}
else
{
    if ($buscar)
    {
        mostrar();
    }
    else
    {
        if ($actualizar)
        {
            $sw=validar();
            if ($sw==true)
            {
                actualizar();
            }
            limpiar();
            $op=true;
            forma();
        }
        else
        {
            if ($eliminar)
            {
                eliminar();
                limpiar();
                $op=true;
                forma();
            }
            else
            {
                if($encontro)
                {
                    buscar();
                    forma();
                }
                else
                {
```

```

        if($cancelar)
        {
            $op=true;
            forma();
        }
        else
        {
            $op=true;
            forma();
        }
    }
}
}
else
{
    echo "erro";
}
pg_close($link);
?>
<!-- CIERRA EL CUERPO !-->
</body>
</html>

```

- El siguiente Script pertenece al código de la página Web **“frm_consulta_fun.php”**, diseñada para Consultar Funcionarios de Telecom.

```

<html>
<head>
<title>Canales Telematica</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body text="#000000" bgcolor="f7f7f7">
<?php
$link=pg_connect("host=telepasto.telecom.com.co user=$usuario password=$password
dbname=telepasto");
if ($link)
{
$result=pg_exec("SELECT * FROM funcionarios");
if ($result)
{

```

```

?>
<div id="Layer10" style="position:absolute; left:10px; top:20px; width:300px;
height:30px; z-index:13">
  <div align="Left">
    <p><b><font face="long cool woman" size="7"
color="927e57">Funcionarios Telecom</font></b></p>
  </div>
</div>
<div id="Layer10" style="position:absolute; left:10px; top:80px; width:300px;
height:30px; z-index:13">
  <FONT FACE='alien encounters' SIZE='2'
COLOR='0a1070'>FUNCIONARIOS REGISTRADOS</FONT>
</div>
<div id="Layer1" style="position:absolute; left:10px; top:60px; width:600px;
height:10px; z-index:1">
  <hr>
</div>
<div align=right style="position:absolute; left:550px; top:50px">
  <A HREF=frm_general.htm><FONT FACE='alien encounters' SIZE='2'
COLOR='0a1070'>VOLVER</FONT></A>
</div>
<div align=center style="position:absolute; overflow:scroll; left:10px; top:110px;
width:600px; height:170px;z-index:12">
  <TABLE FRAME=BELOW WIDTH=600 ALIGN=LEFT CELLBORDER=0
COLSPAN=7 CELLPADDING=0 CELLSPACING=0>
  <TR ALIGN=LEFT>
  <TH BGCOLOR=12b7ff WIDTH=130><FONT FACE='ARIAL NARROW
ITALIC' SIZE='2'>Nombres
  <TH BGCOLOR=12b7ff WIDTH=130><FONT FACE='ARIAL NARROW
ITALIC' SIZE='2'>Apellidos
  <TH BGCOLOR=12b7ff WIDTH=50><FONT FACE='ARIAL NARROW
ITALIC' SIZE='2'>Ciudad
  <TH BGCOLOR=12b7ff WIDTH=80><FONT FACE='ARIAL NARROW
ITALIC' SIZE='2'>Seccion
  <TH BGCOLOR=12b7ff WIDTH=80><FONT FACE='ARIAL NARROW
ITALIC' SIZE='2'>Area
  <TH BGCOLOR=12b7ff WIDTH=80><FONT FACE='ARIAL NARROW
ITALIC' SIZE='2'>Cargo
  <TH BGCOLOR=12b7ff WIDTH=50><FONT FACE='ARIAL NARROW
ITALIC' SIZE='2'>Telefono
  <?
$num=pg_NumRows($result);
if ($num < 1)
{
  ?>
  <TR>
  <TD> Ningun Cliente encontrado
  <?

```

```

    }
else
{
    $i=0;
    while($i<$num)
    {
        $txtnom=pg_result($result, $i, "nom_fun");
        $txtape=pg_result($result, $i, "ape_fun");
        $ciudad=pg_result($result, $i, "mun_fun");
        $seccion=pg_result($result, $i, "seccion_fun");
        $area=pg_result($result, $i, "area_fun");
        $txttel=pg_result($result, $i, "tel_fun");
        $cargo=pg_result($result, $i, "cargo_fun");
        echo "<TR >";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'>$txtnom";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'>$txtape";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'>$ciudad";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'>$seccion";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'>$area";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'>$carga";
        echo "<TD BGCOLOR=ffffff><font face='ARIAL' size='1'
color='0f03a9'>$txttel";
        $i++;
    }
    ?>
</table>
</div>
<?

}
}
}
else
{
    echo $php_errormsg;
}
pg_close($link);
?>
<!-- CIERRA EL CUERPO !-->
</body>
</html>

```

Anexo E. Referencias Internet

- www.linux.org.mx. Documentación, Artículos, Noticias, Grupos de Usuarios, Proyectos, etc. del Sistema Operativo Linux.
- www.redhat.com. Distribución de Linux RedHat, Documentación en general.
- www.apache.org. Distribución de Servidor Web Apache, Documentación en general.
- www.mx.postgresql.org. Distribución del Motor de Base de Datos Postgresql, Documentación en general.
- www.php.org.mx. Distribución del Lenguaje de Desarrollo, Documentación en general.
- www.es.qmail.org. El más grande portal de Internet para qmail.
- www.macromedia.com. Distribución de productos Macromedia (Flash, Fireworks, Dreamweaver), Documentación en general.
- www.lucas.hispalinux.es. Interesante sitio de Internet, donde se puede encontrar las traducciones de muchos manuales de distintos software para Linux.
- www.osmosislatina.com. Portal de Internet con innumerables guías de Instalación, configuración de software para Linux.

•

