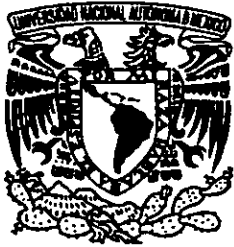


31



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
CAMPUS ARAGÓN**

**DISEÑO E IMPLANTACIÓN DE UNA INTRANET CON  
HERRAMIENTAS DE DOMINIO PÚBLICO.  
CASO DE APLICACIÓN: INTRANET DEL  
INSTITUTO NACIONAL DE ECOLOGÍA**

*gastaldi*

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACION**

**P R E S E N T A:  
FRANCISCO JAVIER ROJAS ROMERO**

**ASESOR: ING. JUAN GASTALDI PEREZ**





Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis padres por su apoyo y motivación. Esto es gracias a ustedes.

A mis hermanas Claudia, Arianda y Gricel para que sirva como un ejemplo, más no como una obligación, de lo que se puede hacer con esfuerzo y dedicación.

Y a Dios por permitirme lograr una de las metas más importantes de mi vida.

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO I. CONCEPTOS BÁSICOS RELACIONADOS CON LAS INTRANETS</b> .....	<b>5</b>
1.1 ARQUITECTURA CLIENTE/SERVIDOR .....	5
1.1.1 Definición de arquitectura cliente/servidor .....	5
1.1.2 Características del cliente .....	7
1.1.3 Características del servidor .....	8
1.1.4 Ventajas y desventajas sobre otros modelos.....	10
1.1.5 Sistemas cliente/servidor vs. sistemas de tiempo compartido .....	12
1.1.6 Ejemplos de sistemas cliente/servidor .....	13
1.2 SISTEMAS DISTRIBUIDOS.....	14
1.2.1 Sistemas abiertos .....	16
1.2.2 Arquitectura de los sistemas distribuidos .....	18
1.3 PROTOCOLOS TCP/IP.....	20
1.3.1 La estructura de TCP/IP .....	21
1.4 LA WORLD WIDE WEB.....	28
1.4.1 ¿Cómo funciona WWW? .....	29
1.5 PROTOCOLO DE TRANSFERENCIA DE HYPER TEXTO (HTTP) .....	33
1.5.1 Descripción del protocolo HTTP.....	34
1.5.2 Comandos.....	37
1.5.3 Encabezados.....	38
1.5.4 La Transmisión de la información y el ancho de banda .....	39
1.5.5 Servidor HTTP .....	40
1.5.6 Cliente HTTP .....	40
1.6 LENGUAJE DE MARCADO DE HYPER TEXTO (HTML).....	41
1.7 INTERNET .....	43
1.7.1 Breve historia .....	43
1.7.2 ¿Qué es Internet? .....	46
1.7.3 Principales servicios de Internet.....	47
<b>CAPÍTULO II. ASPECTOS DISTINTIVOS DE LAS INTRANETS</b> .....	<b>52</b>
2.1 ANTECEDENTES.....	52
2.2 EL CONCEPTO DE INTRANET.....	53
2.3 RAZONES PARA LA IMPLANTACIÓN DE INTRANETS .....	54
2.4 VENTAJAS QUE OFRECEN LAS INTRANETS.....	55
2.5 CARACTERÍSTICAS DE LAS INTRANETS.....	58
2.6 DIFERENCIA CON LAS SOLUCIONES PARA TRABAJO EN GRUPO .....	61
2.7 TECNOLOGÍAS WEB Y COMPONENTES BÁSICOS .....	62
2.7.1 Tecnologías Web.....	62
2.7.2 Componentes básicos .....	63
2.8 APLICACIONES DE LAS INTRANETS .....	64
2.8.1 Posibilidades comerciales.....	67
<b>CAPÍTULO III. RECURSOS DISPONIBLES EN EL INSTITUTO NACIONAL DE ECOLOGÍA</b> .....	<b>69</b>
3.1 RESUMEN DEL PROGRAMA DE DESARROLLO INFORMÁTICO CONTENIDO EN EL PLAN NACIONAL DE DESARROLLO 1995-2000 .....	69
3.1.1 Introducción .....	69
3.1.2 La Informática en el Plan Nacional de Desarrollo 1995-2000.....	72
3.1.3 Aprovechamiento de la Informática en los diversos sectores.....	73
3.1.4 El Desarrollo de la Infraestructura Informática .....	77
3.1.5 Mecanismos de coordinación y seguimiento.....	81
3.2 EL INSTITUTO NACIONAL DE ECOLOGÍA.....	82

3.2.1 Semblanza histórica.....	82
3.2.2 Normas Oficiales Mexicanas.....	85
3.2.3 Principios y Orientaciones.....	86
3.3 RED DEL INSTITUTO NACIONAL DE ECOLOGÍA.....	90
3.3.1 Topologías.....	90
3.3.2 Redes tipo Ethernet.....	92
3.3.3 Token Ring.....	93
3.3.4 FDDI.....	95
3.3.5 Distribución física de la red del INE.....	96
3.4 EQUIPO Y SOFTWARE DISPONIBLE ACTUALMENTE.....	99
3.4.1 Comunicación con el exterior.....	103
3.4.2 Equipo de cómputo y software instalado.....	106
<b>CAPÍTULO IV. SELECCIÓN DE HERRAMIENTAS DE DOMINIO PÚBLICO PARA LA CONSTRUCCIÓN DE UNA INTRANET.....</b>	<b>109</b>
4.1 SOFTWARE DE DOMINIO PÚBLICO.....	109
4.2 EL PROYECTO GNU Y EL SOFTWARE LIBRE.....	110
4.3 SELECCIÓN DE UN SERVIDOR DE WEB.....	113
4.3.1 ¿Cómo se eligió el servidor de Web?.....	114
4.4 SELECCIÓN DE UN LENGUAJE DE PROGRAMACIÓN PARA CGI'S.....	118
4.4.1 CGI. Common Gateway Interface.....	118
4.4.2 ¿Cómo se eligió el lenguaje de programación para CGI's?.....	120
<b>CAPÍTULO V. CASO DE APLICACIÓN: INTRANET DEL INSTITUTO NACIONAL DE ECOLOGÍA.....</b>	<b>123</b>
5.1 DELIMITACIÓN Y JUSTIFICACIÓN DEL PROYECTO.....	123
5.2 PLANTEAMIENTO DE LA NECESIDAD.....	125
5.3 INSTALANDO Y CONFIGURANDO PHP PARA UNIX.....	126
5.3.1 Obteniendo PHP.....	126
5.3.2 Instalación de PHP.....	126
5.4 INSTALANDO Y CONFIGURANDO APACHE PARA UNIX.....	129
5.4.1 Obteniendo Apache.....	129
5.4.2 Compilando Apache.....	130
5.4.3 Configuración del servidor Apache.....	133
5.4.4 Dando de alta y de baja el servidor.....	136
5.5 CREACIÓN DE FORMAS Y APLICACIONES EN HTML Y PHP.....	138
5.5.1 Página principal de la Intranet del INE.....	140
5.5.2 Directorio telefónico.....	142
5.5.3 Foros de discusión.....	149
5.5.4 Salas de discusión en tiempo real (Chat).....	161
5.5.5 Resúmenes informativos.....	168
5.5.6 Aplicaciones.....	170
5.5.7 Informática.....	172
5.5.8 Capacitación.....	173
5.5.9 Clarín informático.....	174
5.6 SEGURIDAD Y MANTENIMIENTO.....	175
5.6.1 Seguridad.....	175
5.6.2 Instalación de la autenticación de usuarios.....	178
5.6.3 Mantenimiento.....	185
<b>CONCLUSIONES.....</b>	<b>187</b>
<b>BIBLIOGRAFÍA.....</b>	<b>191</b>
<b>ANEXOS.....</b>	<b>198</b>
Anexo I. Estructura de archivos y directorios de PHP.....	198

Anexo 2. Listado de la ejecución del script <i>setup</i> .....	201
Anexo 3. Listado de la ejecución del comando <i>make</i> de PHP.....	208
Anexo 4. Listado de la ejecución del comando <i>make</i> de Apache.....	213
Anexo 5. Listado del archivo <i>httpd.conf</i> .....	216
Anexo 6. Listado del archivo <i>srm.conf</i> .....	219
Anexo 7. Listado del archivo <i>access.conf</i> .....	223
Anexo 8. Listado de la página principal de la Intranet del INE.....	225
Anexo 9. Listado del archivo <i>directorio.phtml</i> .....	227
Anexo 10. Listado del archivo de mensajes personalizado <i>form.var</i> .....	232
Anexo 11. Listado del script <i>chat.phtml</i> .....	233
Anexo 12. Listado del script <i>input.phtml</i> .....	239
Anexo 13. Listado del archivo <i>refresh.phtml</i> .....	244
Anexo 14. Listado del archivo <i>index.phtml</i> . Sección Resúmenes Informativos.....	246
Anexo 15. Listado del archivo <i>index.phtml</i> . Sección Aplicaciones.....	251
Anexo 16. Listado del archivo <i>index.phtml</i> . Sección Informática.....	253
Anexo 17. Listado del archivo <i>index.phtml</i> . Sección Capacitación.....	254
Anexo 18. Listado del archivo <i>index.phtml</i> . Sección Clarin Informático.....	255
GLOSARIO.....	257

## *INTRODUCCIÓN*

---

Hoy en día los campos de las redes de computadoras y de las telecomunicaciones son de las áreas que más demanda tienen tanto en el sector público como en el privado, todos los sectores reconocen que las tecnologías de Internet marcarán el rumbo de las actividades laborales y económicas, pero en la realidad, son pocos los que se atreven a utilizarlas como una auténtica plataforma de trabajo.

Una empresa normalmente tiene cientos o miles de necesidades de aplicaciones, grandes o pequeñas, muchas de las cuales nunca han podido resolverse porque no existía un método universal para que los empleados individuales tuvieran un acceso instantáneo a los recursos de información o a las aplicaciones desde el escritorio, y no existía un medio para desarrollar y desplegar nuevas aplicaciones con rapidez, sin problemas y de un modo universal.

Por eso el propósito de este proyecto es explicar y mostrar lo que actualmente se está utilizando en el mundo de las Tecnologías de Información para satisfacer una serie de tareas así como las necesidades de comunicación del sector empresarial, de gobierno o privado, es decir, estamos hablando de las Intranets. Esto no es nuevo, ya existe y está cambiando la forma de hacer negocios de las empresas.

Las empresas que usan Intranets se encuentran con que una combinación de aplicaciones nativas, personalizadas y comerciales, basadas en el entorno de la Intranet, resuelven fácilmente sus necesidades de comunicación. Expresado en términos sencillos, una Intranet es la red TCP/IP de una empresa, que enlaza a sus empleados y su información de tal manera que: aumenta su productividad, facilita el acceso a la información, y convierte la navegación por los recursos y aplicaciones de su entorno informático en un proceso más infalible de lo que era.

El término Intranet y todas las maravillas que se puedan hacer con una red que tenga toda la información que se requiere para tomar decisiones, manejar la operación de una empresa, ponerse

al tanto de la industria y capturar la atención de los empleados de una manera dirigida, son algunas de las ventajas de contar con esta poderosa herramienta.

Actualmente los empresarios requieren soluciones que contribuyan a que la operación y los procesos de la empresa se vuelvan efectivos para que los grupos de trabajo se mantengan en unión constante, en colaboración para gestar con ello un nuevo sistema de colaboración que además se mantenga en actualización en cuanto a innovación tecnológica con el fin de integrar estas herramientas a sus procesos productivos siempre que sea necesario.

De tal forma que resulta imprescindible que los empleados cuenten con la información necesaria y la habilidad de ayudarse entre ellos, basándose en la inteligencia y la experiencia que resulte de utilizar adecuadamente los recursos, para que puedan así distribuir estos conocimientos con toda la planta productiva.

Actualmente las Intranets están generando más interés que el mismo Web en EUA y Canadá. Por un lado las compañías han notado que un sitio de Web requiere de infraestructura interna de soporte para su uso eficiente. Además, el Web le ha abierto las puertas de las oficinas corporativas a expertos en Internet en general, los cuales han tenido la oportunidad de mostrar el beneficio de otros aspectos de Internet además del popular Web. De hecho, más de la mitad de las ventas de Netscape, los creadores del popular navegador, en el último cuarto fueron servidores Web para uso en Intranet. La revista *Fortune* de EU, especializada en tópicos que son de interés para el ejecutivo, recientemente incluyó un artículo<sup>1</sup> en el que enfatizan algunos de los puntos mencionados en esta sección, e ilustran con algunos ejemplos el uso de Intranets en corporaciones privadas.

Este panorama hace suponer que las Intranets tienden a erigirse como toda una transición hacia un nuevo y revolucionario vehículo de transmisión de información, en el que los antiguos procesos intraempresariales se sustituyen por sus equivalentes en lenguajes operativos de informática.

---

<sup>1</sup> Sprout, A.L., The Internet inside your company, *Fortune*, 27 de noviembre de 1995.  
<http://www.pathfinder.com/fortune/magazine/1995/951127/infotech.net.html>



En la actualidad la implantación de una Intranet es muy costosa si no se cuenta con los recursos tecnológicos y humanos necesarios, además de que existe un gran número de empresas que ofrecen sus productos (*software*) y servicios para la construcción, sin mencionar el monto económico que todo esto implica; aquí hablaremos de cómo funcionan las Intranets, qué necesidades cubren y cómo implantarla con herramientas gratuitas y aprovechando los recursos disponibles.

Se demostrará en el caso de aplicación, que realizando una investigación y habiendo conocido las necesidades de un sector, en particular del Instituto Nacional Ecología, en cuanto a comunicación interna, se ha logrado construir una Intranet, con los servicios básicos y específicos, utilizando los recursos disponibles y con un costo mínimo, ya que todo el *software* utilizado en su funcionamiento es de dominio público, en otras palabras, gratuito.

Para lograr este proyecto se seguirán una serie de procedimientos, los cuales se listan en los 5 capítulos de esta tesis.

En el capítulo I denominado “**Conceptos básicos relacionados con las Intranets**” se describe la tecnología fundamental usada tanto por la Web como por la Intranet. Se explica cómo trabajan y que se puede hacer para “sacar” el mayor provecho de la Intranet, mencionando todas las tecnologías que interactúan de alguna manera para hacer posible la funcionalidad y tener como objetivo primordial una Intranet. Se describen de manera general sus características, lo que ofrece y puede lograrse con el uso de este tipo de tecnologías, programas y protocolos, así como un panorama general de todos los elementos necesarios que hacen que una Intranet se utilice como una herramienta de comunicación interna.

En el capítulo II, “**Aspectos distintivos de las Intranets**”, se explican las razones por las que se está aplicando la implantación de Intranets en las empresas a partir de la tecnología Internet en uso, así como las características, ventajas, componentes, etc. de las Intranets. Se describen las

aplicaciones típicas de las mismas para organizaciones comerciales y otras no necesariamente vinculadas al mundo de las tecnologías de la información. Posteriormente se discuten las diferencias con las soluciones de mercado para trabajo en grupos.

El capítulo III, "**Recursos disponibles en el Instituto Nacional de Ecología**", analiza el papel que juega el Instituto Nacional de Ecología dentro de la sociedad y de los distintos recursos de infraestructura tanto de *software* como de *hardware* con los que cuenta para la implantación de la Intranet. Se describen las características de los medios de transmisión que actualmente se utilizan tanto de servidores SUN como de ruteadores y concentradores que utilizan como medio de conexión a Internet.

En el capítulo IV, "**Selección de herramientas de dominio público para la construcción de una Intranet**", se describen las herramientas de dominio público y de los fundamentos que se tomaron en cuenta para la decisión del servidor de Web y del lenguaje de programación para los CGI's.

En el capítulo V, "**Caso de aplicación: Intranet del Instituto Nacional de Ecología**", se explica el diseño de la Intranet para el Instituto Nacional de Ecología, así como la importancia que tiene la integración en una red existente y qué características son útiles. La configuración e instalación de cada una de las herramientas así como la creación de los CGI's y las formas en HTML para la Intranet. Se expone la importancia del mantenimiento del servidor y de los documentos; además de describir precauciones que se tomaran para el acceso a la información. Este capítulo es el más importante de la tesis, puesto que se detalla el trabajo realizado en cada una de las etapas de la construcción hasta llegar a la liberación de una Intranet con funcionamiento exitoso que permitió dar por terminado el proyecto.

## *CAPÍTULO I*

### *CONCEPTOS BÁSICOS RELACIONADOS CON LAS INTRANETS*

---

#### **1.1 Arquitectura cliente/servidor**

En un sistema cliente/servidor, uno o más clientes y uno o más servidores, junto con el sistema operativo y los protocolos de comunicación, conforman el ambiente que permite y facilita el cómputo distribuido.

En una aplicación basada en esta arquitectura existen dos procesos independientes, en lugar de uno solo. De esta forma se puede repartir el trabajo a través de varias computadoras en una red. Estos dos procesos, cliente y servidor, se comunican mediante un protocolo definido. Esta técnica permite la comunicación entre distintas computadoras (servidores de archivos, estaciones de trabajo con alta calidad de graficación, etc.), para que cada una de ellas se dedique a realizar el trabajo que hace mejor.

De manera introductoria, se puede decir que un servidor es un sistema o un programa que provee de algún servicio a otros sistemas a través de una red. Un ejemplo típico es un servidor de archivos, que permite el acceso a información remota a cualquier usuario a través de la red. Un cliente es un sistema que requiere y recibe alguna acción de un servidor.

##### **1.1.1 Definición de arquitectura cliente/servidor**

La arquitectura cliente/servidor es una forma de cómputo en red en el que ciertas funciones solicitadas por clientes son servidas por los procesos adecuados.

Un *servidor* es un administrador de recursos. Un *recurso* puede ser identificado como algo físico (por ejemplo, una impresora) o abstracto, para el caso de una base de datos. Un *cliente* es un usuario de los recursos que administra el servidor.

Bajo este esquema, se reparte el proceso de una aplicación entre un *front end*<sup>2</sup> y un *back end*<sup>3</sup>. En el estándar de la ciencia de la computación, un *proceso* es un programa corriendo en un procesador.

De manera general, para que se inicie la comunicación entre un cliente y servidor es necesario establecer una sesión. Por lo tanto, el servidor debe estar esperando (escuchando) que algún cliente trate de establecer una sesión. Esto quiere decir que un cliente puede “hablar” pero si no es “escuchado”, la comunicación va a fracasar. Es muy posible que, por algún momento el servidor también “hable” y que el cliente “escuche”, pero esto sólo se hará cuando el servidor así se lo indique al cliente.

Un servidor también se reserva el derecho de establecer comunicación con uno o más clientes. Así, el servidor se encargará de atender a cada cliente y establecer los mecanismos para la distribución de sus servicios. Un servidor define operaciones que son exportadas a los clientes. Los clientes invocan estas operaciones para que el servidor controle el manejo de datos. Típicamente, una aplicación (cliente) comenzará una transacción (mediante una sesión), ejecutará una o varias operaciones en el servidor y terminará la transacción (terminando la sesión). Lógicamente, los servidores están estructurados como un ciclo infinito. El servidor simplemente recibe los requerimientos de los clientes para invocar operaciones en favor de esas transacciones. Para implantar las operaciones que exporta, el servidor puede requerir de otro servidor o puede manipular sus propios datos.

---

<sup>2</sup> El cliente, que puede ser una PC o una estación de trabajo.

<sup>3</sup> El servidor.

### 1.1.2 Características del cliente

En un sistema cliente/servidor, un cliente es un proceso que interactúa con el usuario, observando las siguientes características:

*a) Presenta la interfaz del usuario (UI)*

Esta interfaz permite al usuario introducir sus consultas para recuperación y análisis de datos, así como recibir los resultados de dichas consultas, típicamente en un ambiente gráfico GUI (Graphic User Interface).

*b) Forma consultas o comandos en un lenguaje predefinido para su presentación al servidor.*

El cliente y el servidor pueden usar un lenguaje estándar, como el SQL (Structured Query Language), o un lenguaje propietario predefinido.

*c) Se comunica con el servidor por medio de una metodología de comunicación de procesos determinada y transmite consultas o comandos al servidor.*

Un cliente ideal hace que esta comunicación con el servidor sea transparente al usuario.

*d) Realiza análisis de datos sobre los resultados de la consulta o comando que regresan del servidor y lo presenta al usuario.*

La naturaleza y el grado de análisis que se ejecuta en el cliente pueden variar de un sistema a otro.

Las características *b)* y *d)* marcan la diferencia entre cliente y las terminales tontas que se encuentran conectadas a un *host*, dado que los clientes, como ya vimos, deben poseer capacidades de procesamiento.

Por otra parte, la última característica no debe ser confundida con el tipo de procesamiento que se da en una red local, donde se toman del servidor de archivos todos los elementos necesarios para ser procesados localmente y responder al requerimiento del usuario.

Existen además elementos clave que deben ser considerados en relación al cliente:

- *Sistema Operativo de la estación de trabajo.*

Se deben diseñar sistemas cliente/servidor que soporten diversos sistemas operativos en las estaciones de trabajo, como lo es UNIX.

- *Consideraciones respecto al hardware.*

El cliente debe ser tan pequeño como sea posible para que pueda ser soportado por estaciones de trabajo.

- *Consideraciones respecto a la conectividad.*

Estas consideraciones influyen también en el diseño del *software* del cliente. Un diseño modular asegurará que las aplicaciones en los clientes sean portables a diversas plataformas tanto de *software* como de *hardware*.

- *GUI.*

La interfaz al usuario es definida, principalmente, por el objetivo del sistema cliente/servidor y por el sistema operativo de la estación de trabajo. En general, una interfaz del usuario debe ser lo más modular posible para que el cliente pueda pasar de un sistema operativo a otro.

### **1.1.3 Características del servidor**

En un sistema cliente/servidor, un servidor es un proceso o conjunto de procesos que deben existir en un equipo que da servicio a uno o más clientes. Tiene las siguientes características:

- 1) *Un servidor provee servicio al cliente.*

La naturaleza y el grado de este servicio es definido por el objetivo del sistema cliente/servidor. Así mismo, un servicio puede requerir un mínimo de computación en el

servidor, como es el caso de los servidores de archivos o de los servidores de impresión, o necesitar de procesamiento intensivo, como se da en los servidores de bases de datos.

2) *Un servidor solamente responde a las consultas o comandos de los clientes.*

Esto es, ningún servidor inicia la conversación con un cliente, tampoco atiende directamente interfases con el usuario final. Simplemente actúa como repositorio de datos (servidor de archivos), o de conocimiento (servidores de bases de datos) o como prestador de servicios de impresión. Sin embargo, un servidor sí puede iniciar la conversación con otro servidor, solicitándole un servicio que a su vez le permitirá atender el requerimiento de un cliente. Esto, por supuesto, debe ser transparente para el usuario.

3) *Un servidor ideal hace transparente, todo el esquema cliente/servidor al cliente y al usuario.*

Un cliente que se comunica con un servidor no tiene por qué estar enterado de la plataforma de *hardware* y *software* que intenta acceder, así como de la tecnología de comunicación (*hardware* y *software*) que hace posible ese enlace. Para ello, es deseable y recomendable que en un ambiente de servidores múltiples, los servidores se comunican entre sí para proporcionar un servicio al cliente sin que éste conozca de esta múltiple existencia, ni de la comunicación entre servidores.

Así, una arquitectura cliente/servidor divide la aplicación en procesos separados que corren en distintas máquinas enlazadas por una red. Es por ello que el diseñador de aplicaciones divide las tareas en subtareas a ser llevadas a cabo ya sea por el cliente o por los servidores, teniendo como únicas limitantes las facilidades que le ofrezca el sistema operativo de red, así como las reglas de la organización.

Entre más avanzado sea un sistema operativo de red, las aplicaciones serán más pequeñas y fáciles de desarrollar en menos tiempo. Además, el hecho de que en el servidor resida la información de la organización permite incrementar la seguridad, pues se establecen mejores controles de acceso a la misma. Existen además elementos clave que deben ser considerados en relación con el servidor:

- *Escalabilidad*

Un servidor debe ser escalable, es decir, debe soportar números crecientes de clientes. Por ejemplo, en el caso de los servidores de bases de datos, el diseño debe permitir el crecimiento a un sistema de bases de datos distribuidas.

- *Interfaz con el servidor.*

Es importante que las características del servidor (tanto en *software* como en *hardware*), sean transparentes a los clientes, excepto en lo que se refiere a la interfaz estándar de acceso a sus servicios, como puede ser SQL, lo que además protege a los clientes de cambios en la tecnología.

#### **1.1.4 Ventajas y desventajas sobre otros modelos**

Un sistema cliente/servidor ofrece soluciones a las desventajas de los clientes centralizados tanto en *mainframes* como en servidores locales, descritos anteriormente, al tener las siguientes características:

- *Es inteligente a nivel de equipo de escritorio, ya que el cliente es el responsable de la interfaz con el usuario.*

El cliente transforma las consultas y/o comandos del usuario a un lenguaje predefinido que es comprendido por el servidor y presenta los resultados que el servidor le envía como respuesta, con lo que se obtiene mayor capacidad de proceso a un menor costo.



- *Permite compartir los recursos del servidor de manera óptima.*

Estos recursos pueden ser tanto el procesador, como almacenamiento secundario y periféricos. Un cliente puede pedir al servidor que realice procesos grandes, o puede pedirle que corra grandes aplicaciones (servidores de bases de datos), y como fruto obtener solamente los resultados de ese proceso.

- *Utilización óptima de la red.*

Dado que los clientes se comunican con el servidor a través de un lenguaje predefinido, como puede ser SQL, y el servidor sólo remite al cliente los resultados de la consulta o comando, reduciendo el tráfico en la red de los sistemas centralizados que transfieren los archivos de datos en su totalidad.

- *Permite cierta independencia sobre el sistema operativo y los protocolos de comunicación.*

Esto facilita el mantenimiento de las aplicaciones y asegura su portabilidad.

- *Permite desarrollos más flexibles a un menor costo.*

Hoy, la mayoría de los componentes de un sistema cliente/servidor, están disponibles comercialmente, provenientes de diversos proveedores, lo que da a las organizaciones libertad de elección, sin por ello exentarlas de los desarrollos internos.

Con el crecimiento de sistemas cliente/servidor, los proveedores se dirigen por el camino de los sistemas abiertos, que son fáciles de integrar y que requieren de menor trabajo para ser incluidos al sistema de la organización.

Entre los puntos débiles de una arquitectura cliente/servidor, tenemos que existe poca experiencia con ella, parte del *software* disponible se encuentra en su versión *beta* y el ya liberado tiene algunos errores (*bugs*); además, las facilidades para la administración de los datos y herramientas para la seguridad de los mismos deben ser mejoradas con el fin de ofrecer sistemas confiables.

### 1.1.5 Sistemas cliente/servidor vs. sistemas de tiempo compartido

El procesamiento en tiempo compartido utilizando los grandes sistemas de cómputo (*mainframe* por ejemplo), no es siempre la mejor solución para todas las aplicaciones. Un buen ejemplo, se presenta con el advenimiento de los sistemas con interfaces gráficas, que demandan una respuesta instantánea del subsistema gráfico, requiriendo un procesador dedicado y muy posiblemente personal.

El poder de las nuevas estaciones de trabajo, que comúnmente cuentan con un monitor a color de alta resolución y que opcionalmente cuentan con capacidades de multimedia, hacen que los conceptos y las tendencias del cómputo se modifiquen.

Un sistema de tiempo compartido, provee a los usuarios un ambiente en el que se comparten los recursos, tales como el espacio en disco, las impresoras, los programas y datos. Para que se compartan estos recursos en distintas estaciones de trabajo, éstas tienen que estar unidas a través de una red de cómputo. Por ejemplo, en la UNAM se tiene una red con un gran número de estaciones de trabajo, en la que los usuarios tienen que aprender a diferenciar entre lo local y remoto, saber en qué máquina se tienen tales y cuales recursos y programas disponibles. Esto nos lleva a encontrarnos con una serie de problemas.

El problema de administración de la red y de equipos conectados a ella, se convierte en un problema enorme. En el ambiente de estaciones de trabajo, cada usuario debe ser operador, administrador y programador de sistemas. Esto debido obviamente, a que ni los administradores de sistemas, ni los operadores pueden hacerse cargo de 100 máquinas al mismo tiempo. En un sistema de tiempo compartido se ha llegado a un grado de madurez en estos aspectos: un equipo de trabajo se dedica a la operación del sistema, otro a la administración y en otro equipo se desarrollan los programas de aplicación necesarios para las labores de los usuarios finales.

Se han implantado a la fecha algunas soluciones a estos problemas, pero no existe una solución tan adecuada como a la que se llegó en un tiempo con los sistemas de tiempo compartido. Se tienen ya comandos para copias de archivos en una red, y aún mejor, existen posibilidades de manejar archivos remotos. Desgraciadamente en la mayoría de los casos, el usuario debe estar

consciente de la diferencia entre un sistema local y uno remoto. El problema radica en que los sistemas operativos de hoy no se diseñaron para ser *sistemas operativos distribuidos*.

### 1.1.6 Ejemplos de sistemas cliente/servidor

#### *Servicio de datos estructurado*

Uno de los mejores ejemplos de la arquitectura cliente/servidor, está dada en un ambiente de base de datos, en donde el cliente es responsable de la interacción humana y formulación de consultas y actualizaciones a la base de datos. SQL (Lenguaje de consulta estructurado) será el lenguaje que interactuará entre esas consultas y actualizaciones con el servidor.

#### *X Window. Sistema de clientes y servidores.*

El papel del sistema X Window es el administrar los recursos físicos en la máquina del usuario, por ejemplo los subsistemas gráficos y los dispositivos de entrada. El sistema corriendo el servidor X, recibe como entrada estándar el teclado, *ratón* o cualquier otro dispositivo, la cual es producida en las formas especificadas por el protocolo X y pasadas al proceso del cliente apropiado. El cliente puede tener un papel general, por ejemplo, como administrador de ventanas que pueda controlar el tamaño, localización y visibilidad de las mismas, o una función muy específica, como la copia de un archivo de un disco a un servidor X.

#### *Servicios del Daemon Internet (inetd)*

Dentro de la familia de protocolos de Internet, existen cientos de ellos que ofrecen servicios, partiendo de los más fundamentales hasta los generalmente utilizados como el protocolo de transferencia de archivos *FTP*.

Sin *inetd*, un sistema podría prácticamente soportar sólo un pequeño subconjunto de estos servicios (esto es cierto porque la mayoría de ellos necesitan estar corriendo para responder las peticiones de los clientes). Con *inetd*, existen archivos de configuración (*/etc/services* y */etc/inetd.conf*), que son una lista de todos los servicios soportados por el sistema, los números de

puertos de *TCP* para establecer la comunicación cliente/servidor, y el programa asociado a este servicio.

## 1.2 Sistemas distribuidos

En los años 70's, las redes de computadoras comenzaron a ser un aspecto importante en los sistemas de cómputo. Manejadas inicialmente en aerolíneas e industria militar, los sistemas de cómputo fueron interconectados comenzando así, con las primeras operaciones remotas. Durante la década de los años 80's, los sistemas distribuidos comenzaron a ser un aspecto vital en muchos sistemas de cómputo. La gente comenzaba a depender de los sistemas conectados en red para sus transacciones financieras, registros y correo electrónico entre oficinas. Esta dependencia fue incrementándose rápidamente y con ella, las limitantes del trabajo en red. La década de los años 90's brincaré completamente a un esquema de red donde los sistemas distribuidos serán muy importantes; la gente podrá acceder a recursos de cómputo independientemente de su localización física.

En el proceso electrónico de información, el concepto de red implica la interconexión de computadoras. El tipo de conexión puede variar desde línea telefónica hasta enlace vía satélite. Su tamaño también puede ser variable, el cual puede ir desde un pequeño departamento hasta un edificio, e incluso llegar a una compleja red internacional con cientos de miles de usuarios usando sistemas diferentes. Usaremos la frase *cómputo distribuido* para enfatizar el concepto general de sistemas de cómputo interconectados, esta definición intenta incluir nociones del procesamiento distribuido en un modelo cliente/servidor, llamadas a procedimientos remotos, etc.

El concepto de red por lo tanto, es más que la interconexión de sistemas de cómputo. El valor de un sistema en red, depende de la capacidad de transferencia de información entre computadoras y la habilidad de producirla combinando los sistemas conectados. Referimos esta noción como parte en la definición de un *sistema distribuido*, aunque este último significa muchas cosas para la gente, nosotros usaremos este concepto para referimos a la noción general del acceso de una computadora a los recursos de otra, por ejemplo; discos, impresoras, etc., el usuario no distingue

entre operaciones locales y remotas, los programas no necesariamente se ejecutan en la computadora donde el comando fue especificado.

Los sistemas distribuidos y redes de computadoras tienen sus orígenes en la comunicación de datos. La tarea de transferencia de información hacia un equipo de cómputo central demandaba una solución electrónica en la década de los años 50's, varios mecanismos fueron desarrollados, entre ellos las primeras terminales. Posteriormente esos desarrollos permitieron a los sistemas, comunicarse con otros a grandes velocidades de transmisión de datos, dando origen a las primeras redes de computadoras con las topologías y protocolos de comunicación que actualmente conocemos. Los sistemas distribuidos, sin embargo, no aparecieron sino hasta la década de los 80's, donde emergieron nuevas tecnologías y el precio de las computadoras fue más económico, aparecieron las primeras computadoras personales y estaciones de trabajo, cada una de ellas contaba con su propio microprocesador de tal forma que se podían conectar con casi cualquier sistema de cómputo en la red, con la ventaja adicional que este tipo de sistemas no dependen de una computadora maestra para que los controle, sino que pueden trabajar independientemente, por lo tanto, fueron reemplazando a las terminales simples. Por otra parte, la introducción de las primeras *GUI's* y el *ratón*, dieron origen a un nuevo conjunto de usuarios, quienes comenzaron a disfrutar de un tiempo de respuesta consistente e independencia asociada en sus propias computadoras personales, silenciosamente demandaban los recursos de los grandes sistemas. La compatibilidad de recursos llegó a ser un objetivo de los diseñadores de sistemas, y los avances ocurrieron en muchas áreas; mecanismos de transferencia de archivos, ejecución de comandos remotos, entre otros.

La demanda de acceso remoto a recursos no es tarea fácil, las ventajas económicas asociadas con las estaciones de trabajo y computadoras personales no han extendido por completo este ambiente, el *software* requerido para administrar la distribución y compatibilidad de recursos es complejo. La cantidad de administradores de red, administradores de sistemas y desarrollo de *software* para aplicaciones distribuidas determinan el paso de las innovaciones en los sistemas distribuidos.

Por otra parte, el desarrollo paralelo en comunicaciones y tecnología de redes han acelerado la conexión de estaciones de trabajo y equipos de cómputo en general, estos desarrollos están cambiando la vida de las personas. Por ejemplo, actualmente una persona puede obtener dinero desde un cajero automático usando alguna de las varias redes, independientemente de la localización física de la institución bancaria. Otro ejemplo, es cuando un agente de viajes tiene acceso a las principales bases de datos de aerolíneas, hoteles, etc., independientemente de su localización geográfica.

### 1.2.1 Sistemas abiertos

Uno de los conceptos que actualmente tiene mucho apogeo es sin duda alguna el de *Sistemas Abiertos*. Este concepto es difícil de definir, pero de acuerdo a la IEEE:

*“Un Sistema Abierto es aquel que soporta un conjunto complejo de estándares tecnológicos internacionales de información, acoplado interoperabilidad y portabilidad de aplicaciones, datos y gente”.*

En este contexto, el término estándar se refiere a una especificación, un conjunto de funciones, o una implantación física que ha sido aceptada ampliamente formal o informalmente como un método preferido o interfaz. No necesariamente significa el mejor sino el más deseable.

El concepto de sistema abierto por otro lado, es una filosofía, una actitud, un punto de diseño y una característica de los sistemas. El objetivo es compartir información y tecnología fomentando su intercambio, eliminando redundancia y minimizar la necesidad de que la gente aprenda diferentes caminos para hacer la misma cosa.

Muchas organizaciones han instalado una variedad de plataformas, y en cada una de ellas manejan información necesaria para la vida de la organización. El reto de los sistemas abiertos, y con ellos los estándares, está en hacer que esas diferentes plataformas trabajen de manera conjunta. Los estándares permiten mejorar la compatibilidad, dejan mayor libertad al usuario para elegir lo que más convenga a sus necesidades (tanto en *hardware* como en *software*), permiten

explotar nuevas tecnologías y generan un ambiente de competencia tanto en precio como en desempeño dentro del mercado.

A pesar de que los estándares de programación y de comunicaciones han existido y evolucionado por décadas, UNIX de AT&T fue el primer sistema operativo que funcionó en diversas plataformas con éxito, permitiendo escalabilidad, portabilidad e interoperabilidad, como principales ventajas.

Sin embargo, existen varias versiones de UNIX, que no son totalmente compatibles entre sí. Es por ello que ahora, sistemas abiertos no es precisamente sinónimo de UNIX, sino de escalabilidad, portabilidad e interoperabilidad basada en estándares. Refiriéndonos específicamente a la familia de UNIX, este sistema operativo tiene como atributos significativos, los siguientes:

- *Una arquitectura abierta.*

UNIX fue diseñado para ser expansible tanto en *hardware* como *software*.

- *Disponibilidad de código fuente.*

Este aspecto es especialmente interesante en universidades y pequeñas compañías para poderlo modificar de acuerdo a sus intereses.

- *Portabilidad del Sistema Operativo.*

El diseño e implantación del sistema es tal que el trabajo de adaptarlo a otra máquina es relativamente mínimo.

- *Portabilidad de aplicaciones.*

El sistema y las herramientas asociadas con el ambiente de trabajo están escritos en lenguaje C. Este lenguaje está diseñado también para correr en múltiples arquitecturas. Por lo tanto, el *software* correspondiente a un sistema UNIX, puede ser recompilado en otro sistema. Esta

portabilidad probablemente ha sido el factor más importante en el crecimiento y popularidad del sistema operativo UNIX.

### 1.2.2 Arquitectura de los sistemas distribuidos

Hay algunos conceptos básicos que forman parte del diseño de *software* en los sistemas distribuidos, los cuales deben tomarse en cuenta como aspectos principales de cualquier sistema de cómputo. En este contexto discutiremos los más importantes:

- *Transparencia.*

La red debería ser “invisible” para los usuarios, programas de aplicación y administradores.  
¿Cuáles son las ventajas y limitantes?

- *Integración.*

¿Cómo pueden ser relacionadas las diferentes máquinas actualmente en un ambiente distribuido?

- *Funcionamiento.*

Últimamente, cualquier sistema de cómputo debe “ganarse” su lugar dentro de la red, con un funcionamiento que justifique su permanencia.

- *Balanceando carga.*

Los sistemas distribuidos ofrecen el poder de esparcir la demanda de servicios entre la colección de máquinas disponibles en el ambiente de red corporativo.



## **Transparencia**

La transparencia es un aspecto fundamental que forma parte del diseño, configuración, *software* y consideraciones operativas en un sistema distribuido. Por un lado, las computadoras en una red operan como sistemas individuales, con interacciones limitadas a hacer una conexión remota y transferencia de archivos por ejemplo. En esos casos, el usuario y administrador están explícitamente enterados de la existencia de la red y deben manejar sus errores directamente. Por otro lado, se contempla un sistema operativo de red, con el objetivo de permitir a usuarios, programadores y administradores tratar a una colección grande de computadoras, como si este conjunto fuera uno solo. Obviamente, este sistema está compuesto de múltiples componentes capaces de operar individualmente en forma eficiente.

Así por ejemplo, en el caso de UNIX, un sistema operativo corriendo en algunas computadoras podría tener, jerárquicamente un sólo sistema de archivos identificados con una sola "raíz".

## **Integración**

La forma, naturaleza de servicios disponibles y funcionamiento son profundamente afectados al tratar de asociarlos con otras máquinas que componen al sistema distribuido. El grado de integración es afectado por un número de consideraciones: madurez en la tecnología de *software*, el grado de similitud entre los sistemas involucrados y el nivel de administración presente, así como aspectos tecnológicos tales como el ancho de banda de la red.

Hay numerosas formas mediante las cuales los servicios remotos se hacen disponibles a una computadora determinada. La más simple y comúnmente usada es manejar el concepto de conectar una terminal de una computadora a otra utilizando un *software* especializado. Este método, también conocido como emulación de terminal ha sido usado por casi una tercera parte de este siglo y se aplica principalmente a la transferencia de archivos. En este caso, el usuario y programas asociados actúan directamente con las máquinas involucradas.

Después, fue posible el acceso remoto a archivos manejándolos con las mismas operaciones que uno usaría para un archivo local por ejemplo: abrir, cerrar, leer y escribir. Así, usuarios y programas no tienen que emplear diferentes medios para acceder a datos remotos.

Posteriormente, continúa una etapa en la que se pueden ejecutar programas remotamente entre máquinas compartiendo así los primeros recursos entre las mismas.

### **Funcionamiento**

Cualquier sistema de cómputo debe proveer un nivel de funcionamiento lo suficientemente alto para satisfacer las necesidades del usuario.

### **Balanceando la carga**

Es importante aprovechar los recursos conjuntamente de las estaciones de trabajo que forman parte de nuestra red. Tal vez algunas puedan realizar una tarea específica y otras puedan desplegar los resultados, etc.

## **1.3 Protocolos TCP/IP<sup>4</sup>**

Las Intranet son redes IP, así como la Internet, diseñada para ser usada dentro de una compañía. Las redes IP son utilizadas también en muchas empresas porque están basadas en sistemas abiertos. Esto permite que el *software* de las empresas adopte la tecnología IP y la incorpore dentro de sus productos. En efecto, las redes IP han tenido mucha popularidad en los últimos años, compañías que desarrollan estándares competitivos, como Novell y Microsoft, han tenido en su mercado demanda y han agregado IP como una opción de red.

El ruteo con IP está ahora incluido en Microsoft Windows 95, Microsoft Windows NT, OS/2 Warp, Warp Server, y Novell Netware. También está disponible para Windows para trabajo en

---

<sup>4</sup> Donde TCP e IP significan, respectivamente, *Transmission Control Protocol* e *Internet Protocol*

grupo. Windows 3.x y clientes Macintosh. El ruteo con IP está en casi todos los UNIX desde hace años, y es muy difícil encontrar versiones actuales de UNIX sin ruteo IP.

IP es usado en muchas aplicaciones como FTP (Protocolo de transferencia de archivos), NFS (Sistema de archivos en red), SMTP (para correo electrónico), y *rlogin* (conectarse a una máquina remotamente). Las redes IP usan dos tipos principales de transporte, el protocolo de control de transmisión (TCP) y el protocolo de datagrama de usuario (UDP). Los programas más usados como *telnet*, *rlogin*, o *ftp* usan la capa del TCP además de IP. de aquí el tan popular nombre de redes TCP/IP. UDP es usado más comúnmente en sistemas de archivos, como NFS, y servidores de nombres, como DNS.

Cuando un usuario, desde su computadora (conectada a Internet a través de una línea telefónica o una red local), visita alguna página de Web o envía un mensaje de correo electrónico, se realizan un sin número de pequeños procesos que tienen como objetivo conjunto transferir la información deseada y asegurar que dicha transmisión se realice libre de errores. Durante esta transmisión se utilizan varios protocolos. Al conjunto de estos protocolos se les conoce como conjunto de protocolos TCP/IP, o simplemente TCP/IP; que agrupa muchos protocolos, que implementan funciones a todos los niveles de las capas OSI<sup>5</sup> excepto el físico.

### 1.3.1 La estructura de TCP/IP

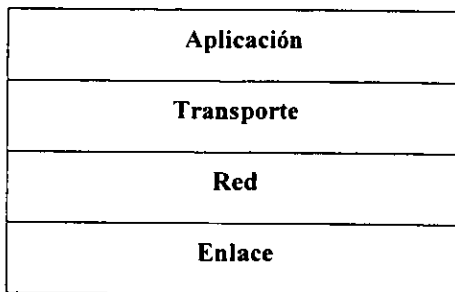
El modelo de comunicaciones de OSI define 7 capas. En cambio, TCP/IP utiliza 4, como lo muestra la figura 1.

Algunos autores prefieren llamar capa Internet a la capa de red y capa de acceso a la red a la capa de enlace. La correspondencia entre las capas de TCP/IP y las capas OSI es la siguiente: la capa de aplicación de TCP/IP corresponde a las capas de aplicación, presentación y sesión de las capas OSI; la capa de transporte de TCP/IP corresponde a la capa de transporte de OSI; la capa red de

---

<sup>5</sup> Modelo OSI: Open System Interconnect (Interconexión de Sistemas Abiertos)

TCP/IP corresponde a la capa de red y también se intercepta con la capa de enlace de OSI, por último, la capa enlace de TCP/IP corresponde a las capas de enlace y física de OSI.



**Figura 1** Estructura de capas TCP/IP.

Supóngase que un usuario desea visitar la página <http://www.netscape.com/home.html> y así se lo indica a Netscape (que corre en la capa de aplicación). Netscape le pedirá a la capa de transporte que cree una conexión de punto a punto con el servidor de Web en la computadora [www.netscape.com](http://www.netscape.com), la que utilizará para solicitar y recibir la página [home.html](http://www.netscape.com/home.html). A su vez, la capa de transporte utilizará los servicios de la capa de red (que se encarga principalmente de enviar y recibir paquetes de información como se verá más adelante). A su vez, la capa de red utilizará los servicios de la capa de enlace, que se encarga de colocar en el cable mediante el método de acceso adecuado la información para que viaje a través de la red. Este proceso se representa en el diagrama de la figura 2.

Cada capa tiene funciones bien especificadas que permiten que el proceso completo se lleve a cabo. A continuación se describen brevemente cada una de ellas.

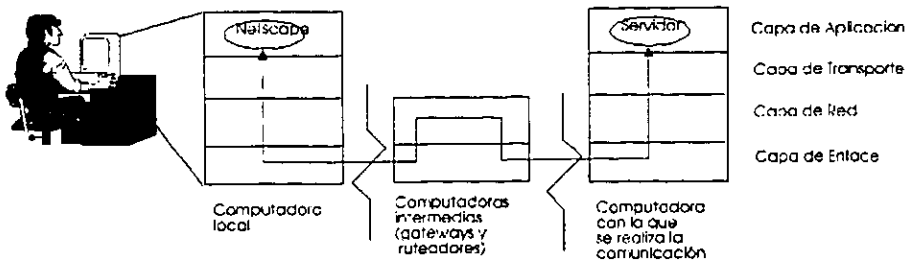


Figura 2 Flujo de información y relación entre las capas.

## Capa de Enlace

La capa de enlace está implementada en el manejador de dispositivos del sistema operativo y en la tarjeta de interfaz que conecta a la computadora con la red. Esta capa tiene a su cargo los detalles de la comunicación en la parte física (*hardware*) así como garantizar la confiabilidad de ésta. La capa de red le entrega a la capa de enlace paquetes de información llamados datagramas. Cada datagrama contiene el número IP (o dirección IP, el cual es un número de 32 bits) de su destinatario. La capa de enlace tiene las funciones principales siguientes:

- Convertir los datagramas en tramas (*frames*). Esto se debe a que las tarjetas de red requieren que la información que éstas envíen esté encapsulada en forma de tramas.
- Convertir el número IP del destinatario en su dirección física. Cuando una computadora desea enviar una trama de una computadora a otra es necesario que conozca la dirección física de la computadora destinatario (cada tarjeta de red tiene una dirección única, a la que llamaremos dirección física); esto se debe a que a ese nivel, las direcciones IP no son significativas. La traducción de número IP a dirección física se realiza mediante el Protocolo de Resolución de Direcciones (*ARP – Address Resolution Protocol*). Mediante ARP se evita que las capas superiores requieran conocer direcciones físicas.
- Enviar la información a esta computadora, utilizando el método de acceso que la red local especifique o los protocolos SLIP (*Serial Line Internet Protocol*, Protocolo Internet de Línea

Serial), CSLIP (*Compressed SLIP*, SLIP Comprimido) o PPP (*Point to Point Protocol*, Protocolo de Punto a Punto) si se trata de una línea telefónica.

- Y convertir de regreso las tramas recibidas en datagramas para entregarlas a la capa de enlace en el lado del receptor.

## Capa de Red

Esta capa es el corazón de Internet. Su función principal es la entrega de paquetes (llamados datagramas) de una computadora fuente a otra destino. Implementa algoritmos para ruteo, para evitar congestionamientos y para interconexión de redes (gateways y ruteadores). Los servicios que provee son no orientados a conexión (*connectionless*). Toda la información que se transmite a través de Internet son datagramas IP. Esta capa no es confiable, es decir, no se encarga de verificar que un datagrama haya sido recibido, o de volverlo a mandar en caso de existir algún error.

El protocolo central de esta capa es el IP y realiza las siguientes tareas:

- Recibe de la capa de transporte la información a enviar (en paquetes llamados segmentos) que incluyen la dirección IP del destinatario.
- Encapsula dichos segmentos en datagramas.
- Determina cuál es la ruta que debe seguirse para entregar cada datagrama. IP sólo es capaz de entregar paquetes a computadoras físicamente conectadas en la misma red local. Así, si se desea enviar un datagrama a otra red, será necesario que IP determine cuál es el ruteador o gateway al que deberá enviarle la información. Una vez determinada la dirección de la siguiente computadora a contactar, le entrega a la capa de enlace el datagrama (que incluye la dirección IP de destino).

- Cuando la computadora recibe un datagrama, verifica si está destinado para ella. Si es así, lo reensambla en segmentos y lo pasa a la capa de transporte. Si no está destinado para ella, realiza nuevamente la operación descrita en el punto anterior.

Otro protocolo importante de esta capa es el Protocolo de Mensajes de Control de Internet (ICMP *Internet Control Message Protocol*) que se encarga de realizar las siguientes funciones:

- Control de flujo. Evita que una computadora envíe más datagramas de los que el receptor puede procesar.
- Detección de errores en las rutas que siguen los datagramas. En ocasiones, algunas rutas no estarán disponibles, y si IP desea comunicarse con una computadora para la que no haya ruta, ICMP se encarga de notificarle el error.
- Verificación de que una computadora esté conectada y su capa de red funcionando correctamente.

### **Capa de Transporte**

La función principal de esta capa es permitir la comunicación directa del remitente a los destinatarios. Consta de dos protocolos: TCP, cuya función principal es permitir comunicación libre de errores orientada a conexión; y UDP (*User Datagram Protocol*, Protocolo de Datagramas de Usuario), cuya función principal es permitir el uso directo de datagramas IP. Las funciones de TCP son, principalmente:

- Dividir la información que recibe de la capa de aplicación en segmentos que pasarán a la capa de red.
- Al enviar un segmento inicializa un reloj, en espera de una contraseña (indicando que el mensaje se recibió); si el reloj expira antes que esta última se reciba, reenvía el segmento suponiendo que el segmento se ha perdido.

- Cuando TCP recibe un mensaje, envía al remitente una contraseña confirmando la recepción.
- Implementa algoritmos para verificar que la información recibida fue la misma que la enviada; en caso de que el segmento llegue dañado a su destino, se indica al remitente del hecho y este último lo reenvía.
- Puesto que IP no garantiza el orden de llegada de los segmentos que envía, TCP debe reordenarlos si es necesario.
- Implementa algoritmos de control de flujo.
- Da la impresión a una aplicación de tener una línea directa en ambos *sentidos* (*full duplex*) a través de la cual se realiza la comunicación.

TCP otorga a la capa de aplicación una comunicación libre de errores punto a punto (de fuente a destino) que aparenta ser orientada a conexión (aun cuando siempre se implemente mediante servicios no orientados a conexión); a esta conexión se le conoce como conexión TCP. Además, TCP define un nivel de direccionamiento, llamado puerto, que permite distinguir entre diferentes conexiones que se estén realizando simultáneamente. Con la capa de aplicación cada puerto es identificado con un número de 16 bits. Su uso es claramente ejemplificado por el modelo cliente/servidor. Para que el cliente pueda conectarse con el servidor, es necesario que el primero sepa dónde encontrar al segundo; para resolver este problema, varios números de puertos están reservados para algunas aplicaciones (correo electrónico, telnet, ftp, web, etc.). Los números de puerto son asignados por IANA (*Internet Assigned Number Authority*, Autoridad Asignadora de Números en Internet). Esta agencia reserva números a los servicios que puede ofrecer un servidor. Por ejemplo, el número de puerto del servicio ftp es el 21, el de telnet es 23, el de Web es el 80. En general, los números de puerto entre 1 y 255 los asigna la IANA. Un cliente de Web sabe que para conectarse con un servidor (también de Web), debe establecer una conexión TCP al puerto 80 de la máquina en cuestión.

UDP (*User Datagram Protocol*) da acceso directo, a los programas de aplicación, al servicio de entrega por datagrama, como el servicio de entrega que proporciona IP. Esto permite a las



aplicaciones intercambiar mensajes en la red con un mínimo de encabezado en el protocolo. UDP no es un protocolo fiable, es un protocolo no orientado a conexión (*connectionless*), esto significa que no hay técnicas en el protocolo para verificar que los datos llegaran correctamente.

¿Pero por qué los programadores ocupan UDP como un servicio de transportes de datos? Hay muchas buenas razones para esto. Si la cantidad de datos a transmitir es poca, el encabezado para crear las conexiones y asegurar la entrega puede ser más grande que el trabajo de volver a retransmitir el conjunto de datos. En este caso, UDP es la elección de protocolo más eficiente en la capa de transporte. Las aplicaciones que usan el modelo “*query-response*” (pregunta-respuesta) son excelentes candidatos para usar UDP, porque la respuesta puede ser usada como un reconocimiento positivo para la pregunta. Si la respuesta no es recibida en un cierto periodo de tiempo, la aplicación sólo enviará otra pregunta.

### Capa de Aplicación

La capa de aplicación, como su nombre lo dice, es donde se encuentran los protocolos de las aplicaciones utilizadas por el usuario. Algunos protocolos de aplicaciones son tan comunes que se decidió estandarizarlos, entre ellas se encuentran acceso remoto (TELNET y RLOGIN), transferencia de archivos (FTP), correo electrónico (SMTP), Web (HTTP), etc. La figura 3 muestra como estos protocolos se relacionan entre sí.

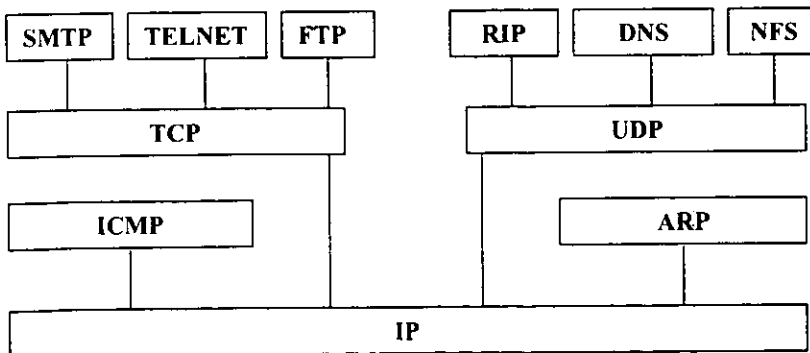


Figura 3 Relación entre los diferentes protocolos de TCP/IP.

## Conclusiones sobre TCP/IP

TCP/IP es único en diferentes maneras pero una de las más importantes características es el hecho de que es un protocolo en capas, las redes TCP/IP fueron diseñadas para ser usadas en redes heterogéneas y puede correr en cualquier tipo de máquina y sobre casi cualquier tipo de topología de red. Ya sea que una compañía use Token Ring, Ethernet o cualquier tipo de topología, TCP/IP puede funcionar.

Desde que las redes TCP/IP están en capas también trabajan para enlazar redes WAN como las líneas dedicadas de 56kbps, o líneas T1. Esta característica permite usar TCP/IP entre sitios corporativos. Así trabaja muy bien tanto para transportar información en una LAN como en una WAN. También trabaja sobre líneas de acceso telefónico y su uso puede fácilmente permitir que usuarios remotos marquen usando un módem. Esta conexión es llamada comúnmente como enlace PPP (*Point to Point Protocol*) o conexión SLIP (*Serial Line IP*) según el protocolo usado, como se muestra en la figura 4, y el *software* existe para casi cualquier computadora que use PPP o SLIP para conectarse a una red LAN.

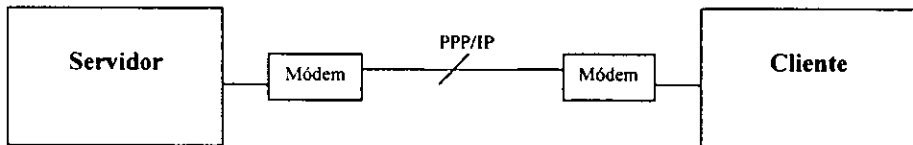


Figura 4. Enlace PPP.

## 1.4 La World Wide Web

La WWW (*World Wide Web*), o simplemente Web, ha despertado grandes expectativas en los usuarios de Internet y el público en general. Esto ha creado una demanda de los niveles de dirección en el personal de sistemas para tener una "página en la Web" lo antes posible y una de las formas de establecer la presencia de su organización en la Web es tener su propio servidor de Web.

### 1.4.1 ¿Cómo funciona WWW?

Por muchas razones históricas, Internet emergió como una fuente rica de información accesible solamente a través de una serie de interfases no muy amigables para los usuarios. Los comandos básicos eran *telnet*, *ftp*, *archie*, *wais* y el correo electrónico que son poderosos pero no intuitivos.

Una Interfase anteriormente usada fue el *gopher*, pero era limitado por las restricciones de su presentación en pantalla; un cliente *gopher* es primordialmente una tabla de contenidos que podía ser leída o copiada por los usuarios.

Después del *gopher* inicio el Web como un proyecto diseñado para distribuir información científica por medio de redes de computadoras en un sistema conocido como hipertexto. La idea fue permitir investigaciones conjuntas entre investigadores y presentar sus notas completas con texto, gráficas, ilustraciones y últimamente sonido y video.

Ideas importantes contenidas en publicaciones pueden ser conectadas por una serie de ligas de hipertexto (hiperligas) y lo más importante es que puede ser mostrada en diferentes plataformas de computadoras. El proyecto *World Wide Web* ha hecho posible la idea de una interfase accesible y atractiva en Internet. Con un navegador de Web, se pueden ver documentos con formato que contienen gráficos e hiperligas a otros documentos. También permite navegar en Internet únicamente moviendo el ratón y dando *click* a alguna hiperliga deseada, a diferencia de *telnet* y otras interfases que es necesario el uso de comandos. Instantáneamente el cliente de *World Wide Web* establece un contacto con la computadora remota y transfiere el archivo requerido a la computadora, desplegándolo en la ventana del navegador como otro documento con formato previamente definido.

Desde su introducción al mundo en 1990 el *World Wide Web* (llamado comúnmente Web) ha crecido de forma rápida y ya son millones las personas que lo utilizan a diario en todo el mundo. Estas personas pertenecen a todo tipo de ocupaciones, profesiones, estilos de vida y creencias.

La capacidad del Web en crear una audiencia mundial por la información que contiene ha sido la causante principal del formidable estallido de popularidad del mismo, al grado que las grandes

corporaciones mundiales e instituciones<sup>6</sup> lo utilizan de forma cotidiana en sus labores. El Web se usa para comunicación, distribución de información o interacción entre miembros de una entidad, así como un vasto y prácticamente inagotable almacén de información de cualquier índole disponible de forma global.

En 1945 el Sr. Vannevar Bush (uno de los principales teóricos de los estudios sobre la información y su organización) describió en un artículo titulado "*As We May Think*" la forma en que el cerebro trabaja con ligas asociativas para la recuperación de la información que contiene y usó este para construir un sistema que, por desgracia, nunca pasó del modelo teórico. Bush llamó a su sistema *memex* (de *memory extension*) y lo propuso como una herramienta que ayudaría a la mente humana a manejar información al hacerse de forma consciente.

En las décadas posteriores al artículo de Bush emergieron más ideas y proyectos de sistemas diseñados como modelos operantes en computadoras para el manejo de información. En 1965, Ted Nelson acuñó el término "hipertexto" para describir texto asociado de manera no lineal ni secuencial a otras ideas o textos.

El hipertexto como lo describió Nelson significa ligar información en una "telaraña" (Web en inglés) de relaciones que hacen posible extenderlo y aumentar el significado o contenido de un documento individual al ligarlo con otros. De igual forma el Sr. Nelson acuñó el término "hipermedia" refiriéndose a todo aquel hipertexto que no está compuesto por texto. Actualmente, el término hipermedia incluye al término multimedia (películas, sonidos, gráficos, etc.).

Nelson también propuso el primer sistema global de hipermedia al que llamó *Xanadu*. Las ideas de Bush y Nelson comenzaron a utilizarse en otros proyectos hacia final de la década de los 80. En marzo de 1989, Tim Berners-Lee, un investigador del Consejo Europeo para la Investigación Nuclear (CERN por sus siglas en francés) propuso un sistema de hipertexto que permitiría compartir información de manera más eficiente entre los miembros de la comunidad de investigadores del área de la física de alta energía. Él llamó a su sistema *HyperText and CERN* del cuál cabe destacar algunos elementos que forman parte hoy de las herramientas que utilizamos:

---

<sup>6</sup> Podemos mencionar como ejemplos a Coca-Cola Co. y a la NASA

- 1) El sistema tendría una cara de presentación al usuario (interfase) que sería independiente de la marca de equipo de cómputo y que permitiría a las personas acceder a la información desde cualquier lugar.
- 2) Un lenguaje esquemático de programación que permitiría manejar varios tipos de documentos y protocolos para vaciarlos en la interfase del usuario.
- 3) El sistema debería de ser de acceso "universal" para permitir el acceso a la información a cualquier usuario de la red.

En marzo de 1991 un prototipo completamente operativo del sistema se encontraba funcionando en una computadora NeXT y usaba una interfase en línea a la que llamaron WWW. En este momento de la historia todas las piezas esenciales para el surgimiento del Web estaban en su sitio aunque no se encontraban disponibles de forma mundial sino sólo de manera local en la red del CERN.

La historia continua de esta forma hasta que en 1993 un joven estudiante de la universidad de Illinois en Urbana-Champaign llamado Marc Andreessen capturó la atención mundial al desarrollar la primera interfase gráfica para el Web a la que llamó Mosaic cuya primera versión salió al público en febrero de 1993. Esta nueva interfase gráfica aumentó la popularidad del Web de forma mundial y según algunas estadísticas el uso del mismo se incrementó del 0.1% de la información que se trasmite en las redes al 1.0% en tan sólo 6 meses. Para septiembre de ese mismo año se liberó la primera versión final de Mosaic conocida como 1.0 y para octubre del mismo año existían 500 servidores de Web en el mundo contra 50 existentes a principios del año.

En 1994 aparecieron varias empresas en el mercado deseosas de participar en el incipiente, pero con gran futuro, negocio que se suponía se convertiría la tecnología del Web. Entre ellas cabe destacar a Spry, Inc. y la formada por Marc Andreessen y algunos colegas que abandonaron el CERN para formar la compañía que hoy se conoce como Netscape. Para mediados de 1994 se formó el *World Wide Web Consortium* o *W3C* que hasta el día de hoy es el órgano internacional que regula y define los estándares que estarán vigentes en el desarrollo del Web.

Con esto se concluye la "fase histórica" del Web sin embargo es difícil predecir los alcances que esta tecnología podrá tener en un futuro próximo aunque se calcula que, al día de hoy existen más de 8.000.000 de servidores Web en el mundo con un flujo de usuarios de muchos más millones de personas.

Por último existe una confusión generalizada al considerar al Web como un equivalente de Internet cuando el Web no es una red de computadoras sino un sistema de aplicación (programa) que puede ser usado sobre Internet o sobre cualquier red o, incluso, sin red alguna.

Una definición aceptada del WEB es: "El WWW es un sistema de comunicación e información basado en hipertexto que se usa normalmente a través de Internet con un esquema de transferencia de datos acorde al modelo cliente/servidor". Los "clientes" Web (conocidos como *browsers* o navegadores) pueden acceder información de hipermedia y de múltiples protocolos usando un esquema de direcciones de red.

WWW utiliza como protocolo de comunicación el HTTP (*HyperText Transfer Protocol*). Inicialmente, HTTP (versión 0.9) era un protocolo para el intercambio de texto, ya fuera llano o marcado en HTML (*Hyper Text Markup Language*). Posteriormente, NCSA (*National Center for Supercomputing Applications*) extendió tanto al navegador Mosaic como al protocolo, para permitir la transmisión y despliegue de gráficas en formato GIF. Esto fue la gran diferencia que llevó a la Web a rebasar la popularidad de Gopher en menos de 20 meses. Una vez que las gráficas fueron incluidas, rápidamente se observó que la Web era útil para la transmisión de prácticamente todo tipo de formatos, muchos de ellos inteligibles para el navegador (o para aplicaciones externas que apoyan a éste, como lo describe HTTP/1.0). HTTP 1.0 es un *Internet-Draft*. Los *Internet-Drafts* son documentos de la IETF (*Internet Engineering Task Force*), y son borradores válidos hasta por un máximo de 6 meses. Se espera que eventualmente pasen de borradores a RFC<sup>7</sup>. Actualmente, es común encontrar en Web imágenes en formato JPEG y GIF, películas (*video-clips*) en formato MPEG, AVI y QUICKTIME, sonido en formato AU, y texto en formato de alta resolución (en formatos POSTSCRIPT, DVI de TeX y Acrobat, además, claro, del popular HTML).

---

<sup>7</sup> Request For Comments (Solicitud para comentarios)

HTTP facilita la distribución de documentos en hipertexto. Si bien WWW es capaz de transferir multitud de formatos, el que utiliza fundamentalmente es HTML, que describe la estructura interna de cada documento, algunas características de su presentación y las ligas de hipertexto a las que apunta. Una liga (o URL, de las que hablaremos más tarde) apunta a la dirección de algún recurso de Internet, ya sea otra página en HTML, un archivo GIF, una página del gopher, un archivo en un sitio de ftp, etc. HTTP describe principalmente, dado un URL, cómo acceder a dicho documento. Este protocolo es sorprendentemente simple dada su versatilidad.

### **1.5 Protocolo de Transferencia de Hyper Texto (HTTP)**

HTTP (*Hyper Text Transfer Protocol*), es el protocolo que permite la comunicación con la World Wide Web. Así como IP, HTTP es también un estándar abierto y cualquier computadora que hable TCP/IP puede hablar HTTP.

HTTP es un protocolo sin conexión (*connectionless protocol*), el cual permite hacer muchas conexiones de manera rápida sin tener que dejar puerto de comunicación abiertos.

Los protocolos sin conexión tienen un defecto en un ambiente comercial; son inestables. Esto quiere decir que los usuarios no pueden ser rastreados a través de un sitio de la red de donde consulten alguna página de información. Para muchas conexiones al mismo sitio, teniendo que abrir y cerrar una nueva conexión TCP para cada página, puede generar una sobrecarga.

La función principal de HTTP es gestionar las comunicaciones entre el cliente y el servidor, permitiendo al primero efectuar consultas y a éste último responder, enviando los documentos solicitados junto con metainformación suficiente, de modo que el cliente pueda interpretarlos correctamente. Además, es deseable que las consultas contengan también metainformación para que el servidor, al interpretarla, pueda satisfacerla en forma más efectiva.

Hasta junio de 1996 no existía una norma real para el HTTP, a pesar de que se ha estado utilizando desde 1990 como protocolo básico del Web. Se describe la versión 1.0 del protocolo en RFC1945, el cual no es un documento normativo, sino meramente informativo.

Afortunadamente, ya se están finalizando las discusiones relativas a versiones posteriores, las cuales constituirían una norma.

### 1.5.1 Descripción del protocolo HTTP

Típicamente, un cliente de WWW, esto es, un navegador de Web (o *browser*) tal como Mosaic, Lynx o Netscape, inicia la transacción con una requisición al servidor, como NCSA Server, WebServer, Apache y otros, en la computadora donde se encuentra dicho recurso. Esta transacción consiste de cuatro pasos:

1. *Establecer la conexión.* El cliente establece una conexión al sistema servidor de Web. Esta conexión opera, por omisión, en el puerto 80 de TCP/IP. El servidor de Web se encuentra en escucha permanente de dicho puerto y en cuanto establece la conexión espera una requisición del cliente.
2. *Requisición.* Un cliente puede efectuar dos tipos básicos de requisiciones:
  - a) *Requisición simple.* Una requisición simple (*simple request*) obtiene un documento en HTML o cualquier otro formato. La línea de requisición tiene la forma: GET <URL> <return>, donde el <return> es un *Carriage Return* junto con un *Line Feed*. El servidor siempre responde a esta requisición con un documento. Si hay algún error, por ejemplo, si el archivo indicado por la URL no existe en el servidor, éste reporta un mensaje de error en inglés en formato HTML que para el cliente es indistinguible de un documento genuino en HTML. Este formato de requisición sencilla se incluye únicamente por razones de compatibilidad con la versión 0.9 de HTTP.
  - b) *Requisición extendida.* Permite el uso de modificadores al comando GET usual. El formato del comando es <Método> <URL> HTTP/1.0 <return> [Modificadores \*] [<return> <datos>]. Entre los modificadores de comando tenemos, por ejemplo, el *if-modified-since*; con este modificador se puede evitar el bajar un archivo que se encuentra en *cache* local si



éste no ha sido modificado. Como lo indica, es posible tener más de un modificador de comando por requisición.

3. *Respuesta.* Como mencionamos en 2.a. las respuestas a requisiciones sencillas son documentos o archivos. La respuesta a una requisición de formato extendido es un mensaje de la forma HTTP/1.0 <número clave> <razón> <return> <??Data??>. El número clave indica si la requisición pudo ser efectuada exitosamente o si hubo algún error y de qué tipo.
4. *Cerrar la conexión.* Como se puede observar, se establece una nueva conexión cada vez que se requiere un nuevo archivo. Por ejemplo, una página personal con texto y tres imágenes genera cuatro conexiones cuando es leída. Las conexiones no son exclusivas, por lo que el servidor puede atender más de una requisición al mismo tiempo. Los URI/URL/URN son la abreviatura de *Uniform Resource Identifier/ Locator/Name*. Las URL y URN son tipos de URI. Todo URI comienza con un nombre de protocolo (tal como http, ftp, nntp, afs, news, mailto, entre otros) seguido por ":" y después un identificador único. En URL's este identificador consiste del nombre de la máquina (por omisión es la máquina local) y el nombre del archivo, programa o recurso a ser accedido. En el futuro, los URN serán nombres asignados a recursos (computadoras, archivos, procesos, dispositivos, etc.) que deben ser transformados a una dirección física por un *Domain Name Server* tal como los nombres de las computadoras en Internet.

Como un ejemplo de cómo funciona el protocolo, tenemos la siguiente sesión de transmisión manual de un documento por la Web, esta sesión tiene como objetivo recuperar el documento <http://daisy.uwaterloo.ca:80/~alopez-o/prueba.html>

```
S telnet daisy.uwaterloo.ca 80
```

```
GET /~alopez-o/prueba.html
```

```
<HTML>
```

```
<HEAD><TITLE>Prueba de Simple-Request</TITLE></HEAD>
```

<BODY>

*Esta es una prueba del formato Simple-request de HTTPD.*

</BODY>

*Connection closed by foreign host*

S

Nótese que el URL no incluye `http://máquina` y así mismo que la conexión es cerrada automáticamente por el servidor justo después de la transmisión del documento. Nuevas versiones de HTTP permitirán establecer conexiones extendidas que se mantendrían abiertas por más de una transacción. El cliente le indicará al servidor con un modificador de comando que mantenga la conexión hasta que le indique que la cierre.

El HTTP utiliza un tipo de encabezado MIME tanto para las consultas como para las respuestas para enviar metainformación entre el cliente y el servidor. Sin duda, la línea más importante en el encabezado de respuesta es la denominada *Content-Type* (Tipo-Contenido) que le indica al cliente qué tipo de información está enviando el servidor: texto HTML, texto plano, imagen GIF, sonido AU, etc. No obstante, para todos los tipos de textos, la interpretación correcta necesita conocer la codificación de los caracteres utilizada, además de la información de tipo básico. El MIME proporciona un parámetro denominado *charset* (abreviación de *character* [carácter] y *set* [conjunto]) para dicho propósito; este parámetro está incluido al tipo en la línea *Content-Type*.

Desgraciadamente, este parámetro ya casi no se utiliza actualmente en el Web, por razones históricas, ya que el Web originalmente utilizaba sólo ISO 8859-1 y este conjunto de caracteres se transformó en su código implícito; incluso se menciona como tal en el RFC1945.

Los creadores europeos y estadounidenses de servidores HTTP no se tomaron la molestia de implementar uno, considerando adecuado ISO 8859-1 para sus propósitos. Hoy en día, ISO 8859-1 ya no es el código implícito, a pesar de lo que se pueda afirmar en RFC1945; de hecho, no existe ninguno, ya que no hay uno estándar, aun cuando se utilizan muchos códigos. Es de

esperar que la futura norma de HTTP reconozca esta condición de facto y haga obligatorio para todos el parámetro *charset*.

La negociación del idioma es una de las facetas de la negociación Tipo-Contenido; utilizando líneas especiales de encabezados, el cliente indica sus capacidades y/o preferencias para que el servidor pueda entonces escoger la versión más apropiada de entre todas las variantes. A pesar de que se desechó en el HTTP 1.0 (RFC1945), la idea resurgió en la norma siguiente del HTTP 1.1. Pero como suele suceder en el Web, las normas resultan atrasadas con respecto a la realidad: ya existen diversos examinadores y servidores capaces de manejar la negociación del contenido en general y la negociación del idioma en particular. Uno de los servidores más populares es el de NCSA que ha sido modificado para permitir la selección del idioma del documento según las preferencias indicadas por el examinador y para identificar también correctamente la codificación de los caracteres. La negociación del idioma es una gran ventaja cuando se navega por localizaciones multilingües. El cliente recibe directamente la versión apropiada sin perder tiempo leyendo páginas difíciles o incompresibles, buscando un hiperenlace para el idioma correcto.

HTTP permite dos formas de comunicación, con lo que permite al *browser* enviar información al servidor y viceversa. La especificación también puede permitir contenido de negociación entre el servidor y el *browser*; aunque esto no está actualmente en uso.

### 1.5.2 Comandos

HTTP soporta varios comandos. Estos incluyen, GET para recuperar páginas, HEAD, el cual trae los encabezados de un documento, POST para enviar información, PUT para poner información en el servidor, DELETE para borrar una página, y TRACE para depurar.

Estos comandos son parte de la especificación de HTTP 1.1. Los comandos más populares son GET, HEAD, y POST. DELETE y TRACE son raramente usados.

### 1.5.3 Encabezados

Las especificaciones de HTTP definen diferentes encabezados que son requeridos para ser enviados. El servidor de Web y el browser usan estos encabezados para pasar información. Algunos servidores permitirán a los desarrolladores, leer el encabezado y usar esta información para tomar decisiones. Algunos de los encabezados incluyen:

- *Accept, Accept-charset, Accept-Encoding, Accept-Language.* Estos son usados por algunos clientes para definir que tipos, conjunto de caracteres, codificación y lenguajes pueden ser utilizados.
- *Authorization.* Usado para decirle al servidor que tipo de autorización está siendo usada. Sólo la básica es soportada.
- *Content-Encoding, Content-Language, Content-Length, Content-Type.* Esto sirve para decirle al cliente que tipo de codificación, lenguaje, longitud, y tipo está siendo usado.
- *Date.* Enviado desde el servidor para decir cuando una respuesta es generada.
- *Expires.* Usado para decir al cliente cuando una página debe ser actualizada en cache.
- *Host.* Para definir que dirección está haciendo la petición del URL.
- *If-Modified-Since.* Usado para recuperar un documento si ha sido cambiado desde cierta fecha.
- *If-Unmodified-Since.* Usado para recuperar un documento si no ha sido modificado desde cierta fecha.
- *Last-Modified.* Es enviado por el servidor para permitirle al cliente conocer cuando el documento fue cambiado por última vez.
- *Location.* Usado para decirle al cliente que el documento ha sido movido.

- *Referer*. Es enviado por el cliente para decirle al servidor el nombre del documento accedido previamente.
- *Server*. Dice que servidor está corriendo
- *User-Agent*. Enviado desde el cliente al servidor para decirle que tipo de cliente es.
- *WWW-Authenticate*. Enviado desde el servidor para decirle al cliente que es requerida su autenticación.

Estos encabezados son los que más frecuentemente se usan. Hay otros que también son utilizados con menor frecuencia y para mayor información de estos es necesario consultar las especificaciones de HTTP.

#### **1.5.4 La Transmisión de la información y el ancho de banda**

Como se ha podido atestiguar por muchos, la Web tiende a ser muy lenta, sobre todo a través de líneas telefónicas. Incluso en archivos pequeños de tipo texto que aparentemente deberían ser de rápido acceso. Cuando estamos bajando imágenes, el cuello de botella es ciertamente la capacidad de transmisión del módem. En cambio, para el acceso de archivos pequeños de texto, el cuello de botella está dado por el establecimiento de la conexión misma, más que la transmisión de la información. Por ejemplo, para establecer la conexión más simple, se requiere enviar un mínimo de siete datagramas. Si encima de esto estamos accediendo a un servidor que intenta identificar el nombre del usuario (varios servidores tienen una opción, que si es seleccionada por el usuario, usa el protocolo de identificación *identp* definido en RFC1413) que solicitó la información (en el cliente), resulta que el tiempo para establecer la conexión, al menos, se duplica. La transmisión de un datagrama de 500 bits en un módem de 9600 bps consume entre 5 y 10 centésimas de segundo. Si incluimos el tiempo de respuesta pueden transcurrir hasta 4 segundos antes de que el archivo en cuestión pueda comenzar a ser enviado.

Si usted requiere una línea para proveer su propia página personal, un módem de 28.8 kbps es suficiente, siempre y cuando su página sea de uso promedio (pocas páginas reciben más de 100 accesos al día). Los proveedores comerciales de Internet requieren por lo menos 64.4 kbps.

### 1.5.5 Servidor HTTP

Un servidor de HTTP es entonces un programa encargado de responder a requisiciones que se efectúan en el puerto 80, según lo define el protocolo HTTP. Dado la URL de una requisición, el servidor debe encontrar el recurso, archivo o programa que corresponde a dicha URL y, si es necesario, ejecuta los programas CGI<sup>8</sup> y SSI<sup>9</sup> asociados a dicha página. Un programa CGI se ejecuta localmente en el servidor y normalmente produce información en HTML o en algún otro de los formatos de la Web; dicha información le es servida al cliente como respuesta a su requisición. Así mismo, dentro de una página de HTML pueden existir comandos de inclusión de información en dicha página. Dichos comandos son ejecutados y la información que producen es incluida en el texto de HTML en el que aparecen. El servidor también se encarga de labores de autenticación y seguridad en los casos en que así sea requerido.

### 1.5.6 Cliente HTTP

Un cliente de HTTP es normalmente un navegador de Web tal como Mosaic o Netscape. Otro tipo común de clientes son el robot de indexado y el *mailbot*. El primero es un robot encargado de visitar páginas para la búsqueda o catalogación de información, mientras que el segundo es un robot que permite el acceso a páginas de Web mediante correo electrónico. Un cliente como Mosaic o Netscape se encarga de desplegar la información en los formatos de uso más común y de llamar aplicaciones auxiliares para el despliegue de información para los formatos que no son tan convencionales, como lo es postscript. Los clientes más avanzados “*cachean*” información localmente para evitar accesos repetidos y además permiten el uso de *proxies*.

---

<sup>8</sup> *Common Gateway Interface*

<sup>9</sup> *Server Side Includes*

## 1.6 Lenguaje de Marcado de Hyper Texto (HTML)

El lenguaje de marcación de hipertexto conocido como HTML (por las siglas en inglés de: *HyperText Markup Language*) es usado para crear documentos para el Web. Concebido como un lenguaje semántico para marcar la estructura lógica de un documento, el lenguaje HTML ofrece a los usuarios una gran variedad de formas para identificar partes estructurales en documentos.

Aprender el lenguaje HTML involucra encontrar etiquetas (*tags*) utilizadas para marcar diferentes secciones del documento y como estas etiquetas son aplicadas a un texto HTML. El lenguaje HTML es definido como un subconjunto del estándar internacional SGML<sup>10</sup> utilizado para la creación de procesadores de palabras y otras aplicaciones que involucran el manejo de textos.

SGML es un meta-lenguaje, esto es, un lenguaje que define lenguajes, cuya finalidad es la de estandarizar la información documental en formato electrónico para una distribución eficiente y coherente entre diversas aplicaciones así como la búsqueda y recuperación de la misma. Un documento creado de acuerdo al estándar SGML no contiene información relevante a la presentación del documento en sí, sino que contiene toda la información que una aplicación como Word o Word Perfect necesitan saber para presentar el documento de forma adecuada, o incluso, para transportarlo de una aplicación a otra.

Las etiquetas en SGML representan e identifican la estructura del documento, como pueden ser, encabezados, párrafos, listas y otros componentes y, como SGML es un meta-lenguaje, los desarrolladores de aplicaciones deben especificar las reglas de la estructura de sus documentos a través de lo que se conoce como "definición de tipo de documento" o DTD (*Document Type Definition*) que especifica exactamente como su aplicación maneja la información del documento y provee una forma para hacerla transportable a otras aplicaciones. Es por el uso tan difundido de SGML que los programas de procesamiento de textos tienen la habilidad de almacenar la información en varios tipos estándar y que pueden ser "vistos" por cualquier otro que utilice dicho estándar.

---

<sup>10</sup> *Standard Generalized Markup Language*. Lenguaje de Marcado Estándar Generalizado.

SGML fue y es usado para definir al lenguaje HTML en todos sus niveles por lo que este se ajusta a su filosofía de datos, estructura y formato al 100%. Los autores de documentos HTML pueden crear los mismos con la acción equivalente a "programarlos" para que el programa de aplicación (en nuestro caso el navegador de Web) los interprete de forma adecuada sin alcanzar nunca la complejidad de los lenguajes de programación formales.

Al usar HTML los desarrolladores de documentos para el Web pueden estar seguros que cualquier navegador presente o futuro podrá "leer" sus documentos y desplegarlos de forma correcta. Con esto se logra no tener que crear una versión de HTML para Windows, una para UNIX y una para Machintosh sino que el mismo sirve para todas las plataformas.

El lenguaje HTML está siendo definido de manera formal por una organización llamada "*HTML Working Group of the Internet Engineering Task Force*" aunque algunos navegadores como Netscape proponen "extensiones" al lenguaje HTML que no son parte de los estándares y que se enfocan principalmente a mejorar la manera de presentar la información que a definir nuevas estructuras.

Aunque existen en el mercado varias aplicaciones que nos permiten manipular el texto de forma dinámica al realizar documentos HTML, éstas, por lo general, se limitan a convertir párrafos y algunos otros elementos como listas y atributos del texto pero muestran deficiencia al manejar gráficos o tablas y no tienen capacidad para manejo de formas. Por esto cualquier persona que desee publicar información en el Web deberá manejar al menos las instrucciones básicas del lenguaje HTML que, afortunadamente, son simples y fáciles de aprender.

Existen varias versiones del lenguaje HTML que van desde la versión 1.0 hasta la versión 4.0. Debido a esta situación muchas páginas Web actualmente especifican con que navegador pueden ser vistas de mejor forma.



## 1.7 Internet

### 1.7.1 Breve historia

Hace 30 años atrás, las agencias encargadas de la seguridad de la Nación Americana confrontaban una preocupación muy genuina: ¿Cómo las autoridades se comunicarían efectivamente luego de un ataque nuclear? Las comunicaciones juegan un papel importante en la seguridad de las naciones. Cualquier autoridad central sería el blanco principal de un ataque.

En 1964 se da a conocer la primera propuesta para dicho problema. En primer lugar la red de comunicaciones sería diseñada desde sus orígenes sin ninguna autoridad central. El principio era sencillo: todos los nodos en la red tendrían igual estatus con la misma capacidad de transmitir, pasar y recibir mensajes. El mensaje por su parte sería dividido en paquetes, cada uno con la información suficiente para llegar a su destino, por lo que el viaje a través de la red sería independiente. La ruta que cada paquete tomase no tendría importancia, siempre y cuando llegase a su destino. A este concepto se le conoce como *packet switching networking*.

La primera red grande y ambiciosa basada en dicho concepto en Estados Unidos fué realizada por la ARPA<sup>11</sup>. Para diciembre de 1969 se encontraban ya conectadas cuatro computadoras, tres en California y una en Utah, en la red que se conoció como ARPANET. Gracias a esta red, científicos e investigadores podían intercambiar información y hacer uso de facilidades de forma remota.

Rápidamente otras facilidades con recursos computacionales comenzaron a hacer uso de esta innovadora tecnología de *packet-switching* para interconectar sus propios sistemas y eventualmente conectarse con ARPANET. En 1971 ya se encontraban alrededor de 20 nodos en la red y en 1972 habían aumentado a 40.

Para este segundo año de operación se había descubierto algo inesperado. La mayoría del tráfico en ARPANET no era precisamente computación a distancia sino noticias y mensajes personales.

---

<sup>11</sup> *Advanced Research Projects Agency*. Agencia de Proyectos e Investigaciones Avanzadas.

Se desarrolla para entonces lo que se conoce como *mailing-lists*, técnica para distribuir mensajes automáticamente a un número grande de "suscriptores".

En los años 70 la red continuó creciendo. Para 1980 había más de 200 nodos, incluyendo la primera conexión internacional (Inglaterra y Noruega - 1973). La estructura descentralizada de la red hacía fácil su expansión. El tipo de computadora que se conectara no era importante; sólo debía ser capaz de "hablar el mismo lenguaje" basado en *packet-switching*.

Originalmente el "lenguaje" utilizado por ARPANET fue NCP (*Network Control Protocol*). Luego fue sustituido por un estándar más sofisticado conocido como TCP/IP. TCP (*Transmission Control Protocol*) es el responsable de convertir el mensaje en paquetes y luego reconstruir este en el destino. IP (*Internet Protocol*) es el que maneja el viaje de los paquetes a través de distintos nodos y redes dada la dirección de su destino. Dado que el *software* que implementaba los protocolos de TCP/IP en las computadoras era de fácil acceso - y sobre todo gratis - unido a la descentralización de la red, no impedía que más y más computadoras se conectasen.

En 1983 ARPANET separa su parte militar en lo que se conoce como MILNET. Otras grandes redes se organizan en los 80's tales como BITNET (*Because it's Time Network*), CSNET (*Computer Science Network*), NSFNET (*National Science Foundation Network*), así como agencias gubernamentales tales como la NASA, *The National Institute of Health* y *The Department of Energy*.

Para finales de 1986 más de 5,000 nodos se encontraban conectados. En 1989 NSFNET reemplaza su *backbone* de 56 Kbps por líneas T1 de 1,544 Mbps. La atención de NSF, así como su gran contribución en el establecimiento de la infraestructura en las comunicaciones, hizo que se le considerara el "*backbone* del Internet". Para este momento más de 100,000 nodos estaban conectados.

Por razones de organización se establece un sistema de "dominios" para distinguir a que grupo o localización pertenece cada nodo. Computadoras internacionales fueron nombradas por su localización geográfica (.us, .uk, .au, etc.). Las demás fueron agrupadas en seis distintos dominios (.edu, .gov, .mil, .com, .org y .net). Los dominios .edu, .gov y .mil pertenecen a instituciones

educativas, gubernamentales y militares respectivamente. Estas fueron las pioneras en ARPANET. Por otra parte el dominio *.com* pertenece a instituciones comerciales, *.org* a organizaciones sin fines de lucro y *.net* a redes que sirven de enlace o apoyo a otras redes.

En 1990 ARPANET deja de existir, pero deja un legado que continua creciendo exponencialmente. Aunque el propósito original de la "red de redes" (Internet) ha ido sufriendo cambios, esta sigue siendo muy popular en la comunidad científica. Posiblemente pueda ser considerado el instrumento más importante de la investigación del siglo XX.

En lo que va de la década la Internet ha alcanzado una atención importante sumado a un crecimiento espectacular. Su acceso se ha movido desde el sector militar y de investigación, a escuelas, sectores públicos y comerciales.

Para el 1992 se crea el *Internet Society* con el propósito de estandarizar protocolos en el Internet y proveer organización a tan creciente movimiento. Ese mismo año NSF crea el InterNIC para proveer servicios de información a la comunidad de la Internet y actualiza su *backbone* a T3 (44.736 Mbps). Para ese momento el número de computadoras conectadas superaba el millón.

El año 1994 fue sumamente importante en la historia del Internet. En este año se eliminan las restricciones comerciales existentes hasta entonces. Es pues que un mayor interés de parte del sector privado y comercial hace que el Internet llame la atención de los medios. Para fines de 1994 había más de 3.8 millones de nodos registrados y más de 30 millones de usuarios estimados. Las oportunidades comerciales, educativas y de acceso a la información para todo tipo de aplicación son realmente numerosas. Esto unido a la descentralizada organización da un sentido de "libertad" donde se eliminan barreras geográficas y sociales.

Las últimas estadísticas demuestran que el crecimiento exponencial continúa y se estima en más de 19 millones el número de nodos en esta gran red de redes.

### 1.7.2 ¿Qué es Internet?

Como se mencionó anteriormente Internet es la red de redes. Internet constituye una fuente de recursos de información y conocimiento compartidos a escala mundial. Es también la vía de comunicación que permite establecer la cooperación y colaboración entre un gran número de comunidades y grupos de interés por temas específicos, distribuidos por todo el planeta.

En Internet es posible encontrar toda clase de *software* para una gran variedad de computadoras y sistemas operativos. De modo sencillo se puede establecer una conexión con alguna de las miles de computadoras dedicadas a proveer, de forma gratuita, los archivos que poseen. Así pueden copiarse programas de uso público, de *shareware* y aplicaciones comerciales para evaluación, incluidos juegos de computadora. Los fabricantes de *hardware* suelen tener servidores donde es posible obtener actualizaciones de los controladores (*drivers*) de sus productos.

A través de la Internet pueden consultarse los catálogos de las bibliotecas más importantes del mundo, acceder a bases de datos con los temas más diversos y transferir copias de los documentos encontrados.

Es posible visualizar y copiar archivos de imágenes con fotografías de todo tipo o reproducciones de cuadros. Pueden hacerse cosas como conversar en tiempo real. Dos o más personas, separadas por miles de kilómetros de distancia, pueden comunicarse a través de la Internet escribiendo en la computadora.

No solamente es posible obtener información o utilizar algún tipo de servicio. El usuario también puede ofrecerlos si lo desea. Una de las formas más sencillas es participar en un grupo de noticias o en una lista de correos. Los artículos que allí se envíen serán distribuidos automáticamente entre todos los miembros de la lista, y éstos pueden ser miles repartidos por todo el mundo.

La lista de cosas que pueden hacerse o conseguirse a través de Internet sería interminable. Lo que se busque o se encuentre depende en gran medida de los intereses particulares y de la ocupación de cada uno, resultando imposible dar una imagen de ello en unas cuantas líneas. Lo mejor es que

cada uno explore por sí mismo y descubra lo apasionante que puede resultar un viaje a través de la autopista de la información, un viaje por la Internet.

### 1.7.3 Principales servicios de Internet

#### Correo electrónico

El correo electrónico es la aplicación más utilizada de Internet. Cada usuario tiene una dirección electrónica, que le permite comunicarse por escrito de manera casi instantánea con otros usuarios de la red situados en cualquier punto del planeta. Si la computadora no está prendida, el mensaje queda almacenado en un servidor.

El correo electrónico, más conocido como *e-mail*, es la aplicación más clásica de Internet. *El e-mail* permite la toma de decisiones y el desarrollo de proyectos comunes aunque las personas estén alejadas físicamente.

Para enviar un mensaje electrónico sólo basta estar conectado a Internet y tener un programa de correo electrónico, que se puede conseguir gratuitamente en la red. Igualmente, será necesario que la persona a quien se envía, esté conectada y que se conozca su dirección electrónica.

Como en el correo convencional, todos tenemos una dirección electrónica personal donde recibir la correspondencia y además también se tiene la opción de inscribirse a periódicos y revistas electrónicas.

El correo electrónico también se utiliza para transferir todo tipo de archivos que pueden contener bases de datos, gráficos o *software* comprimido que es posible enviar de forma sencilla sólo conociendo la dirección del destinatario.

Ventajas más importantes respecto al correo postal y al teléfono:

- Más velocidad (un mensaje puede llegar a cualquier punto del mundo en horas, minutos o incluso segundos)

- Costo más económico
- Se pueden transferir archivos y programas
- El destinatario puede responder cuando le interesa

Cada usuario de Internet está identificado con una dirección electrónica que consta de:

El nombre del usuario seguido del signo @ (arroba) que sirve para especificar donde se encuentra el usuario. Después se le añade el nombre del servidor, el lugar donde se encuentra la máquina (empresa o institución) y el país (que en el caso de México es "mx"). Una dirección típica de Internet podría ser ésta: paco@ine.gob.mx.

### **Listas y News**

Auténticas pláticas internacionales a través de la red. En estos foros de debate se puede platicar electrónicamente con usuarios de todo el mundo sobre cualquier tema. Existen más de 10,000 foros dedicados a los temas más variados.

Una lista de correo es una dirección electrónica a la que puede subscribirse cualquiera que disponga de correo electrónico. Cualquiera de los subscriptores de la lista puede enviar mensajes a dicha dirección, y cualquiera de los mensajes que recibe la lista es automáticamente reenviado al resto de los subscriptores de la misma.

Cualquier materia que pueda proponer, puede tener respuesta en una lista electrónica. Existen diferentes tipos de listas, algunas de las cuales sólo proporcionan información y otras son de libre acceso. A las listas de información sólo envían mensajes a los suscriptores una persona o un grupo reducido. A las de libre acceso, en cambio, todos los suscriptores pueden enviar correos a los miembros de la lista.

Otros servidores mantienen direcciones de correo de listas de distribución, es decir, cuando un usuario envía un mensaje a la lista, el servidor envía una copia a cada uno de los usuarios suscritos a ella. Para recibir información sobre los servidores de listas basta que conozcamos la

dirección electrónica de uno de ellos, ya que ese nos puede facilitar información de cómo podemos utilizarlo y la dirección de otros servidores.

Los grupos de noticias o *News* es el sistema más sencillo para localizar colectivos y personas relacionadas con nuestros intereses. Son grupos de debate o trabajo en grupo organizados temáticamente que tratan cualquier tema que imaginemos. Es algo similar a las listas, pero con la diferencia de que los mensajes no son depositados en un buzón de correo, sino en una lista pública que se puede mirar cuando se desee.

Mediante el programa correspondiente, podemos expresarnos libremente sobre cuestiones relacionadas con el grupo. También hay grupos donde el moderador filtra la información. Existen entre 4000 y 5000 grupos temáticos, algunos de los cuales son mantenidos por empresas privadas a las cuales se les paga una cuota para recibir estas noticias.

### **Transferencia de archivos**

FTP (*File Transfer Protocol*) es una herramienta que permite copiar desde una computadora archivos procedentes de cualquier usuario o servidor del mundo. La oferta de los servidores de FTP es muy variada. Vía Internet, puede incorporarse a la PC *software* gratuito, documentos técnicos, bibliotecas gráficas, aplicaciones informáticas, etc.

El término FTP se refiere a la aplicación diseñada para la transferencia de archivos entre dos computadoras. Estos archivos pueden ser documentos, textos, imágenes, sonidos, programas, etc.

### **Telnet o conexión remota**

Telnet es un programa que nos permite la conexión remota con cualquier servidor conectado a Internet. Para acceder a los sistemas remotos hay que estar autorizado como usuario. Una vez que estamos dentro, podemos actuar como usuarios locales.

Existen multitud de servicios accesibles vía telnet, la mayoría de ellos de carácter público, donde cualquier usuario puede acceder a la información que se ofrece.

## **Archie**

Archie es un sistema de información diseñado para la localización de archivos en Internet. Archie realiza búsquedas en una base de datos que contiene información actualizada de los archivos almacenados en la mayoría de los lugares de FTP anónimo.

Su arquitectura cliente/servidor permite localizar los archivos que busca entre los millones existentes en la red. Para conectarse a un servidor de archie se tiene que utilizar el protocolo telnet. Los servidores de archie ofrecen información sobre los archivos disponibles en los diferentes servidores de FTP de todo el mundo.

Fundamentalmente, hay dos formas distintas de usar archie: Estableciendo una sesión telnet con alguno de los servidores disponibles o usar un programa cliente para archie.

## **Gopher**

Este servicio facilita la búsqueda de información dentro de Internet a través de menús jerárquicos, evitando que nos podamos perder en la red. Cuando se accede a un servidor gopher, se ve una serie de directorios temáticos con títulos indicativos que le facilitan donde dirigirse para encontrar la información que se desea localizar.

## **World Wide Web**

El World Wide Web (WWW) es el servicio de moda en Internet. Como se mencionó en la sección 1.4 de este capítulo, es un sistema de información muy ágil que utiliza el formato hipertexto: con un simple "*click*" del ratón usted escoge las opciones que más le interesan de entre un amplio menú de imágenes, textos y sonidos. Podrá leer periódicos a través de la computadora, visitar museos, entrar en la Casa Blanca, consultar catálogos de empresas, etc.

## **WAIS**

WAIS (*Wide Area Information Servers*). Es una interfase única para acceder a las bases de datos y bibliotecas. A partir de una palabra o frase en lenguaje natural, busca los documentos que hacen



referencia a estos temas, dentro de un conjunto de bases de datos dispersas por todo el mundo. Este servicio es accesible conectándose vía telnet a un servidor que lo proporcione

### **Talk, IRC**

Se puede conversar con cualquier usuario que tenga acceso directo a Internet, de forma individual o colectiva.

*Talk* una herramienta de UNIX que solicita al usuario especificado una conversación en línea a través de una ventana del servidor. La pantalla de cada uno de los interlocutores se divide horizontalmente en dos mitades. En la mitad superior aparece el texto que escribe un usuario, y en la inferior el texto que escribe el segundo usuario. Todo esto ocurre al mismo tiempo que escribimos en el teclado, y aunque los dos escriban al mismo tiempo el texto aparece donde le corresponde.

El IRC (*Internet Relay Chat*) permite mantener gran número de conversaciones, cada una de ellas con un número indefinido de usuarios, situados en cualquier lugar del mundo.

Cuando nos conectamos con un servidor que dispone del servicio de IRC, podemos “sintonizar” uno de los canales que en ese momento estén activos o abrir uno nuevo. Todo lo que envía cualquiera de los participantes en la conversación es automáticamente “retransmitido” al resto de los participantes del mismo canal. La conexión a un sistema IRC puede hacerse con clientes específicos o con telnet.

## CAPÍTULO II

### ASPECTOS DISTINTIVOS DE LAS INTRANETS

---

#### 2.1 Antecedentes

Desde 1995 el modelo World Wide Web (WWW) ha experimentado una significativa evolución en los ámbitos corporativos al considerarlo como una herramienta que puede mejorar notablemente la productividad. Ha surgido así el concepto "Intranet", es decir las Internets internas de las empresas, lo que está suponiendo una autentica revolución en el manejo de la información corporativa. La Intranet se deriva de la Internet actual como un paso natural en su propia evolución. Utiliza los mismos protocolos y aplicaciones TCP/IP, y en particular el modelo cliente/servidor del WWW.

El crecimiento de la Internet en los últimos años ha sido explosivo, en efecto, el crecimiento ha sido tan rápido que el número de usuarios es desconocido. Mientras los medios de comunicación han considerado a la Internet como el mayor invento en el ámbito de la computación, la Intranet, Internet interna compuesta principalmente por servidores de WWW y listas de correo, son un gran éxito en muchas compañías. Por eso podemos decir que el gran impacto que está teniendo la Internet como forma universal de acceso a la información, está penetrando también dentro de las corporaciones.

Como se dijo en el capítulo anterior, al hablar de Internet nos estamos refiriendo a eso que llaman la red de redes, que es un enlace multinacional y anárquico en el que es posible incluir a millones de redes de computadoras sin importar a qué distancia se encuentren. Intranet, al igual que Internet, es un concepto global de muchas propiedades y no un objeto tangible. Necesita y utiliza una gran variedad de productos de *hardware* y *software*, aún cuando no existan paquetes realmente imprescindibles. Es además una especie de Web de uso privado para una empresa que puede utilizar muchas de las herramientas creadas para Internet sin ninguna alteración, y que

están mantenidas sobre redes internas para que la información manejada sea vista tan sólo por los empleados de la empresa.

La Intranet es la visión de futuro de cómo las empresas pueden usar las tecnologías normativas de Internet para implantar un entorno omnipresente, completo y dotado de todas las funciones necesarias para poder compartir información, comunicaciones y aplicaciones. Un entorno creado sobre tecnologías abiertas de conexión de red y sobre una plataforma abierta de aplicaciones de red.

## 2.2 El concepto de Intranet

El término Intranet se usa ampliamente para describir las tecnologías Internet en redes corporativas internas. En general, las empresas usan actualmente una Intranet para publicar y compartir información en forma más efectiva.

La Intranet es un concepto estudiado inicialmente por *Forrester Research* en un informe del mismo nombre con fecha del 1 de marzo de 1996. Expresado en términos sencillos, una Intranet es la red TCP/IP de una empresa, que enlaza a sus empleados y su información de tal manera que: aumenta la productividad de aquellos, facilita el acceso a la información, y convierte la navegación por los recursos y aplicaciones de su entorno informático en un proceso más infalible de lo que era.

La Intranet saca partido de la serie de protocolos y normas abiertas que han surgido a partir de Internet. Dichas normas hacen posible la existencia de aplicaciones y servicios tales como el correo electrónico, el *software* de trabajo en grupo o *groupware*, la seguridad, los directorios, la compartición de información, el acceso a bases de datos y la administración, que son tan avanzados, y en muchos casos más, como los sistemas propietarios tradicionales, por ejemplo, Lotus Notes o Microsoft BackOffice. Dado que la Intranet se basa en estas normas abiertas, los clientes recogen los beneficios de la compatibilidad con bases de datos y plataformas universales, de la flexibilidad y de la independencia del proveedor. Así mismo, adquieren la capacidad de

aprovechar las innovaciones y los productos creados por todo un sector, no sólo por un proveedor.

Entonces una red privada dotada con la tecnología de Internet se convierte en una Intranet que consiste en un equipo de conmutación y de comunicaciones que es propiedad de una organización interconectada mediante líneas de comunicación alquiladas o propias.

### **2.3 Razones para la implantación de Intranets**

Desde el punto de vista operativo la Intranet permite a las organizaciones compartir eficazmente una gran cantidad de recursos de información que fluye de forma rápida y transparente desde las fuentes de origen hasta los puestos de trabajo de los empleados, todo ello con el mínimo costo, tiempo y esfuerzo.

Imaginemos una empresa con varias oficinas distribuidas geográficamente; los empleados necesitan conocer puntualmente los cambios en las normas y directrices de la compañía, o bien consultar información sobre especificaciones de productos, precios; y otras más simples, aunque importantes, tales como el listado telefónico. Hasta ahora lo normal era el uso de materiales impresos del tipo guías, listas de precios, manuales de ventas, etc. La producción, distribución y mantenimiento de estos materiales resulta, por un lado, costosa y lenta, y por otro, poco eficaz a la hora de su necesaria actualización. Con los procedimientos tradicionales, basados en papel, es imposible asegurar que cada individuo vaya a tener la versión correcta y actualizada de la información que necesita, muchas de las cuales quedan obsoletas incluso antes de que se reciba el material impreso. Hay que tener en cuenta que los cambios organizativos son cada vez más frecuentes en el mundo actual de los negocios, cada vez más dinámico y competitivo. Y para ello es preciso que la comunicación y el flujo de información sean lo más exacto y puntual posible.

El concepto Intranet surge así para dar respuesta a esos problemas, ya que es una solución tecnológica que: permite la entrega de información bajo demanda y en el momento en que se requiere; garantiza la precisión y actualización de la información; asegura que la información se

almacena en una única fuente (aunque no es preciso que esa sea la única fuente de toda la información); y permite que la información sea mantenida por los propios departamentos o personas que la generaron.

Actualmente es posible resolver esos problemas con el uso de las tecnologías genéricas surgidas en la Internet, lo que supone un cambio drástico, una auténtica revolución en el uso de las tecnologías de la información en el mundo empresarial.

Una Intranet contribuye a reducir tanto el tiempo como los costos de distribución de la información. En lugar de hacer copias y repartirlas por correo ordinario, documentos tales como boletines, actas, normativas, proyectos, agendas comunes, etc., se pueden centralizar en páginas Web internas y ser compartidas por todos los interesados, que las consultarán en el momento más oportuno, siempre y cuando tengan acceso a la Intranet. Por ejemplo, en el servidor Web interno se pueden poner hiperenlaces a informes y publicaciones en línea de interés para los empleados; también a información pública sobre empresas de la competencia. De este modo se tiene acceso puntual y actualizado a los temas de mayor relevancia.

En resumen, las Intranets se están extendiendo hoy entre empresas de diversos campos (*software*, editoriales, finanzas, seguros, salud, etc.) porque permiten a las organizaciones: reducir costos y ahorrar tiempo, centralizar la información, compartir la información interna, y organizar la información con páginas Web.

## **2.4 Ventajas que ofrecen las Intranets**

La principal ventaja es que, al utilizar la estructura de Internet como red WAN, se ahorra el gasto de tener que instalar todo el cableado, pagar las comunicaciones, los equipos, etc. para crear la red. Los usuarios remotos, estén donde estén, pueden comunicarse con la red corporativa a través de una llamada local al proveedor de acceso a Internet. Se pueden conectar computadoras de distintas plataformas de *hardware* y de distintos sistemas operativos. La información se puede estructurar en forma de ligas de hipertextos para facilitar la búsqueda de información. La

interfase de uso es muy amigable y fácil de aprender, por lo que se ahorran muchas horas de capacitación a los empleados. Se puede incorporar multimedia (audio, vídeo, aplicaciones interactivas, etc.). La inversión inicial es muy reducida si ya se posee una estructura de red y es muy escalable, se puede empezar con lo más necesario e ir ampliándola de acuerdo a las necesidades. Al utilizar estructuras de computación distribuida se utilizan los recursos más eficientemente. Y por si esto fuera poco existe una gran cantidad de *software* gratuito y de dominio público para implementar esta tecnología.

Podemos resumir las ventajas en los siguientes puntos:

- Sistema abierto (soporta múltiples plataformas). Es una tecnología abierta que no depende de un sólo fabricante ya que utiliza el conjunto de protocolos TCP/IP.
- Tiempo de implantación pequeño y un desarrollo de las aplicaciones con acceso a bases de datos a un costo reducido, en relación con los beneficios inmediatos que se obtienen.
- Precisan de un escaso personal de mantenimiento y mínimos gastos de integración con otras soluciones de red.
- Supone un método sencillo y efectivo de comunicar a los empleados con los centros de información y los focos de toma de decisiones.
- Se agiliza el acceso de los clientes, proveedores, canales de distribución, etc. a toda la información que cada empresa quiera facilitar para hacer más fácil la compra, y gestión de productos y servicios.
- La calidad de la información que se genera se incrementa de forma notable, ya que, entre otras cosas, procede de una sola fuente, se entrega cuando se necesita y es mantenida por quien la genera por lo que está actualizada.
- No hay que cambiar el sistema de información existente en la empresa, la infraestructura es utilizable.

- Plazo de retorno de la inversión mínimo (de instalaciones investigadas en E.U., la más baja obtuvo un retorno de 11.3 %, plazo de amortización nueve semanas)<sup>12</sup>
- La tecnología que se emplea es segura, conocida y probada, todos los adelantos que aparezcan en la red Internet serán inmediatamente aplicables a las Intranets.
- Se puede aplicar multimedia (audio, vídeo, realidad virtual, etc.)

Y decir que las Intranets están creciendo porque ofrecen:

- Costo/eficiencia.
- Facilidad de actualización.
- Utilidad para liberar información.
- Utilidad para manejar información disponible en demanda.
- Seguridad.
- Facilidad relativa de configuración, uso y coordinación
- Aptas para correr en toda la compañía, así como por departamento o por área funcional.
- Capacidad de crear aplicaciones en multimedia.
- Reducción de los costos de impresión y tiempo de distribución.
- Capacidad de correr a través de cualquier plataforma y para integrar a todos los clientes.
- Un puente hacia Internet.

---

<sup>12</sup> Estudios de casos Intranet. Septiembre de 1998. [http://www25.netscape.com/es/comprod/at\\_work/customer\\_profiles/](http://www25.netscape.com/es/comprod/at_work/customer_profiles/)

- Aptas para integrarse con bases de datos internas de corporativos
- Utilidad para desarrollar contenido por medio de cualquier usuario, en lugar de que lo realicen técnicos profesionales
- Accesibilidad al usuario al estar basada en la tecnología de Internet.

## 2.5 Características de las Intranets

Una de las características principales de la Internet es la capacidad para comunicar entre sí, mediante la serie de protocolos TCP/IP, sistemas informáticos heterogéneos, equipos de diferentes fabricantes que están conectados a una o varias de las miles de redes que forman hoy esta red de redes mundial. Pues bien, esa misma situación se reproduce a menor escala en una empresa, donde existe un ordinario y variado ambiente informático, tanto en *hardware* como en *software*, diversidad que se extiende así mismo a los individuos que utilizan los equipos y a sus funciones.

En una empresa el reto del Departamento de Informática consiste en desarrollar plataformas comunes para toda la organización que contribuyan a mejorar la productividad. Precisamente es aquí donde incide de lleno el concepto de Intranet: se trata de crear una plataforma flexible, portable y potente que sirva a todos los empleados de la organización para todas las funciones que realizan, desde buscar y archivar documentos e informes hasta pedir material de oficina, incluso para el control de horario de asistencia. Todo esto es posible hacerlo mediante Web's internos convenientemente diseñados y adaptados a las necesidades específicas de la empresa.

Se puede decir que el Web es el corazón de las Intranets, aunque en este caso con los controles de acceso necesarios para limitar su uso internamente. Pero no es el único elemento; hay otros derivados de las aplicaciones de la Internet y que comparten la misma tecnología, aunque con aspectos específicos para los entornos corporativos. En concreto, la serie de protocolos TCP/IP y las aplicaciones asociadas juegan aquí un papel relevante. Así, por ejemplo, es cada vez más



frecuente encontrar en redes de área local (LAN) corporativas, el protocolo IP de la Internet; bien por haber desplazado a otros privados (IPX, AppleTalk, SNA, etc.) o bien por coexistir con ellos. Esta elección técnica del protocolo IP como base de la Internet ha sido uno de los factores de éxito de esta red de redes sin costuras visibles. Y sobre TCP/IP tenemos todas las aplicaciones bien conocidas de *telnet*, *ftp*, *sftp*, etc., inicialmente desarrolladas para sistemas UNIX, pero que hoy funcionan sobre otras muchas plataformas (DOS, MS-Windows, Apple Macintosh, etc.), y otras aplicaciones nuevas relacionadas con el acceso distribuido a fuentes de información multimedia (que funcionan normalmente en modo cliente/servidor), de las que el máximo exponente actual es el WWW, basado en el protocolo HTTP y en el lenguaje HTML, ambos abiertos.

Pues bien, en la Intranet tenemos esos mismos protocolos y aplicaciones pero en ámbitos locales de LAN, lo que de entrada supone una mayor eficacia y un mejor tiempo de respuesta que en la Internet global, así como un mejor aprovechamiento del ancho de banda. Los insoportables tiempos de espera (frecuentes en la Internet para que aparezca en la pantalla una página Web, rica en imágenes, de un servidor remoto), en la Intranet desaparecen, lo que supone un aspecto muy positivo para la productividad de la empresa. Mediante la interfaz Web, además de acceder a documentación multimedia diversa, se pueden transferir archivos, consultar bases de datos, enviar correo electrónico, acceder a boletines y foros de discusión, rellenar formularios, etc.

En los casos en que la empresa tiene varias sucursales distribuidas geográficamente, las alternativas que se presentan para conectar las LAN son: líneas dedicadas privadas, redes públicas convencionales y la Internet. La primera de ellas es la más directa y segura, pero tiene un costo elevado, afrontable únicamente por grandes multinacionales, y se circunscribe al mundo interno de la corporación. La segunda ha venido siendo utilizada normalmente por empresas más reducidas a las que no les resulta rentable montar redes corporativas privadas. En esta segunda opción se contemplan las redes públicas X.25 y la interconexión de LAN por *Frame Relay*; las primeras son poco eficaces para los servicios basados en TCP/IP y las segundas utilizan la tecnología de comunicaciones de la Internet, pero sin abrirse a ese mundo.

La opción de comunicación utilizada para montar una Intranet corporativa es precisamente la Internet, es decir unir las redes locales de las distintas sucursales mediante uno o varios proveedores de servicio Internet, creando túneles IP privados dentro de la infraestructura global IP e instalando medidas de seguridad, tales como "paredes de fuego" (*firewalls*), entre el Web interno y el World Wide Web, lo que evita la intrusión desde otros dominios externos no autorizados. De este modo, cualquier usuario de una Intranet puede acceder a cualquier Web interno al que tenga autorización, tanto en el ámbito nacional como en el internacional, por la propia naturaleza global de la Internet.

Por último, interesa destacar una característica distintiva de las Intranets: la gestión. Si bien la Intranet usa los mismos protocolos y lenguajes que la Internet, esto no implica que deba seguir las mismas normas "caóticas" de funcionamiento. De hecho, este aspecto, junto al de la seguridad, es de los más importantes para las empresas, que deben velar por crear sistemas informáticos que aumenten la productividad. Por ello habrá que configurar los privilegios de acceso en función de la actividad de cada departamento, evitando la posibilidad de navegar libremente por servidores ajenos con información poco relevante para la empresa o de simple ocio. Así mismo habrá que crear espacio cache y servidores *proxy* para mejorar la eficacia en el acceso a servidores externos más frecuentados. También hay que considerar la opción de instalar aplicaciones sobre estadísticas de uso, que pueden ayudar tanto para control interno como para el ajuste fino de determinados parámetros en las distintas aplicaciones.

En resumen, podemos decir que las características principales de las Intranets son:

- a) El uso interno de los protocolos TCP/IP en las LAN corporativas.
- b) El uso de un *browser* como interfaz de acceso común para todas las aplicaciones corporativas sobre un conjunto heterogéneo de equipos y sistemas.
- c) La utilización de proveedores de servicio Internet para la comunicación entre LAN corporativas en ámbitos geográficos dispersos.

- d) Los procedimientos de gestión de los sistemas y el control de acceso de los usuarios, que deben configurarse a la medida de las necesidades y actividades de la empresa.

## 2.6 Diferencia con las soluciones para trabajo en grupo

Frente a las soluciones ya existentes para crear un entorno de trabajo en grupo, que permita producir, difundir y compartir información dentro de una organización, la tecnología Intranet se convierte en un modelo alternativo con unas características propias, que la diferencian de los productos comerciales desarrollados por empresas líderes de este mercado como Lotus, Microsoft y Novell.

- a) Intranet utiliza una tecnología basada en protocolos abiertos, sin soluciones propietarias o particulares, y disponibles para su uso en plataformas muy diversas tales como Microsoft Windows, Apple Macintosh, UNIX, etc.
- b) Intranet es una solución graduable que necesita una inversión inicial baja y es capaz de crecer conforme lo hagan los medios disponibles en la empresa y los servicios requeridos. Es, además, una solución tanto para empresas con un solo centro de trabajo como para aquellas que se encuentran diseminadas en varios.
- c) Implantar la tecnología Intranet requiere un menor costo que los modelos comerciales tradicionales. En el mercado existen productos de uso libre (*freeware*) que cubren todas las necesidades: *software* para servidores, para clientes, para creación de contenidos, etc.
- d) Una solución Intranet permite que los contenidos sean gestionados directamente por sus creadores, en vez de por técnicos informáticos, lo que supone mejorar la productividad y reducir los costos de desarrollo.
- e) Los datos en la Intranet se encuentran almacenados siempre en una única fuente, sin necesidad de que estén replicados en varios servidores, como es el caso de Lotus Notes, por comparar con uno de los productos de mayor difusión en esta área.

- f) Como valor añadido, un desarrollo Intranet se convierte, sin necesidad de costo adicional, en una plataforma válida para proporcionar a la empresa conexión hacia el mundo exterior y el uso de sus aplicaciones en la Internet.

Todo ello ha hecho reflexionar a las casas comerciales y que apuesten por soluciones compatibles Intranet/Internet. Se está produciendo, por tanto, una convergencia de los productos comerciales para trabajo en grupo hacia la tecnología Intranet: servidores que usan el protocolo HTTP, editores HTML integrados en sus productos para edición, compatibilidad de la mensajería electrónica, etc.

## 2.7 Tecnologías Web y componentes básicos

### 2.7.1 Tecnologías Web

El World Wide Web, en forma abreviada WWW o simplemente Web, como se suele denominar en muchos casos, está distribuido en una red que constituyen una telaraña mundial de Internet, en la que el acceso a la información y a los recursos se realiza a través de los documentos de hipertextos, que combinan en su estructura diversos recursos multimedia.

Para establecer conexión con servidores Web, y obtener la información y los servicios que estos prestan, es necesario, como es habitual en Internet, disponer de un programa cliente "browser" o llamados también paginadores o navegadores.

Además la sencillez en el uso, otro de los factores que han influido en la popularidad de WWW, es la presentación atractiva de la información que se consigue mediante las técnicas de hipermedia.

Cuando se accede a un servidor Web lo primero que aparece es una página Web o documento de hipertexto. El aspecto de este documento dependerá en gran medida del *hardware* que se disponga y del *software* instalado. Estos están formados por una combinación de texto, imágenes, y un nuevo tipo de elementos llamados hiperenlaces.

Los hiperenlaces se presentan, usualmente, en la forma de palabras o frases que están resaltadas, mostrándose en distinto color o subrayadas, estas constituyen enlaces a otros documentos que pueden estar dentro del mismo servidor o en cualquier otro ubicado en cualquier parte del mundo y con un solo *click* del *mouse* sobre el enlace automáticamente nos movilizamos hacia la página Web, que este enlace hace referencia.

### 2.7.2 Componentes básicos

Los componentes básicos para el WWW son:

- El World Wide Web, para establecer la conexión entre un usuario y un servidor crea un nuevo protocolo de hipermedia, el HTTP (*Hyper Text Transfer Protocol*).
- Se pueden acceder desde el navegador, a los servicios Intranet o Internet tales como servidores WWW, FTP y otros.
- En el WWW el usuario puede recibir información, en muchos casos puede enviarla al servidor con el que está conectado, mediante formas dentro de una página Web.
- Los enlaces de hipertexto no sólo se nos presentan en texto, sino que puede ser una imagen la que nos transporte con un sólo *click* a otra página Web ubicada dentro de la red.
- La estructura de una página Web, se da en un nuevo lenguaje llamado HTML (*Hyper Text Markup Language*). Este lenguaje permite dar indicaciones precisas, al programa cliente, de cómo debe presentarse el documento en pantalla o al imprimirse.
- Los motivos básicos ideales para una Intranet son los siguientes: desde el Navegador hasta cada servidor, deben administrarse de manera centralizada. Sencillamente puede usar una herramienta de administración para configurar y gestionar miles de navegadores, desde cualquier lugar de la red. Puede administrar el entorno de servidores mediante una interfaz

basada en HTML, incluso desde un sitio remoto. Además, los administradores pueden gestionar la distribución y el mantenimiento del *software* desde lugares remotos.

Los programadores de aplicaciones pueden sacar partido de los servicios y funciones del marco de aplicaciones de Internet, mientras crean sus aplicaciones. Miles de programadores de *software* comercial de otros fabricantes están creando productos de *software* que se ejecutan en la plataforma Internet e Intranet.

La Intranet facilita la búsqueda de cualquier información o recurso que se encuentre en la red. Los usuarios, al ejecutar una sola consulta, obtienen una lista organizada de toda la información coincidente en todos los servidores de toda la empresa y en Internet. Los índices y jerarquías de exploración se crean y mantienen con la mayor facilidad. Con servicios de agentes, los usuarios pueden hacer que los servidores estén atentos a la llegada de nueva información o vigilen si se producen cambios en los recursos existentes. El control de acceso sólo permite que accedan los usuarios autorizados a los documentos restringidos. En consecuencia, todos los usuarios tienen un acceso completo y personalizado a toda la información importante de su red interna y de Internet.

Por ejemplo, un ejecutivo podría efectuar una sola consulta cuyo resultado fuera toda la información interna y externa referente a un producto en particular, con informes internos del equipo de producción, planes de mercado, información de la competencia y artículos de prensa, todo ello presentado de una forma organizada.

## **2.8 Aplicaciones de las Intranet**

La utilización de la tecnología de Internet, y en especial la tecnología World Wide Web, para crear una Intranet dentro de una organización, está provocando profundos cambios en la cultura corporativa de las empresas, que se plasman en la significativa evolución de los modelos actuales de flujo de la información interna y de trabajo en grupo, en los que están comprometidas todas las áreas funcionales de la empresa, desde los departamentos de mercado y ventas hasta los de recursos humanos y formación, ya que todos ellos son potenciales generadores y beneficiarios de

la información corporativa. En un breve repaso, se pueden señalar como las aplicaciones más relevantes las siguientes:

- Publicación de documentos corporativos. Estos documentos pueden ser de difusión general, tales como boletines de noticias, guías de recursos humanos, memorias anuales, catálogos de productos, listas de precios, ofertas de trabajo, etc., o bien de difusión restringida como actas de reuniones, salarios, previsiones económicas, especificaciones de nuevos productos o información sobre proyectos.
- Acceso a búsquedas en directorios y bases de datos corporativas. Podemos citar como ejemplos: listas de teléfonos, documentación bibliográfica, lista de clientes, etc.
- Aplicaciones típicas del trabajo en colaboración. Por ejemplo: agenda de reuniones, gestión de proyectos, etc.
- Distribución de *software*. El uso de algún lenguaje de programación en el WWW permite la distribución sencilla de *software* a través de la red corporativa, asegurando que todos los usuarios utilizarán la última versión del programa, que sólo necesitará actualizarse en el servidor.
- Cursos de formación. Acceso a materiales de formación y tutoriales, incluyendo información en audio y vídeo.
- Presentaciones de productos dirigidos a los clientes de la empresa
- Mensajería electrónica y formularios electrónicos
- Foros de discusión. Los foros son áreas de debate, pequeñas salas virtuales donde los usuarios intercambian mensajes públicos (que pueden ser leídos por todos los usuarios que estén en el foro), con una estructura muy parecida a la de los *newsgroups* de USENET.

- Pláticas en tiempo real o en línea (*Chats*). Son también pequeñas salas virtuales al igual que los foros de discusión, con la diferencia de que en estos los comentarios son vistos por otras personas al mismo tiempo y pueden ser contestados de igual forma. Es decir la interacción entre usuarios es en tiempo real.

Es interesante recalcar que la tecnología Intranet permite producir y publicar información a distintos niveles dentro de la organización, desde información general de la propia empresa hasta información individual generada por cada persona, pasando por la específica de cada departamento, y todo ello presentado mediante una interfaz de usuario gráfica basada en hipertextos, que por su diseño y sencillez se considera dentro de las denominadas *End User Comfortable Interfaces*, es decir adaptada a las necesidades de un usuario final no especialista, a diferencia de las conceptuadas simplemente como *Graphical User Interfaces* (GUI's), que al no tener en cuenta dichas necesidades son poco aprovechadas en su funcionalidad total, poniendo en duda la productividad que con ella obtiene un usuario normal.

Esta es una cuestión polémica de suficiente interés si se tiene en cuenta que más del 50% de los servidores World Wide Web existentes están destinados a ofrecer servicio en Intranets.

En resumen podemos decir que las empresas que en la actualidad tienen instalada una Intranet la emplean para las siguientes utilidades<sup>13</sup>:

- Publicación de documentos de la empresa. (86%)
- Correo electrónico. (73%)
- Acceso a bases de datos. (43%)
- Foros de discusión. (39%)

---

<sup>13</sup> Bit 99. Intranet: Un nuevo concepto para el manejo de la información corporativa. Agosto de 1998.  
<http://www.iies.es/teleco/bit/bit99/intranet.htm>



- Distribución de *software*. (30%)
- Procesos de transacción. (18%)

### 2.8.1 Posibilidades comerciales

Una gran variedad de aplicaciones se ejecutan en el entorno comercial de la Intranet. Algunas se derivan directamente de los servicios de la Intranet, otras son desarrollos personalizados de cada empresa y otras las suministran vendedores de *software* comercial.

Estas aplicaciones resuelven necesidades comerciales urgentes, como las que se mencionan a continuación:

- Los vendedores necesitan información actualizada sobre los productos y servicios de la empresa, así como un sistema que permita aceptar pedidos desde la carretera y llevar un seguimiento de éstos.
- Los miembros de un equipo de proyecto, que pueden estar distribuidos en distintos territorios e incluso husos horarios, necesitan una forma de comunicarse y compartir información que les permita trabajar de la manera más eficiente.
- No se comparte información entre distintos equipos de un proyecto, lo que produce solapamientos e incongruencias. Se necesita un método universal para que cada equipo de proyecto suministre información y documentación de su trabajo y resultados.
- El personal de asistencia técnica necesita una forma de llevar un seguimiento de los clientes y sus problemas, para asegurar una resolución satisfactoria de cada problema que se comunique y ofrecer a los clientes un medio para "ayudarse a sí mismos" a la hora de resolver problemas básicos.

- Numerosas empresas grandes usan cinco o seis, e incluso más, sistemas de correo electrónico propietarios existentes y tienen enormes problemas de compatibilidad y trabajo conjunto. Por ejemplo, cada sistema de correo electrónico tiene su propio directorio propietario, y los archivos anexados no se transmiten sin problemas a través de los distintos sistemas de correo electrónico, lo que crea islas de usuarios en lugar de conexiones infalibles. Una empresa necesita un solo sistema de correo electrónico capaz de incluir a todos los escritorios y todas las personas.
- Los nuevos empleados necesitan una forma de encontrar la información relativa a los procedimientos, la organización y las ventajas de la empresa tan pronto como empiezan a trabajar.
- El personal de mercado necesita un método de acceso, a través de una interfaz uniforme en cualquier escritorio, a todas las bases de datos de estudios de clientes y mercados que mantiene la empresa.

Estos son sólo algunos ejemplos de las aplicaciones que la mayoría de las empresas comerciales necesitan y que una Intranet puede suministrarles.

## *CAPÍTULO III*

### *RECURSOS DISPONIBLES EN EL INSTITUTO NACIONAL DE ECOLOGÍA*

---

#### **3.1 Resumen del Programa de Desarrollo Informático contenido en el Plan Nacional de Desarrollo 1995-2000<sup>14</sup>**

##### **3.1.1 Introducción**

En los umbrales del nuevo milenio, el mundo está experimentando una nueva era tecnológica, que surge y se fundamenta en la convergencia tecnológica entre las ciencias de la computación y las telecomunicaciones, dando origen a la informática. La notable evolución ocurrida en cada una de estas tecnologías ha modificado drásticamente nuestro entorno. Los avances, tanto en capacidad de procesamiento como de transmisión de información e imágenes, son asombrosos. Sin embargo, quizá lo sea aún más el amplio ámbito en el cual ha encontrado aplicación esta tecnología.

En diversas épocas de la historia de la humanidad han acontecido desarrollos tecnológicos con impactos de gran trascendencia, como la imprenta en el siglo XV o la revolución industrial en los siglos XVIII y XIX. En el caso de la imprenta ésta propició el surgimiento de la escritura y la lectura como habilidades sociales; permitió una amplia comunicación a distancia en forma impresa: dió impulso a la generación de conocimientos; y motivó la evolución política, cultural y social de las diversas regiones del mundo. Por lo que toca a la revolución industrial, ésta incrementó las capacidades productivas, aumentó la disponibilidad de satisfactores y diversificó las opciones de empleo. La industrialización propició desplazamientos del campo a las ciudades y produjo un desarrollo heterogéneo entre las naciones, redefiniendo la propia arquitectura del mundo.

---

<sup>14</sup> Revista Soluciones Avanzadas. Mayo 1996

Así como la imprenta amplió las capacidades de comunicación de ideas, y la revolución industrial amplió las capacidades motrices del ser humano, ahora la informática recoge ambos efectos y los multiplica en forma dramática. Esta permite la difusión prácticamente instantánea del conocimiento; contribuye a la automatización acelerada de procesos; incrementó las capacidades intelectuales del hombre; y concreta la visión del mundo como idea global.

Las áreas de impacto son múltiples y diversas. En el ámbito de la medicina, los profesionistas pueden consultar a distancia la historia clínica de un paciente, radiografías, electrocardiogramas, bibliografía sobre padecimientos y nuevos tratamientos, y más aún intercambiar opiniones con colegas de otras latitudes. En el sector educativo, la conexión a redes de computadoras permite que los individuos puedan profundizar en sus áreas de interés, teniendo acceso a información contenida en acervos anteriormente fuera de su alcance. En la actividad económica la informática ha influido favorablemente en todos los sectores, particularmente en el sector servicios el cual es, incluso, el motor del crecimiento económico, y el de mayor generación de empleos en múltiples países. La informática también ha permitido que individuos que se encuentran en polos opuestos del mundo, puedan intercambiar información e ideas, textos e imágenes.

Es claro que la informática está modificando y modificará aún más nuestra vida cotidiana, nuestra forma de ver el mundo y de relacionarnos con él. Estos ejemplos, entre otros muchos que podrían citarse, evidencian la importancia de la informática como herramienta estratégica para el desarrollo de los países en lo que resta del siglo XX y por supuesto en el siglo XXI.

Por todo lo anterior, no sorprende el carácter prioritario otorgado al desarrollo informático por parte del Poder Ejecutivo Federal. Tampoco sorprende que cuando el Presidente de la República, Dr. Ernesto Zedillo, convocó a los mexicanos a participar en la integración del Plan Nacional de Desarrollo 1995-2000, el impulso a la informática haya sido tema recurrente en los diversos eventos de Consulta Popular. Dentro de éstos, se organizó un Foro de Consulta sobre Informática. Tanto en este foro, como en los otros, se captaron propuestas de diversos sectores de la sociedad y se reconoció la necesidad de integrar una estrategia nacional para aprovechar esta tecnología.

Así, entre los programas considerados por el Plan Nacional de Desarrollo 1995-2000, se contempló la elaboración de uno referente a la informática, reconociendo a ésta como herramienta de cambio y palanca de modernización, que beneficia a todos cuando se utiliza en forma adecuada.

Desde entonces los diversos sectores se sumaron al proceso de conformación del Programa de Desarrollo Informático. En este proceso se contó con opiniones de expertos, maestros, periodistas, investigadores, estudiantes, empresarios profesionistas, servidores públicos y personas interesadas en el tema, cuyas contribuciones fueron cuidadosamente analizadas. Además, es preciso reconocer que se contó con la decidida participación de diversos cuerpos colegiados. Merece particular mención, la participación del Grupo Consultivo de Política Informática, órgano que asesora al INEGI en este campo y que está integrado por más de treinta destacados especialistas provenientes de los sectores académico, privado y público; la del Comité de Informática de la Administración Pública Estatal y Municipal, conformado por los responsables de informática de los gobiernos estatales y municipales; la del Comité de Autoridades de Informática de la Administración Pública, compuesto por los titulares de las áreas de informática en entidades y dependencias del Gobierno Federal; y la de miembros del Honorable Congreso de la Unión y de Cámaras del Sector Privado. Todas estas instancias, con sus significativas aportaciones, contribuyeron en la conformación de un programa fundamental para el progreso de México.

El Programa de Desarrollo Informático, que es de observancia obligatoria para las dependencias y entidades de la Administración Pública Federal, no es sólo un programa sectorial de la industria informática, no se restringe a una visión de mercado de bienes y servicios informáticos, sino que es un programa de carácter estratégico que atiende múltiples aristas para impulsar el uso de esta tecnología, a favor de los grandes propósitos nacionales. El Programa establece seis objetivos generales:

Promover el uso y mayor aprovechamiento de la informática en los sectores público, privado y social del país; impulsar la formación de recursos humanos en informática; estimular la investigación científica y tecnológica en informática; desarrollar la industria informática

nacional; desarrollar la infraestructura de redes de datos; consolidar instancias de coordinación y establecer disposiciones jurídicas adecuadas para la actividad informática.

### **3.1.2 La Informática en el Plan Nacional de Desarrollo 1995-2000**

El Plan Nacional de Desarrollo 1995-2000 contiene las estrategias fundamentales para la construcción de un México mejor. En la consecución de los grandes objetivos nacionales consignados en él, la informática es una herramienta de apoyo esencial.

En efecto, a través del uso de la informática es posible realizar un seguimiento preciso y detallado de las características físicas del territorio, elemento consustancial y primigenio de nuestra nación. Además, apoyados en ella, podemos ampliar y consolidar la presencia de México en el mundo y asegurarnos de que la cultura e identidad nacionales se fortalezcan. De esta manera, la informática contribuye a fortalecer el ejercicio pleno de nuestra soberanía.

Dicha tecnología también coadyuva a consolidar un país de leyes y de justicia. Permite a las instancias responsables de la seguridad pública el acceso a información necesaria para combatir la delincuencia y facilita la creación de bases de datos para un eficiente seguimiento de los procesos judiciales.

La informática también aporta elementos valiosos para alcanzar un pleno desarrollo democrático. Al acrecentar las posibilidades de acceso a la información, esta valiosa herramienta tecnológica permite una sociedad más consciente y con mayores oportunidades de participación en todas las actividades de la vida nacional.

También apoya al federalismo al contribuir en los procesos de redistribución de competencias, responsabilidades y capacidades de decisión entre los tres órdenes del gobierno de la República. Con ello, coadyuva al fortalecimiento de los estados y de sus municipios, células básicas del tejido que da forma y sustento al pacto de unidad de los mexicanos. La informática propicia el desarrollo social, al apoyar funciones estratégicas de las instituciones que ofrecen servicios en materia de educación y salud entre otros. Además, da soporte a las tareas de ordenamiento

territorial y ecológico, y a las acciones para el desarrollo urbano y rural, repercutiendo favorablemente en los niveles de bienestar de la población. Así mismo, la informática incrementa la competitividad y productividad de los sectores económicos, en beneficio de todos los mexicanos.

En consecuencia, esta tecnología no debe ser vista como una herramienta fría, de uso exclusivo de especialistas, de aplicación aislada y estrictamente técnica, sino como un elemento de la mayor trascendencia para el presente y para el futuro del país. Ejercicio pleno de la Soberanía, Estado de Derecho, Desarrollo Democrático, Bienestar Social y Crecimiento Económico, son, todos, objetivos nacionales en cuyo logro la Informática puede contribuir de manera decisiva.

Ante el amanecer del nuevo siglo y de un nuevo milenio, el Programa de Desarrollo Informático se constituye en una herramienta más de los mexicanos para trabajar, comprometidamente, en favor de nuestra patria. Con la decidida y entusiasta participación de todos, México sabrá aprovechar las bondades que ofrece la era informática en favor de los más nobles propósitos nacionales, para la construcción de un México más justo, más fuerte y más próspero.

### **3.1.3 Aprovechamiento de la Informática en los diversos sectores**

El primer objetivo general del Programa consiste en promover el uso y mayor aprovechamiento de la informática en los sectores público, privado y social del país.

#### **Sector Público**

##### **Elementos del diagnóstico**

En el caso del sector público, del total de su gasto en informática, el 57% es realizado por las empresas paraestatales y el 28% corresponde a las Secretarías de Estado. El restante 15%, lo ejercen los gobiernos estatales y municipales, lo cual es insuficiente para apoyar la eficaz dotación de servicios públicos. Además existe gran heterogeneidad de la inversión en informática

en dependencias y entidades, que se refleja en diferencias notables en cuanto a infraestructura y uso de esta tecnología. Adicionalmente, se tiene una inadecuada planeación informática institucional que se evidencia en alta proporción de inversión en equipos (a veces obsoletos) y reducida en sistemas, comunicaciones y capacitación.

### **Principales acciones**

Con el fin de obtener el máximo aprovechamiento de la informática en el sector público, el Programa de Desarrollo Informático contempla muy diversas acciones, como impulsar una adecuada planeación informática en el marco de los programas y proyectos prioritarios de cada dependencia; y la vinculación con centros de investigación que posibiliten la incorporación de tecnología de punta.

Además, se fortalecerá y ampliará la capacitación de los servidores públicos, asegurando que los niveles de decisión cuenten con una sólida cultura informática y que el personal de las áreas de informática disponga de mecanismos de actualización que les permita la incorporación adecuada de nuevas tecnologías.

También con alta prioridad, se impulsará el desarrollo de sistemas de cómputo para elevar la eficiencia de tareas administrativas y, sobre todo, para mejorar las actividades sustantivas de servicios de atención al público. Adicionalmente, el Programa propone la creación de bases de datos y sistemas de intercambio de información que fortalezcan la coordinación entre las diversas instancias federales, estatales y municipales, haciendo más eficiente la relación de éstas con la población en general.

Con la implementación de estas acciones, entre otras, se garantizará el aprovechamiento de la informática para mejorar las tareas del sector público, aumentando su eficiencia, y mejorando la calidad y cobertura de los servicios en beneficio de la ciudadanía.



## **Sector privado**

### **Elementos del diagnóstico**

Por lo que se refiere al sector privado, la demanda de bienes y servicios informáticos ha aumentado en forma considerable; actualmente, este sector realiza el 70% de las compras en el mercado informática nacional. Sin embargo, prácticamente la totalidad de la inversión en informática es efectuada por los grandes grupos industriales, comerciales y financieros; en contrapartida, la correspondiente a las micro, pequeñas y medianas empresas es escasa o nula a pesar de que constituyen la gran mayoría de los establecimientos económicos, emplean a casi 80% de la población ocupada y su producción alcanza más del 65% del total nacional. Esta carencia, coloca a este grupo de empresas en una situación de franca desventaja para poder competir y participar en los mercados nacionales e internacionales.

### **Principales acciones**

Para lograr un mayor aprovechamiento de la informática en el sector privado el Programa de Desarrollo Informático contempla diversas acciones. En particular, se impulsará la incorporación de la informática en los programas de capacitación, para elevar el nivel de preparación técnica en esta materia de los trabajadores y para elevar la cultura informática de los directivos. Además, se promoverá que las empresas se vinculen con centros académicos y de investigación, para atender necesidades informativas específicas de las unidades productivas, reduciendo los tiempos de asimilación de la tecnología informática.

Para estimular la adopción de la informática en las micro, pequeñas y medianas empresas las áreas de fomento industrial implantarán un programa que considera mecanismos ágiles de financiamiento para proyectos informativos integrales. También se facilitará el acceso a redes de datos con información relativa a clientes y proveedores, insumos y mercados.

El conjunto de acciones a ser instrumentadas buscan que la informática sea aprovechada -en el sector privado- como herramienta para mejorar la productividad y la competitividad de las empresas mexicanas.

## **Sector social**

### **Elementos del diagnóstico**

Por lo que respecta al sector social en México existe una incipiente cultura informática en la población en general. Por ejemplo, mientras que en países desarrollados se tienen más de 15 computadoras personales por cada 100 habitantes, en México existen solamente 2 por cada 100. En lo que toca a los hogares que cuentan con una computadora, en los países desarrollados éstos alcanzan el 30%; en tanto que en México solamente representan el 3%. Además, en México existe un alto nivel de analfabetismo informática, ya que solamente 5.6% de la población sabe usar una computadora.

### **Principales acciones**

Es claro que las actividades contempladas en el Programa para incrementar el aprovechamiento de la informática en los sectores público y privado tendrán un importante impacto en el sector social. Sin embargo, con el propósito de estimular aún más su uso en el sector social, el Programa propone amplias acciones para sensibilizar a la población sobre los beneficios de la informática. Además, para enseñar cómo usar una computadora, se transmitirán programas educativos por televisión dirigidos a la población en general. También, se promoverá el mayor acceso a redes y servicios de información en centros comunitarios, bibliotecas públicas e instituciones educativas que estimulen a la población a aprovechar esta tecnología.

En síntesis, el primer objetivo general del Programa de Desarrollo Informático es promover el uso eficiente de la informática en los sectores público, privado y social del país. Sin embargo, también es necesario reconocer que, para satisfacer las necesidades informáticas de dichos sectores, es indispensable fortalecer la infraestructura informática nacional. En este sentido, México requiere disponer de recursos humanos capacitados en informática; desarrollar una mayor actividad científica y tecnológica en esta materia; desarrollar una más sólida industria informática nacional, y contar con extensas redes de transmisión de datos. Todos estos elementos constituyen la infraestructura informática nacional, y en ellos se sustenta la posibilidad de uso de esta

tecnología de acuerdo a las necesidades de cada sector y en función de la cultura que nos distingue como nación.

### **3.1.4 El Desarrollo de la Infraestructura Informática**

El segundo objetivo general del Programa es impulsar la formación de recursos humanos en informática.

#### **Recursos humanos en informática**

##### **Elementos del Diagnóstico**

Los recursos humanos constituyen el componente más significativo para garantizar un sano aprovechamiento de las tecnologías de la información en México. En los últimos 30 años, se ha registrado en nuestro país un aumento considerable de programas educativos a nivel técnico y de licenciatura en informática; así mismo, la matrícula en dichos niveles ha crecido en forma considerable. Sin embargo, los contenidos de los programas son muy diversos y el equipamiento informático de las instituciones educativas muestra, en algunos casos, carencias importantes. En consecuencia, la formación de los egresados es deficiente. Además, existen pocos programas de posgrado en informática, lo que limita las posibilidades de especialización.

##### **Principales acciones**

Por todo lo anterior, el Programa propone evaluar y certificar la calidad de los planes de estudio a través de las instancias correspondientes con el propósito de impulsar mejoras en los programas educativos en informática de todos los niveles. Adicionalmente, se reforzarán los mecanismos que permitan dotar de mejor infraestructura informática a las instituciones educativas.

Para mejorar el nivel del personal docente en informática se ampliarán los estudios que permiten conocer las necesidades inmediatas de actualización y así poder instrumentar con mayor eficiencia, programas específicos de capacitación. En complemento, para contribuir a la

formación del personal docente y de los alumnos, se promoverá el acceso de las instituciones de educación superior a la infraestructura de redes académicas y servicios de información.

Además, en coordinación con las autoridades correspondientes, se reforzará y/o se incorporará a la informática en los programas de educación, tanto en los niveles de educación básica y media, como en el nivel superior; con ello, se desarrollará una mayor cultura informática en la sociedad en general.

El tercer objetivo general del Programa consiste en estimular la investigación científica y tecnológica en informática.

### **Investigación científica y tecnológica en informática**

#### **Elementos del diagnóstico**

En el país existen menos de 500 investigadores en informática. De éstos, solamente 33 forman parte del Sistema Nacional de Investigadores, lo que representa menos del 1 % de los miembros de dicho sistema.

Adicionalmente, los grupos de investigación son pocos y de tamaño reducido. En estas condiciones es difícil lograr una amplia producción original y de alta calidad.

#### **Principales acciones**

El Programa confiere elevada prioridad a la informática como disciplina científica y tecnológica, por su trascendencia para el desarrollo del país. Por ello, se propiciará la inversión privada para la investigación en informática. También se fortalecerá la inversión pública en proyectos de investigación y desarrollo en la materia. Además, en la estrategia nacional de desarrollo científico y tecnológico se dará prioridad a la formación de investigadores en informática y a la consolidación de grupos de investigación.

El cuarto objetivo general, claramente relacionado con el fortalecimiento de la infraestructura informática del país, se refiere al desarrollo de la industria informática nacional.

## **Infraestructura informática**

### **Elementos del diagnóstico**

El mercado informático nacional representa casi 30% del latinoamericano pero únicamente 0.8% del mercado mundial, proporción menor a la que le correspondería si se toma en cuenta el monto poblacional y el tamaño de nuestra economía. Aunado a esto la industria informática mexicana se ha concentrado en la distribución de computadoras y de paquetería de origen extranjero. Así mismo, existe equipo que no se utiliza en forma adecuada por falta de herramientas de explotación y de personal capacitado. Además, la oferta de servicios de consultoría, de capacitación y de desarrollo de sistemas es aún incipiente. Por ello, se requiere desarrollar la industria informática en nuestro país.

### **Principales acciones**

Cabe notar que todas las acciones propuestas en este programa que buscan impulsar el uso de la informática en los sectores público, privado y social, y las relativas al fortalecimiento de la infraestructura, permitirán, sin duda, el desarrollo de la industria informática nacional.

Adicionalmente con el propósito de contar con productos y servicios informáticos, que satisfagan los requerimientos de los usuarios, el programa plantea el uso de normas internacionales de calidad en bienes y servicios informativos; así mismo, propone la consolidación y desarrollo de empresas privadas para satisfacer la creciente demanda. Además, plantea aumentar la producción o el valor agregado de sistemas informativos de origen nacional, para aprovechar la gran potencialidad exportadora de México en este sector.

El quinto objetivo que contempla el Programa es desarrollar la infraestructura de redes de datos.

## **Infraestructura de redes de datos**

### **Elementos del diagnóstico**

Las redes de datos permiten la transmisión de grandes volúmenes de información y, por supuesto, el acceso a ella.

En México el uso de redes de transmisión de datos aún es reducido, existiendo incluso la limitante de que en nuestro país se tienen menos de 10 líneas telefónicas por cada 100 habitantes; mientras que en algunos países desarrollados la relación ya llega a más de 70 por cada 100 habitantes.

### **Principales acciones**

Por ello, el Programa de Desarrollo Informático contempla, entre las principales acciones que se llevarán a cabo en el ámbito de la infraestructura de redes de datos, promover el desarrollo de servicios en línea para consulta de información y transmisión de datos. Así mismo, consolidar el marco regulatorio que propicie el desarrollo de redes de acceso y transmisión por parte del sector privado, con niveles de calidad y costos competitivos a nivel internacional. También, garantizar la interconexión con las redes informativas globales. Todo ello, en el marco de las estrategias del sector comunicaciones y transportes.

El sexto objetivo del Programa es consolidar instancias de coordinación y establecer disposiciones jurídicas adecuadas para la actividad informática.

### **Instancias de coordinación y establecer disposiciones jurídicas adecuadas para la actividad informática**

#### **Elementos del diagnóstico**

En el país existen numerosas agrupaciones profesionales, industriales y académicas en el ámbito informática que, por lo general, no actúan de manera coordinada. Así mismo, múltiples dependencias y entidades de la Administración Pública Federal y Local tienen atribuciones y/o responsabilidades en materia de informática. En cuanto al ámbito normativo, existen diversas disposiciones jurídicas que rigen actividades vinculadas con aspectos informativos; sin embargo, éstas no han sido actualizadas al ritmo de la dinámica evolución tecnológica.

## **Principales acciones**

Por todo lo anterior, y con el propósito de asegurar una participación coordinada a favor del desarrollo informática nacional, se fortalecerán órganos colegiados, como el Comité de Autoridades de Informática de la Administración Pública y el Comité de Informática para la Administración Pública Estatal y Municipal; también se estará conformando en materia informática una instancia de coordinación entre la Administración Pública Federal y los diversos sectores.

Por otra parte, se seguirá trabajando con el Congreso de la Unión en la revisión de las disposiciones jurídicas para que el marco normativo favorezca el aprovechamiento y el desarrollo de la informática; también dentro del ámbito jurídico, se estudian temas como la propiedad intelectual de sistemas de procesamiento y de información, y el carácter probatorio de los documentos electrónicos.

### **3.1.5 Mecanismos de coordinación y seguimiento**

En complemento a las múltiples acciones contempladas en el programa para el logro de cada uno de los objetivos enunciados, éste identifica un conjunto de Proyectos Informáticos Nacionales y Regionales de fundamental importancia. Estos proyectos se instrumentarán para atender una problemática importante del país donde las tecnologías de la información constituyan un elemento de primer orden para su solución. Hasta ahora, los proyectos nacionales identificados, algunos de los cuales ya están en fase de desarrollo, abarcan múltiples temas, tales como: educación, salud, seguridad pública y administración. Por su parte, los Proyectos Informáticos Regionales corresponden a temas de interés común para gobiernos estatales y municipales. Tal es el caso de los proyectos relativos a catastros y registros públicos. Cabe destacar que tanto para los Proyectos Informáticos Nacionales como para los Regionales están previstas actividades específicas y tiempos de ejecución, así como mecanismos de coordinación y evaluación. Además, para asegurar el cumplimiento de las metas del Programa, se instala una Comisión de Seguimiento que está integrada por representantes de los diversos sectores.

## 3.2 El Instituto Nacional de Ecología

### 3.2.1 Semblanza histórica<sup>15</sup>

Al igual que en otros países, la experiencia sobre gestión ambiental que se tiene en México es reciente. Entre los primeros antecedentes que se registran, puede señalarse al Reglamento de Higiene en el Trabajo, promulgado en 1945, que promovía acciones orientadas a proteger a la población de los peligros de la insalubridad y de las molestias ocasionadas por la industria, a cargo de la entonces Secretaría de Salubridad y Asistencia Pública.

En la década de los setenta, con la Ley Federal para Prevenir y Controlar la Contaminación Ambiental, se formuló un primer programa para disminuir los efectos de la contaminación del medio. Dentro del sector salud, empieza a funcionar la Subsecretaría de Mejoramiento del Ambiente. No obstante, muchos aspectos relacionados con el control ambiental quedaron dispersos en la jurisdicción de otros sectores. Por ejemplo, el cuidado de los recursos naturales no fue considerado en el programa.

El Instituto Nacional de Ecología tiene su origen en la antigua Secretaría de Salubridad y Asistencia, que durante el periodo 1976-1982, contaba con una Subsecretaría de Mejoramiento del Ambiente; en la Secretaría de Asentamientos Humanos y Obras Públicas, donde existía una Dirección General de Ecología Urbana y en la Secretaría de Agricultura y Recursos Hidráulicos que disponía de una Dirección General de Usos del Agua, Prevención y Control de la Contaminación.

El 29 de diciembre de 1982, ante la creciente problemática en materia ecológica y a fin de combatirla integralmente, el Ejecutivo Federal modificó la Ley Orgánica de la Administración Pública Federal creando la Secretaría de Desarrollo Urbano y Ecología, con todas las atribuciones inherentes y contando con una Subsecretaría de Ecología de donde surge el primer Programa Nacional de Ecología.

---

<sup>15</sup> Información obtenida de la página del Instituto Nacional de Ecología. <http://www.ine.gob.mx/>



En 1988 es publicada la Ley General del Equilibrio Ecológico y la Protección al Ambiente, la cual entra en vigor en marzo del mismo año, promueve una descentralización de funciones, impulsa la creación de leyes estatales de ecología y propicia el surgimiento de órganos estatales en las estructuras de los gobiernos de cada entidad, para la atención de los aspectos ecológicos de jurisdicción local.

En este marco jurídico de referencia ha sido administrada la gestión ambiental, impulsando el desarrollo sustentable, la armonía entre el crecimiento económico y la ecología, así como estableciendo las políticas, y acciones públicas para la protección del medio ambiente, la conservación y aprovechamiento ecológico de los recursos naturales.

El 25 de mayo de 1992, fue publicada en el Diario Oficial de la Federación otra reforma a la Ley Orgánica de la Administración Pública Federal, mediante la cual es creada la Secretaría de Desarrollo Social, contemplando en su organización, como uno de sus órganos desconcentrados, al Instituto Nacional de Ecología con atribuciones técnicas y normativas en materia de ecología.

Con base en la citada reforma, el 17 de julio de 1992 se publica el Acuerdo que regula la organización y funcionamiento del Instituto Nacional de Ecología y de la Procuraduría Federal de Protección al Ambiente, que estuvo vigente hasta el 8 de julio de 1996.

El 28 de diciembre de 1994 fueron publicadas nuevas reformas a la Ley Orgánica de la Administración Pública Federal, por medio de las cuales se crea la Secretaría de Medio Ambiente, Recursos Naturales y Pesca (SEMARNAP), con las facultades que le confiere el Artículo 32 bis, quedando adscrito como uno de sus órganos desconcentrados el Instituto Nacional de Ecología.

Subsecuentemente, el 8 de julio de 1996, se publica en el Diario Oficial de la Federación el Reglamento Interior de la Secretaría de Medio Ambiente, Recursos Naturales y Pesca, que menciona en el Artículo 2º, como órganos desconcentrados de la misma, al Instituto Nacional de Ecología, conjuntamente con la Comisión Nacional del Agua, el Instituto Mexicano de Tecnología del Agua, la Procuraduría Federal de Protección al Ambiente y el Instituto Nacional de la Pesca. El Capítulo XI del citado Reglamento Interno contiene, en sus Artículos 54 al 61, las

atribuciones del Instituto Nacional de Ecología y de las Unidades Administrativas con que este cuenta.

En 1982, la Ley Federal de Protección del Ambiente amplió el ámbito de acción corrigiendo omisiones de la legislación anterior; en ella se incorporaron aspectos sobre la protección y conservación de los ecosistemas, con apertura a un nuevo marco legal, dirigido a la protección de la fauna, de la flora, del suelo y del agua. Además, empezó a manejar principios de orden ecológico, asociados a los mecanismos de desarrollo socioeconómico. A fines de 1982, se expidió el decreto de reformas a la Ley Orgánica de la Administración Pública, para la creación de la Secretaría de Desarrollo Urbano y Ecología (SEDUE), organismo encargado de planear y dirigir la política ambiental a través de la extinta Subsecretaría de Ecología.

Frente al cambio estructural mencionado, la presión ejercida por la ciudadanía, para acelerar los programas gubernamentales de mejoramiento del ambiente, indujeron transformaciones políticas mucho más rápidas. La Ley General del Equilibrio Ecológico y la Protección al Ambiente, en vigor a partir de marzo de 1988, introdujo cambios radicales en la estrategia a seguir. Con esa ley, se establece el principio rector esencial para planear el uso racional del suelo, a través del ordenamiento ecológico; se busca atacar causas y no sólo penalizar efectos; se da lugar a la participación social, como complemento indispensable de la acción del estado y se extiende el rango de la gestión ambiental tanto a los otros dos órdenes de gobierno como a los sectores social y privado.

En este contexto la variable ambiental se sitúa en un lugar prominente, por primera vez en la historia de México, explicitando que el principio de desarrollo sustentable debe guiar la política ambiental.

En junio de 1992, con la creación de la Secretaría de Desarrollo Social, la competencia ambiental se distribuye por una parte, en el Instituto Nacional de Ecología cuya atribución es la de conducir la política ecológica del país, enfatizando el carácter normativo de su acción, y por otra, en la Procuraduría Federal de Protección al Ambiente con atribuciones de carácter coercitivo a efecto de vigilar el cumplimiento de la Ley, a través de la aplicación de la legislación vigente.

La exigencia social en favor de mejores condiciones ambientales rebasa las acciones de la política ecológica, por este motivo las reformas a la Ley del Equilibrio Ecológico y la Protección al Ambiente, aprobadas en octubre de 1996 y publicadas en el diario oficial del viernes 13 de diciembre de 1996, incorporan una iniciativa de excepción como el derecho a la información; perfecciona los instrumentos de gestión ambiental como el impacto ambiental y fortalece y consolida al ordenamiento ecológico como instrumento básico de planeación ambiental y se tipifica el delito ecológico en las modificaciones realizadas al Código Penal.

### **3.2.2 Normas Oficiales Mexicanas**

Con la puesta en vigor de la Ley Federal sobre Metrología y Normalización en 1992, el ámbito normativo ambiental observa una innovación importante al establecer el cambio de normas técnicas ecológicas a normas oficiales mexicanas, estableciendo el procedimiento para su elaboración.

La Ley establece que en la elaboración de normas oficiales mexicanas deben participar, ejerciendo sus respectivas atribuciones, las dependencias a quienes corresponda la regulación o control del producto, servicio, método, proceso o instalación, acción o materia a normalizar. Las innovaciones relevantes que incorpora este ordenamiento son:

La creación de Comités Consultivos Nacionales de Normalización, órganos encargados de la elaboración y expedición de las normas oficiales mexicanas y de promover su cumplimiento; el objeto y procedimiento de formulación, expedición y certificación de las normas; y, la participación de grandes sectores de la sociedad interesados en esta actividad, que abarca desde la iniciativa en su caso, hasta el consenso, con la modalidad de que los proyectos son puestos a consideración pública para recibir comentarios que se publican en el Diario Oficial, previamente a su expedición definitiva.

El anteproyecto conlleva un análisis con una justificación científica, técnica o de protección al consumidor que apoye la expedición de la norma; el beneficio potencial de su aplicación y su

costo potencial cuantificado en términos monetarios, identificando personas o grupos beneficiados o que tienen a su cargo los costos, por mencionar algunos, dando cabal cumplimiento al concepto de transparencia en el procedimiento de elaboración y expedición de normas oficiales mexicanas.

El Comité Consultivo Nacional de Normalización para la Protección Ambiental se instala formalmente el 17 de diciembre de 1992, constituido por un Presidente, un Secretariado Técnico, Subcomités y Grupos de Trabajo. Para este propósito se invita a integrarlo al sector público, industrial y académico, buscando la participación de especialistas en la materia para lograr una normatividad apegada a la realidad y necesidades del país.

### **3.2.3 Principios y Orientaciones<sup>16</sup>**

La Secretaría de Medio Ambiente, Recursos Naturales y Pesca (SEMARNAP), se creó el 28 de diciembre de 1994, al inicio de la actual administración con el mandato presidencial de constituirse en una dependencia integradora, a cargo del aprovechamiento ordenado de los recursos naturales y la protección ambiental bajo un propósito explícito en favor del desarrollo sustentable. En ella, el Instituto Nacional de Ecología (INE), se mantiene como un órgano desconcentrado.

El INE tiene a su cargo el diseño de la política ecológica general y la aplicación de sus diferentes instrumentos. Sus responsabilidades abarcan temáticas tanto sectoriales como regionales, que se despliegan en diferentes planos de actuación. A raíz de la reforma de la Ley Orgánica de la Administración Pública Federal, el Instituto asume nuevas tareas que amplifican sus espacios de incumbencia.

En este sentido, el INE tiene a su cargo el diseño y manejo de los diferentes instrumentos de regulación y gestión ambiental con diversos grados de complejidad en su implementación como son:

---

<sup>16</sup> "Principios, orientaciones y agenda de trabajo del Instituto Nacional de Ecología"

- Normatividad
- Desarrollo de mecanismos económicos
- Sistema de licencias y permisos
- Evaluación de impacto ambiental de proyectos y actividades
- Ordenamiento ecológico del territorio
- Evaluación de riesgo y prevención de accidentes
- Dictaminación de proyectos y programas de manejo de residuos peligrosos
- Control de movimientos transfronterizos de materiales y residuos peligrosos
- Política en materia de residuos sólidos urbanos
- Promoción de infraestructura ambiental
- Declaratoria y manejo de áreas naturales protegidas
- Conservación y manejo de flora y fauna silvestre
- Promoción del manejo sustentable de la flora y fauna silvestre
- Permisos y licencias para el aprovechamiento de flora y fauna silvestre
- Control de movimientos transfronterizos de flora y fauna silvestre
- Aplicación en México de convenios internacionales
- Educación y capacitación ambiental
- Tecnologías para el aprovechamiento sustentable de recursos y calidad ambiental de procesos productivos, servicios y transporte
- Fomento a la investigación científica y tecnológica para mejorar la gestión ambiental y uso sustentable de los recursos
- Sistema de información ambiental

Asumir todas estas responsabilidades de manera eficiente ha implicado reconfigurar y fortalecer el diseño institucional del INE, de acuerdo con criterios básicos como son:

- Atención específica a sectores prioritarios
- Consolidación de instrumentos complementarios de política
- Construcción de enlaces intersectoriales
- Jerarquización consistente con prioridades y niveles de complejidad
- Mayor penetración de instrumentos estratégicos
- Generación, recuperación y aprovechamiento de información
- Equilibrio y conjugación de planos sectoriales y regionales de actividad
- Eficiencia y alta calidad profesional
- Vinculación académica
- Dimensión de apoyo a la política social

Por último se debe mencionar que los principios de política ambiental del INE son:

- **Sustentabilidad.** Equilibrio entre tecnología, patrones de producción y consumo, población y capacidades de carga de los sistemas biofísicos.
- **Promoción.** Trascender una actitud estrictamente regulatoria para participar activamente en la inducción de inversiones en infraestructura ambiental, en la creación de mercados y en el financiamiento para el desarrollo sustentable.
- **Eficiencia.** Lograr objetivos ambientales minimizando costos de oportunidad y logrando más con los recursos institucionales y presupuestales existentes. Ofrecer servicios gubernamentales de regulación ambiental de una manera transparente y expedita.

- Nuevo marco de incentivos. Promover iniciativas regulatorias a través de normas e instrumentos económicos que modifiquen el marco de incentivos (precios, reglas jurídicas y reglas sociales) que confrontan productores y consumidores, con el objeto de compatibilizar sus decisiones con las necesidades de protección ambiental y sustentabilidad.
- Financiamiento múltiple. Promover la apertura de nuevos cauces y mecanismos de financiamiento a los bienes públicos ambientales, involucrando a los individuos, a empresas privadas y públicas, a las organizaciones sociales, a organismos multilaterales y a fuentes bilaterales y fundaciones internacionales.
- Competitividad. Buscar que las regulaciones ambientales se traduzcan en una mayor productividad a través de mejores insumos, eficiencia energética, control de calidad, mejor mantenimiento y control de procesos, reciclaje y recuperación de materiales residuales. Esto, de tal forma que se logren simultáneamente metas de protección ambiental y una mejor posición competitiva para los productores.
- Certidumbre. Las regulaciones ambientales deben ofrecer un marco de certidumbre que afiance expectativas y permita decisiones y compromisos de inversión a largo plazo.
- Dimensión social. La política ambiental debe desplegar un visible componente social redistribuyendo los costos de la protección ambiental de manera equitativa, o bien buscando las compensaciones necesarias cuando esto no sea posible. En materia de recursos naturales, el manejo de los mismos debe traducirse en incentivos a la conservación a través de transferencias financieras a los grupos campesinos de más bajos ingresos.

### **3.3 Red del Instituto Nacional de Ecología**

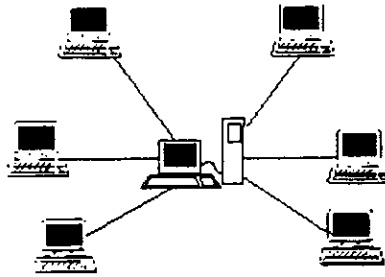
Para poder entender el funcionamiento y la tecnología empleada en la red del Instituto Nacional de Ecología es necesario hacer mención brevemente de algunos conceptos generales sobre redes de computadoras y en especial de las utilizadas en el INE. Por lo cual a continuación se hace una breve mención de éstas.

#### **3.3.1 Topologías**

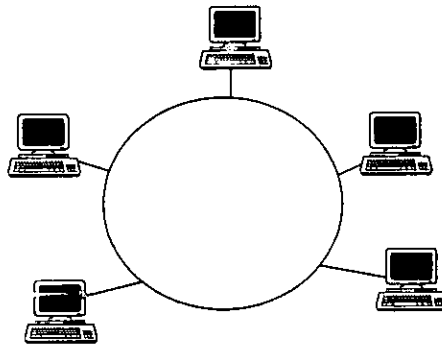
Podemos decir que en menos de 20 años las redes locales dejaron de ser experimentos de laboratorio para convertirse en productos comerciales de enorme utilización en las empresas. La topología de una red local se refiere a la forma en la que se conectan físicamente las computadoras (u otros dispositivos) a la red. En la figura 5 se muestran las tres topologías básicas de las redes locales: estrella, anillo y bus. Los medios de transmisión empleados con mayor frecuencia son el par trenzado, el cable coaxial y la fibra óptica.

◊

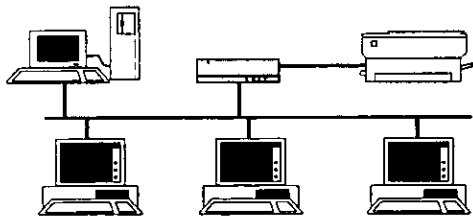




Estrella



Anillo



Bus

**Figura 5.** Topologías básicas de redes locales.

En una topología en estrella cada computadora se encuentra conectada por una línea punto a punto a un nodo central, esta topología presenta la desventaja de depender completamente de un nodo en particular.

Las topologías en anillo y bus no tienen este inconveniente. En la topología en anillo las computadoras se conectan entre sí por líneas punto a punto formando un circuito, mientras que en una topología en bus todas las computadoras se conectan a una línea multipunto. Las topologías en anillo y bus son las más empleadas en la actualidad.

### **3.3.2 Redes tipo Ethernet**

Una red local está formada por un conjunto de computadoras y otros dispositivos interconectados dentro de un área geográfica limitada, con el fin de intercambiar recursos e información. La mayoría de las redes locales se caracteriza por:

- Radio de acción pequeño, hasta de unos cinco kilómetros.
- Velocidad de transmisión en el orden de millones de bits por segundo.
- Ambiente relativamente libre de errores de transmisión.
- Medio de comunicación compartido por todos los dispositivos conectados a la red.
- Flexibilidad en la topología, es decir, facilidad en la modificación y reconfiguración de la distribución física de los dispositivos conectados a la red.

Ethernet, el tipo de red local más difundido en la actualidad, cumple con todas las características anteriores. Su primera implantación fue desarrollada en Xerox por Robert M. Metcalfe a principios de los años 70's, para conectar hasta 100 estaciones de trabajo en un área de 1 km. transfiriendo información a 2.94 Mbps. Es concebida y recomendada en ambiente de oficina y para áreas de industria ligera donde no se requieran tiempos de respuesta determinísticos.

Ethernet toma su nombre como recuerdo de aquella teoría del siglo XIX según la cual el universo estaba suspendido en una especie de éter por el que las ondas electromagnéticas podían propagarse.

En 1978 se publica la primera norma como un trabajo conjunto de las empresas Xerox, Intel y DEC. Esta es la base del estándar ANSI/IEEE 802.3 publicado en 1983 por el IEEE. La norma internacional ISO 8802/3 de la Organización Internacional de Estándares (ISO) también está basada en la especificación de 1978.

En Ethernet, el canal de comunicación común es un cable coaxial y un bus con impedancias de terminación en los extremos, en el que se conectan todos los dispositivos que forman la red.

Cada sitio en la red local tiene un identificador único: su dirección. Cuando una computadora desea mandar información a otro dispositivo, simplemente forma un paquete con el mensaje, la dirección del destinatario, su propia dirección y otra información. En la nomenclatura de redes locales, estos paquetes se llaman tramas. Una vez formada una trama, ésta se envía en serie, es decir, bit por bit a través del cable coaxial. Las señales en el bus son omnidireccionales, esto es, se difunden en los dos sentidos del cable, de tal manera que todos los sitios conectados a la red detectan la información. Aquel dispositivo que reconozca en la dirección destino su propia dirección, sabe que la trama contiene información dirigida hacia él y por lo tanto la leerá del bus. Los demás sitios ignoran esta trama.

### **3.3.3 Token Ring**

Una red local con topología de anillo está integrada por un conjunto de estaciones conectadas en serie, por medio de enlaces (unidireccionales) punto a punto (de par trenzado, cable coaxial o fibra óptica), de manera que se forma una trayectoria cerrada o anillo que permite la comunicación entre éstas.

Cuando una estación recibe datos por un enlace, los retransmite bit a bit por el otro enlace a la misma velocidad que los recibió. Debe observarse que cada estación introducirá un retardo

durante el proceso de regeneración y repetición de los bits. Las estaciones que integran el anillo envían los datos en forma de tramas que también contienen información de control y las direcciones de las estaciones de origen y destino.

De la misma forma que en otros tipos de redes locales (donde el medio físico se comparte entre otras estaciones), una estación que esté preparada para transmitir una o varias tramas deberá esperar su turno de acuerdo al protocolo de control de acceso al medio. Cuando una estación tenga el permiso para transmitir, insertará las tramas al anillo (una a la vez). Cada trama viajará por el anillo (en una sola dirección), pasando por todas las estaciones.

Aunque en la actualidad se conocen diversas técnicas para controlar el acceso al medio en una red con topología de anillo, la técnica más común se conoce como "Token Ring". Esta técnica basada en la circulación de un patrón de bits único (*token*) que otorga el permiso de transmisión, fue adoptada por el comité IEEE 802 encargado de los estándares para redes de área local y metropolitana. En 1989, el IEEE emitió el estándar IEEE 802.5 que describe el método de acceso y las especificaciones de la capa física para una red del tipo Token Ring. Este estándar es compatible con las redes Token Ring de IBM.

El estándar IEEE 802.5 describe el funcionamiento de una red Token Ring con las siguientes características:

- Topología: anillo
- Medio: Par trenzado. El uso de otros medios está sujeto a futuras consideraciones.
- Señalización: Manchester diferencial.
- Velocidad: 4 y 16 Mbps.
- Protocolo de control de acceso al medio: paso de *token*.

### 3.3.4 FDDI<sup>17</sup>

FDDI es una interfaz de datos distribuida sobre fibra óptica. Es una fibra óptica de alto rendimiento Token Ring LAN corriendo a 100 Mbps cubriendo distancias mayores a los 100 kilómetros de radio con hasta 1000 estaciones conectadas. Puede ser utilizada en la misma forma que las LAN's del tipo 802, pero con un ancho de banda más alto, generalmente se utiliza como red troncal para la interconexión de LAN del tipo 802.3 y 802.5.

FDDI utiliza un anillo doble -primario y secundario- de fibra óptica, este último como respaldo en caso de fallo del anillo primario. Los datos circulan en sentidos opuestos en cada anillo para facilitar la reconfiguración en caso de rotura de cualquiera de ellos, siendo posible construir redes FDDI con un único anillo.

FDDI utiliza fibras multimodo porque el costo adicional de las fibras de modo sencillo no se necesitan para redes corriendo a solo 100 Mbps. También utiliza LED's en lugar de láseres, no sólo debido a su bajo costo sino también porque FDDI puede algunas veces ser utilizado para conectarse directamente a las estaciones de trabajo de los usuarios. Existe el peligro de que por curiosidad algún usuario pudiera ocasionalmente desconectar el conector de la fibra y mirar directamente al flujo de bits viajando a 100 Mbps. Con el rayo láser activo el usuario podría terminar con su retina, en cambio los LED's son muy inofensivos para causar daño a la vista y capaces de soportar la transferencia de datos a esa velocidad. La especificación del diseño de la FDDI no exige más de un error en  $2.5 \times 10^{10}$  bits.

El cableado de FDDI está constituido por dos anillos de fibras, uno transmitiendo en el sentido de las manecillas del reloj, y el otro en sentido contrario. Si alguno de los dos se llega a desactivar, el otro puede emplearse como respaldo; si los dos se desactivaran en el mismo punto, por ejemplo, como consecuencia de un incendio o alguna otra causa en el conducto del cable, los dos anillos podrán unirse para formar un solo anillo que tendrá una longitud de casi el doble. En cada estación hay redes que pueden emplearse para unir a los dos anillos o "puentear" una estación, en el momento que ocurre un problema en cualquiera de ellos. También pueden utilizarse centrales de cables, como se vio en el protocolo 802.5.

---

<sup>17</sup> Por sus siglas en inglés *Fiber Distributed Data Interface*.

Mediante la FDDI se definen dos clases de estaciones, la A y B. Las estaciones tipo A se conectan a los dos anillos; en tanto que las estaciones clase B, que son más económicas, sólo se conectan a uno de los anillos. Dependiendo de la importancia que pueda tener la tolerancia a fallos, una instalación puede optar por seleccionar la estación de clase A o B, o una mezcla de ellas.

### 3.3.5 Distribución física de la red del INE

La red del Instituto Nacional de Ecología está distribuida en una torre de 10 pisos (con cuatro niveles cada uno) y las siguientes áreas adicionales: mezzanine planta baja, mezzanine planta alta, anexo mezzanine, centro de cómputo, auditorio y anexo almacén.

De acuerdo a esta distribución y a las necesidades de interconexión se estableció la siguiente distribución de puertos de red de acuerdo a la localidad.

Área	No. de puertos
Mezzanine planta alta	96
Mezzanine planta baja y anexo mezzanine	120
Anexo almacén	12
Torre pisos 1-2	72
Torre pisos 3-6	144
Torre pisos 7-10	144
<b>Total de puertos</b>	<b>588</b>

**Tabla 1.** Distribución de puertos de red

## Topología

La topología de la red institucional está basada en un anillo FDDI de 100 Mbps interconectando redes locales Ethernet de 10 Mbps. El anillo de fibra óptica se instaló de tal manera que pasa por todas las áreas y recorre por la parte central de la torre. Los concentradores están distribuidos de tal forma que se evite el tráfico innecesario en el uso de la red. La distribución de los concentradores está dada de la siguiente forma:

Concentrador	Configuración	Alcance
MMAC-8	5 TPMIM-34 1 EMME-6 pto. FDDI 120 ptos. Ethernet	Mezzanine planta baja y anexo
MMAC-8	5 TPMIM-34 1 EMME-6 pto. FDDI 96 ptos. Ethernet	Mezzanine planta alta
MicroMMAC-8	1 BRIM pto. FDDI 12 ptos. Ethernet	Anexo almacén
MMAC-8	4 TPMIM-34 1 EMME-6 pto. FDDI 72 ptos. Ethernet	Torre pisos 1-2 y anexos
MMAC-8	5 TPMIM-34 1 EMME-6 pto. FDDI 144 ptos. Ethernet	Torre pisos 3-6
MMAC-8	5 TPMIM-34 1 EMME-6 pto. FDDI 144 ptos. Ethernet	Torre pisos 7-10

Tabla 2. Distribución de concentradores

La figura 6 muestra un diagrama lógico general de la red del INE.

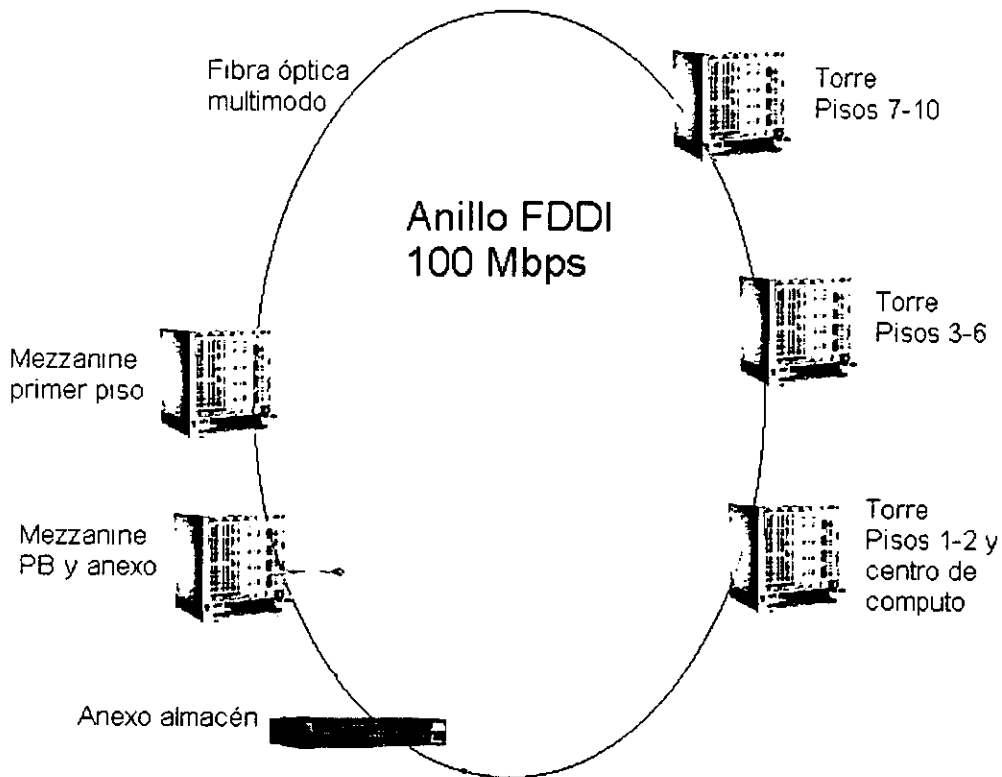


Figura 6. Diagrama lógico general de la red del INE.



### 3.4 Equipo y software disponible actualmente

Concentrador CABLETRON MMAC-M8FNB (Multi Media Acces Center-Flexible Network Bus)

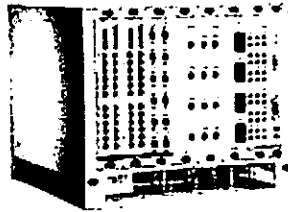


Figura 7. MMAC-M8FNB 8 ranuras MMAC con bus de red flexible con bandeja de ventilador removible.

Se cuenta con 5 concentradores **CABLETRON MMAC-M8FNB** con 8 ranuras para tarjetas **MIM** para diversas aplicaciones y velocidades, estos concentradores se encuentran protegidos por un compartimento de acero llamado *rack*. Entre las características de estos concentradores podemos mencionar las siguientes:

- El concentrador MMAC-M8FNB tiene capacidad para un módulo administrador/repetidor y para más de 7 módulos de interfase media (MIM's *Media Interface Modules*). Usando MIM's de alta densidad, este chasis puede soportar más de 183 puertos. Entre las características del MMAC-M8FNB encontramos ventilador removible y una fuente de poder redundante.
- El M8PSM-E con aumento del módulo de la fuente de poder (*Enhanced Power Supply Module*). Este M8PSM-E ofrece mayor funcionalidad en la fuente de poder, incluyendo ventiladores internos de enfriamiento, aumentando las capacidades de administración remota, puntos de prueba del voltaje para precisar el monitoreo de los voltajes de la fuente de poder, así como salida de 450 watts.

- La característica del módulo M8PSM-E es la de tener leds indicadores de diagnóstico. Cada unidad tiene una línea individual de corriente alterna por conexión para separar los circuitos. Los puntos de prueba de voltaje están incluidos para dar más precisión en el monitoreo de los voltajes de la fuente de poder.
- El M8PSM-EDC (*Enhanced DC Power Supply Module*). Es la más nueva fuente de poder para el MMAC-M8FNB incluye todas las características y funciones del M8PSM-E 36 a 72 Volts de entrada de corriente directa. La entrada de corriente directa con detección de polaridad, un timbre interno (dentro de la fuente de poder) si la polaridad es invertida, alertando al usuario para que tome las medidas correctas.

La distribución de las tarjetas y de los concentradores dentro del edificio del INE se muestran en la siguiente tabla:

Concentrador	Rack en que se encuentra	Ranuras		Tarjetas	Tarjetas	Tarjetas
		En uso	vacías	TPMIM-34	EMM-E6	FDMMIM-24
1	Centro de Cómputo	6	1	4	1	1
2	MEZANINE	7		6	1	
3						
4	NIVEL-16	5	2	4	1	
5	NIVEL-31	6	1	5	1	

Tabla 3. Distribución de tarjetas y de concentradores

Características básicas de las tarjetas contenidas en los concentradores:

#### **Tarjetas TPMIM-34**

Son del tipo 10-BASE-T ETHERNET; lo que indica que trabaja a una velocidad de 10 Mbps, transmiten en banda base y utilizan como medio de transmisión UTP (*Unshield Twisted Pair* - Par Trenzado sin Pantalla o máscara) categoría 5.

- Permite especificaciones 10Base-T de IEEE
- Detección automática de polaridad y corrección
- Para 12 y 24 puertos

#### **Tarjetas EMM-E6**



**Figura 8.** EMM-E6

Para administración e interfaz de FDDI-Ethernet con puertos A-B. Son las que sirven de interfaz entre los concentradores y el anillo de fibra óptica, Por lo tanto se encuentran en cada uno de los concentradores que están distribuidos a través del edificio del Instituto.

- Plena capacidad de operación a todo lo ancho de banda
- Administración mediante SNMP y SMT

- Puertos de consola y módem
- Interruptor óptico de derivación
- Memoria programable EPROM tipo *flash* permite actualizar fácilmente los módulos
- Módulos de administración para redes FDDI y un módulo de puente Ethernet a FDDI

#### Tarjeta FDMMIM-24

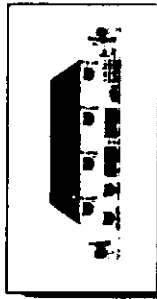


Figura 9. FDMMIM-24

- Soporta cable par trenzado sin malla (UTP)
- Conecta a FDMMIM por vía FNB bus
- Provee conectividad FDDI para más de 4 estaciones de trabajo por medio de cable UTP

Permite conectar a los dos servidores principales a una velocidad de 100 Mbps, ya que estos son los encargados de administrar el *software* necesario a las estaciones de trabajo.

### Concentrador microMMAC-22E

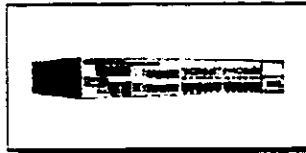


Figura 10. MicroMMAC-22E

- 12 puertos Ethernet 10baseT
- Con conectores RJ45 y RJ71
- Capacidad de administración vía SNMP y monitoreo distribuido
- Leds para diagnóstico visual

#### 3.4.1 Comunicación con el exterior

Para lograr nuestra comunicación con el exterior (Internet) contamos con uno de los ruteadores CISCO de la serie 4000.

#### Ruteador CISCO 4000-M



Figura 11. Ruteador CISCO 4000-M

Los ruteadores de la serie Cisco 4000 son plataformas modulares muy rentables que reducen los costos y la complejidad de las redes al agregar múltiples redes de área local (LAN) en una sola red multiprotocolo. La serie de ruteadores Cisco 4000 amplía las funciones de seguridad mediante el filtrado de paquetes entre LAN y cuenta con las funciones de reserva de ancho de banda y rendimiento necesarios para ejecutar aplicaciones avanzadas, como acceso LAN a Modo Asíncrono de Transferencia (ATM)<sup>18</sup>, conmutación enlace de datos *-data link-* de IBM (DLSw), red avanzada de Igual a Igual (APPN)<sup>19</sup>, y videoconferencia.

El Cisco 4000-M cuenta con memoria *Flash* para almacenar el potente Sistema Operativo de CISCO para trabajos en *Interred -Internetwork Operating System-* (Cisco IOSTM), así como con ranuras para módulos de procesador de Red (NPM) opcionales tarjetas individuales extraíbles empleadas para conexiones de red externas, como por ejemplo Ethernet, Token Ring, Fast Ethernet, ATM, Interfaz de Datos Distribuidos mediante Fibra Óptica *-Fiber Distributed Data Interface-* (FDDI), Interfaz en Serie de Alta Velocidad *-High-Speed Serial Interface-* (HSSI), Red Digital de Servicios Integrados (RDSI), Interfaz de Servicios Básicos *-Basic Rate Interface-* (BRI) e Interfaz de Servicios Primarios *-Primary Rate Interface* (PRI), E1/T1 serie, y serie de alta densidad y baja velocidad. — — —

El ruteador básico CISCO 4000-M está diseñado con un procesador Motorola 68030/40 de 40 MHz para conectividad LAN y WAN de menor densidad y velocidad. Es ideal para variados entornos de oficina regional. Por ejemplo, existen numerosas oficinas multiregionales que cuentan con una mezcla de tráfico LAN y de legado y que desean conectar a servidores o anfitriones *-hosts-* mainframe situados en ubicaciones centrales. El modelo Cisco 4000-M puede convertir los protocolos de legado a protocolos IP, priorizar el tráfico, y ofrecer conectividad Interfaz de Servicios Básicos (BRI) RDSI.

---

<sup>18</sup> ATM. *Asynchronous Transfer Mode*

<sup>19</sup> APPN. *Advanced Peer-to-Peer Networking*

### Características y beneficios

- Procesador Motorola 68EC030 de 40 MHz
- Tres ranuras para añadir módulos de procesador de red (NPM) (véase tabla 4)
- 8 MB de DRAM principal (ampliable a 32 MB), 4 MB de Flash del sistema, 4 MB de boot Flash, y 4 MB de DRAM compartido
- Completa gama de funciones del *software* Cisco IOS

	Número de módulos de procesador de red permitidos	
	Cisco 4000-M	Número de producto
1 puerto Ethernet	3	NP-1E
2 puertos Ethernet	3	NP-2E
6 puertos Ethernet	--	NP-6E
1 puerto Token Ring	3	NP-1RV2
2 puertos Token Ring	3	NP-2R
1 puerto Fast Ethernet	--	NP-1FE
1 puerto ATM DS3/E3	--	NP-1A-DS3, NP-1A-E3
1 puerto ATM OC-3	--	NP-1A-MM, NP-1A-SM
1 puerto Interfaz de Distribución de Datos mediante Fibra Óptica (FDDI)	1	NP-1F-D-MM, NP-1F-S-M, NP-1F-D-SS
1 puerto Interfaz en Serie de Alta Velocidad HSSI)	--	NP-1HSSI (1H 1997)
1 puerto Ch/ PRI RDSI(T1 o E1)	1	NP-CT1, NP-CE1U, NP-CE1B
4 puertos BRI RDSI	2	NP-4B
8 puertos BRI RDSI	1	NP-8B
2 puertos serie	3	NP-2T
4 puertos Serie	3	NP-4T
2 puertos serie y 16 puertos A/S	--	NP-2T/16S
4 puertos G.703	3	NP-4GB, NP-4GU

Tabla 4. Módulos de procesador de red del router 4000-M

### 3.4.2 Equipo de cómputo y software instalado

#### 2 Servidores SUN SPARC station 20 cada una con:

2 procesadores, 9 Gigabytes de disco, 128 Mb de memoria RAM, y una unidad de cinta de 8 mm para respaldos. Cuentan con sistema operativo SunOS 5.5 (Solaris 2.5). Estos dos servidores son los que tienen instalados la mayoría del *software* y son los encargados de proveer del mismo a las estaciones de trabajo a través de NFS. Las utilerías y programas de aplicación están repartidas en ambos servidores, para compensar la carga de trabajo, de la siguiente manera:

#### Servidor 1

- Arcinfo, Siore y Arcview

Son sistemas de información geográfica utilizados para la realización de mapas de distintos tipos y clasificaciones.

#### Servidor 2

- Sistema manejador de bases de datos relacionales **ORACLE**

Es el manejador en el cual reside la base de datos institucional y del cual se tomarán los datos necesarios para presentarlos en la Intranet.

#### En ambos servidores

- Compilador GCC versión 2.7.2.1

Es el compilador de C++ que está instalado en los servidores.

- Interprete de PERL 5.004

Es un lenguaje de programación para procesamiento, manejo de texto, extracción y reporte de información.



- GNU Make versión 3.74

Utilería creada por GNU para el mantenimiento, actualización y regeneración de programas y archivos relacionados. Trabaja de igual forma que el *make* del UNIX estándar.

- GD versión 1.3

Librería gráfica para la creación de imágenes en formato GIF de una manera rápida.

- GDBM 1.7.3

Es una serie de rutinas para bases de datos que utilizan manejo de llaves y que trabajan de igual forma que las rutinas de *dbm* del UNIX estándar.

#### **5 estaciones de trabajo SUN SPARC station 20 cada una con:**

1 procesador, 2 Gigabytes de disco, 64 Mb de memoria RAM. Cuentan con sistema operativo SunOS 5.4 (Solaris 2.4) con sistema operativo básico ya que los servidores son los que les administran de programas, sistemas y utilerías.

#### **8 servidores HP NetServer para la red Novell.**

- Netware 4.1

Sistema operativo de red utilizado para la red Novell.

#### **Computadoras personales**

500 computadoras personales con procesador Pentium II MMX y procesador 486 con el siguiente *software*:

- NetScape versión 4.04

Es el navegador o paginador que está instalado en todas las computadoras del INE y el cual servirá de interfaz única para el acceso a la Intranet.

- Windows 95

Sistema operativo utilizado por la gran mayoría de las computadoras del INE.

- Windows 3.11

Sistema operativo utilizado por las computadoras personales de escasos recursos de *hardware*.

Además de contar con *software* propio de acuerdo a las necesidades de cada uno de los usuarios. Podemos adelantar que gran parte del *software* instalado en los servidores SUN, exceptuando los sistemas de información geográfica, serán utilizados en la construcción de la Intranet del INE, esencialmente para compilar las herramientas que se van a utilizar en el desarrollo de este proyecto.

## CAPÍTULO IV

### SELECCIÓN DE HERRAMIENTAS DE DOMINIO PÚBLICO PARA LA CONSTRUCCIÓN DE UNA INTRANET

---

#### 4.1 Software de dominio público

Se conoce como *software* de dominio público o *freeware*, a los programas gratuitos que empresas o personas pueden utilizar libremente, sin necesidad u obligación de pagar por ello. Muchos creadores de programas distribuyen sus aplicaciones gratuitamente en la red. ¿Qué piden a cambio? De todo: postales, una nota de agradecimiento, una liga a su página, etc. Aunque parezca increíble hay muy buenas cosas en este mundo cibernético que todavía son gratis.

El *software* de dominio público, como cualquier otro programa es seguro siempre y cuando se obtenga de fuentes confiables. La gran mayoría de los programas *freeware*, han sido extensivamente probados. Cabe mencionar que los programas de evaluación, conocidos como versiones BETA (también ALFA) casi siempre son versiones gratuitas de programas que serán comerciales y frecuentemente tienen "*bugs*" (fallas internas) que pueden producir algunos daños, aunque en el 99% estas fallas se limitan a funcionamiento errático del programa. Las versiones BETA son liberadas para que los usuarios prueben y reporten dichas fallas. No se deben confundir las versiones BETA con el *freeware* real (aunque también hay versiones beta de futuro *freeware*).

Normalmente, debido a que los autores no obtienen remuneración económica por su trabajo, no ofrecen ninguna clase de garantías o soporte técnico. Sin embargo, la mayoría de los programas *freeware* disponibles comúnmente, reúnen buenas características de calidad y funcionamiento pero nunca está de más tomar algunas precauciones.

A diferencia de los programas comerciales, la existencia de soporte técnico para el *freeware* depende absolutamente de su creador, es decir, que está en función de que su autor esté dispuesto

a ofrecer tal servicio. Algunos programas *freeware*, si bien no tienen costo, el servicio de soporte técnico puede ser proporcionado bajo esquemas comerciales.

En términos generales, los programas *freeware* no pueden competir con los comerciales porque, como es razonable, el autor *freeware* no vive directamente de sus programas, por lo que no siempre dispone del tiempo y recursos requeridos para producir programas de mayor calidad. Sin embargo, existen muchos programas *freeware* que superan en calidad y funcionalidad a otros similares cuyo costo puede ser elevado.

## 4.2 El proyecto GNU y el software libre

GNU es el nombre de uno de los proyectos de *software* más ambiciosos de la historia. De hecho se ha convertido en el paradigma del *software* libre. Además de ofrecer toda una gama de programas prácticamente gratis, es casi unánimemente reconocida la alta calidad de los mismos, lo cual contradice la idea general de que la calidad es proporcional al precio.

Antes de continuar, es preciso aclarar que la palabra "libre" se refiere a la libertad, no al precio. Al adquirir *software* libre se adquieren automáticamente dos libertades específicas:

- Copiar un programa y compartirlo con quien sea.
- Modificar un programa, pues se tiene acceso total al código fuente.

Si uno se toma la primera libertad con programas comerciales, estaría ejerciendo lo que en el medio se conoce como "piratería", y podría ser sancionado por la ley. La segunda libertad no es posible, ya que muy rara vez se provee el código fuente junto con el *software* comercial. Esta segunda libertad es una fuente de riqueza muy productiva. Los usuarios pueden aportar sugerencias, reportes de errores o problemas (*bugs*), y hasta código fuente original. Esta es la razón de que rara vez persistan *bugs* en programas libres, y a veces se logran resultados excepcionales.

Sin embargo, *software* libre no es sinónimo de *software* de dominio público. Todo programa que se derive de *software* libre debe ser igualmente libre, lo cual implica la restricción a utilizar código fuente libre para construir programas propietarios. Las libertades, restricciones y otras cuestiones legales se detallan en la Licencia Pública General (GPL) de GNU. Este documento se incluye en todos los paquetes del proyecto GNU. Existe una licencia alternativa, la Licencia Pública General de Bibliotecas (GLPL) que permite que algunas bibliotecas libres se ligen a ejecutables propietarios bajo ciertas condiciones.

Es recomendable que el *software* original se registre. Sin embargo el *copyright* tradicional contradice las libertades del *software* libre. Un nuevo concepto, llamado *copyleft*, además de asegurar que se respete la autoría del código fuente, obliga a todo usuario del mismo a mantener los derechos y libertades ya mencionadas. Esto no implica una prohibición a cobrar por redistribuir *software* libre, siempre y cuando el cliente disfrute de las mismas libertades, y se le proporcione una copia de la GPL.

Richard Mathew Stallman, el creador y líder del proyecto GNU, es también uno de los mejores y más creativos programadores vivos. En su opinión modificar partes o el todo de un programa es tan natural y productivo para un programador como respirar, y así debería ser de libre. Además, el acto fundamental de amistad entre programadores es el compartir programas. Antes de consagrarse al proyecto GNU, Stallman participó en la creación de sistemas operativos, compiladores, depuradores, etc., en el Laboratorio de Inteligencia Artificial (AIL) del Instituto Tecnológico de Massachusetts. Hombre de firmes convicciones, repudió cualquier intento de restringir el uso y la distribución de *software* desarrollado por él, de manera que renunció al AIL para evitar cualquier excusa legal que le impidiera desarrollar *software* libre.

Así, Stallman creó la Fundación de *Software* Libre (FSF)<sup>20</sup> con el fin de desarrollar un sistema operativo totalmente libre y eliminar la necesidad de usar *software* propietario. Este sistema sería compatible con UNIX, pero lo superaría en varios aspectos. De ahí surgió el nombre del proyecto, que es un acrónimo recursivo: **GNU's Not Unix.**

---

<sup>20</sup> FSF, *Free Software Foundation*

El núcleo del futuro sistema operativo, que aún no está en circulación, está siendo desarrollado por un equipo compacto contratado por la FSF para ese fin. Este sistema ha estado evolucionando por más de 20 años, en contraste con el tiempo que llevó a dos programadores de los laboratorios AT&T Bell desarrollar el UNIX primigenio: menos de un año.

Afortunadamente, muchos de los programas, herramientas de desarrollo y demás utilerías, han estado disponibles<sup>21</sup> al público desde hace varios años. Entre éstas se encuentra el editor de textos multiusos Emacs, el compresor de archivos gzip, depuradores a nivel fuente, y los compiladores para C, C++, C-objetivo, Smalltalk, y Ada.

Estos programas han sido desarrollados en gran parte gracias a la cooperación de voluntarios, principalmente del medio académico, comunicados a través de Internet. El acceso libre al código fuente ha permitido que usuarios capacitados puedan estudiar el código, mejorarlo y enviar las correcciones a los autores originales, para que éstos puedan incluirlas en versiones posteriores. Esta política también ha permitido que los programas sean portados a otras plataformas, inclusive aquellas que usan sistemas operativos incompatibles con UNIX. De esta manera se evita el despilfarro de recursos humanos que implicaría duplicar sistemas, simplemente se mejoran y adaptan.

Todo esto lleva a que el *software* libre sea rápido, poderoso y más portable que sus contrapartes propietarias. Evoluciona más rápidamente porque los mismos usuarios son libres de mejorar el código. Dadas las limitadas restricciones en su distribución, el *software* libre de alta calidad es ampliamente adoptado en todo el mundo, tanto por empresas como por universidades, cuyos estudiantes de ciencias de la computación se benefician al estudiar código de sistemas de primer nivel. Si aceptamos que la restricción del copiado es la que brinda mayores ganancias monetarias, es difícil evadir la pregunta: si el *software* fuera prácticamente gratis, ¿de qué viviría la industria del *software*? Los vendedores simplemente podrían dedicarse a vender otra cosa, pero ¿morirán de hambre los programadores? Para Stallman la respuesta es simple: No morirán de hambre, simplemente ganarán menos. Entre las alternativas laborales que menciona destacan:

---

<sup>21</sup> El software esta disponible en <ftp://prep.ai.mit.edu/pub/gnu>

- Las empresas de *hardware* pagarían por portar sistemas operativos a *hardware* nuevo
- La venta de capacitación y servicios de mantenimiento sería redituable

Algunas empresas de *hardware* han apoyado económicamente a la FSF para portar las utilerías de desarrollo GNU a los nuevos *chips*, aunque el sistema operativo básico del proyecto aún no esté listo. Ya existen empresas que se mantienen exclusivamente de prestar servicios de soporte técnico en *software* libre. Tal es el caso de Cygnus Support<sup>22</sup>, fundada en 1989, cuya estrategia ha sido ahorrarle a las corporaciones la necesidad de utilizar sus propios equipos de ingenieros y programadores para adaptar *software* libre a sus necesidades. El *software* libre se ha convertido así en una opción atractiva para las empresas, pues sus gastos en *software* se reducen al pago del soporte.

La FSF no provee soporte técnico. En cambio, mantiene un directorio de compañías o individuos que prestan estos servicios comercialmente. No es difícil prever que a estas compañías les va mejor económicamente que a la misma FSF. Sin embargo, esto no es motivo de preocupación para los miembros de la FSF, pues no es una organización lucrativa; su único objetivo es poner a punto el sistema GNU y accesorios. Obtiene sus ingresos de la distribución de *software* libre y manuales impresos, pero principalmente de donaciones deducibles de impuestos en EU.

Hemos visto que el concepto de *software* libre no es sólo un sueño idealista. Proporciona beneficios prácticos a usuarios y programadores. Richard Stallman va más allá: el proyecto GNU es un proyecto social que por medios técnicos cambiará a la sociedad.

### 4.3 Selección de un servidor de Web

WWW es la parte de Internet que más usuarios nuevos atrae y como resultado el número de compañías interesadas en proveer servidores para HTTP ha crecido. En la actualidad hay más de 30 productos diferentes para toda una gama de sistemas operativos.

---

<sup>22</sup> La dirección en Internet es <http://www.cygnus.com/>

Todo servidor debe ser compatible con HTTP. Sin embargo, cada proveedor incluye características únicas a su servidor para tratar de hacerlo más atractivo en un mercado que empieza a ser ya bastante competido. Algunos servidores serán más simples que otros, otros más rápidos, otros gratuitos, otros más fáciles de instalar, otros generarán mejores estadísticas de uso, etc.

#### 4.3.1 ¿Cómo se eligió el servidor de Web?

¿Cuál debe elegirse? La respuesta depende de muchos factores: dinero disponible para comprarlo, sistema operativo, si se requieren transacciones seguras, etc.

Para elegir el servidor para la construcción de la Intranet se investigó en Internet el uso, análisis y comparaciones de la información de servidores presentada en la página del **Netcraft Web Server Survey**<sup>23</sup> así como también se recopiló la información sobre los servidores que existen actualmente en el mercado en el **Web Server Compare**<sup>24</sup>.

Los resultados de una evaluación de eficiencia entre los servidores más comunes de HTTP se encuentran en el **Netcraft Web Server Survey**, que es un examinador de *software* de servidores de Web usados en las computadoras que están conectadas a Internet. Se encargan de coleccionar y ordenar todos los servidores que proveen servicio de HTTP que encuentra, y automáticamente se cuantifica cada uno de ellos al pedir una solicitud de HTTP al servidor. En septiembre de 1998 se recibieron respuestas de 3,156,324 sitios. A continuación se muestra una gráfica de crecimiento de los cuatro principales servidores de Web.

---

<sup>23</sup> <http://www.netcraft.com/Survey/>

<sup>24</sup> La información se puede consultar en <http://webcompare.internet.com/>



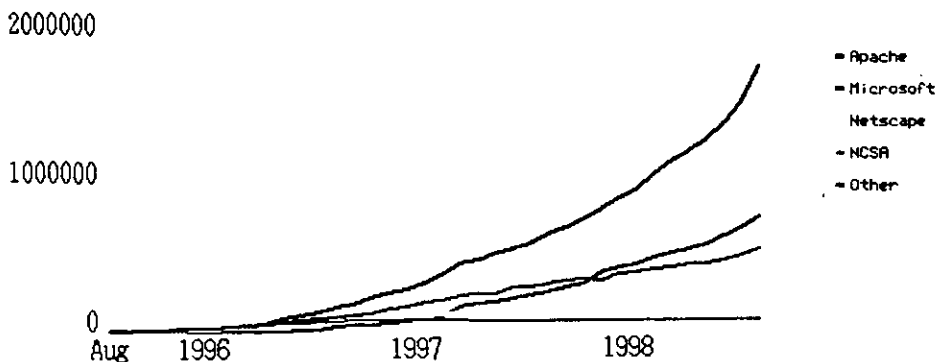


Figura 12. Crecimiento de los servidores Web en Internet. Agosto de 1995 – septiembre de 1998<sup>25</sup>

### Un mercado dominado por el software de dominio público

Para los usuarios que requieren más, Microsoft y Netscape Communications son opciones muy populares, aunque recientemente Apache ha tenido todo el éxito. Esto no es sorprendente, pues Netscape es el fabricante del navegador de WWW más utilizado, lo que le da cierta ventaja, pues entre ambos, navegador y servidor, Netscape podría implementar funciones aún no estandarizadas. Su característica más notable es el uso de criptografía de llave pública y DES para realizar transacciones en la red.

En general, los servidores comerciales tienden a hacer más fácil el proceso de instalación y mantenimiento del servidor mediante interfases de uso ameno que hacen posible que la instalación no necesite hacerla un experto –comparado con la tarea de modificar archivos de configuración a través de un editor en los servidores gratuitos. Además, por lo regular, ofrecen soporte técnico especializado. Otra ventaja es utilizar UNIX, ya que podemos elegir entre una amplia gama de servidores. En cambio, los sistemas operativos "raros" tendrán pocas opciones o ninguna. La decisión final de qué servidor es el mejor para una organización dada depende de muchos factores: precio, tipo de transacciones que se realizarán con el servidor –transferencia de

<sup>25</sup> La información se consultó en <http://www.netcraft.com/Survey/>

información crítica, por ejemplo-, cantidad de información a transferir, tipo de sistema operativo, etc.

Servidor	Septiembre 1998	Porcentaje	Cambio
Apache	1,636,674	51.85	1.50
Microsoft-IIS	697,074	22.05	-0.34
Netscape-Enterprise	149,398	4.73	-0.15
Rapidsite	70,716	2.24	-0.06
NCSA	65,666	2.08	-0.20
WebSitePro	60,946	1.93	0.03
Thttpd	55,883	1.77	-0.18
Stronghold	49,584	1.57	-0.10
WebSTAR	43,293	1.37	0.04
Netscape-Commerce	42,945	1.36	-0.10

Tabla 5. Los servidores más utilizados por los sitios de WWW<sup>26</sup>

Notoriamente y por todo lo mencionado anteriormente, la mayor parte de los sitios de WWW utilizan servidores del dominio público. Según estimaciones, el servidor Apache se disputa más del 50% del mercado como se muestra en la tabla 5. Es robusto y ofrece características que satisfarían las necesidades de la mayor parte de los sitios no comerciales: es rápido, ofrece diversos niveles de acceso a las páginas a través de *passwords* y de nombre de dominio, por ejemplo.

Por tales motivos y porque contamos en nuestros servidores con las herramientas necesarias para la compilación y buen desempeño de este servidor, así como la gran ventaja de tener un *software* gratuito con los archivos fuente, **APACHE** fue el servidor elegido como servidor de Web.

Apache ha sido el servidor que más rápidamente ha crecido en popularidad. Es también gratuito y está siendo desarrollado por el Proyecto Apache<sup>27</sup>, cuyo objetivo es la creación y mantenimiento de un servidor gratuito de WWW compatible con el estándar HTTP.

<sup>26</sup> La información se consultó en <http://www.netcraft.com/Survey/>

<sup>27</sup> <http://www.apache.org/>

## ¿Qué es APACHE?

Apache fue basado originalmente en el código e ideas encontradas en el servidor más popular de HTTP de la época, NCSA httpd 1.3 (cerca de 1995). Desde que fue desarrollado es un sistema superior que puede rivalizar (y probablemente sobrepasar) con casi cualquier otro servidor de HTTP basado en UNIX en términos de funcionalidad, eficiencia y velocidad.

Desde su inicio, ha sido completamente reescrito, e incluye algunas nuevas características. Apache es, desde enero de 1998, el servidor más popular de la WWW en la Internet, de acuerdo con **Netcraft Survey**.

Apache fue creado por el interés de un grupo de proveedores de WWW y una parte por programadores de HTTPD que abandonaron el proyecto ya que no se comportaron como se esperaba. Apache es un esfuerzo enteramente voluntario, completamente fundado por sus miembros y no por ventas comerciales.

El nombre de Apache es un nombre el cual fue tomado de la frase inglesa "*A PAtCHy server*" la cual significa parche o remiendo ya que Apache fue basado en códigos existentes y una serie de parches.

Apache ha demostrado ser sustancialmente más rápido que muchos otros servidores gratuitos que se puedan encontrar en Internet. Aunque ciertos servidores comerciales han alegado sobrepasar la velocidad de Apache (pero no han demostrado que cualquiera de estas pruebas de desempeño son una buena manera de medir la velocidad de los servidores), es mejor tener un servidor de mayor velocidad gratis que un servidor extremadamente veloz que cuesta miles de dólares. Apache corre en sitios donde se alcanzan millones de visitas por día, y no han reportado dificultades en el desempeño.

Ha sido probado completamente por desarrolladores y usuarios. El grupo de Apache mantiene rigurosos estándares antes de liberar nuevas versiones del servidor, y corre sin problemas en una tercera parte de todos los servidores que actualmente se encuentran en Internet. Cuando hay fallas son mostradas de inmediato, y se liberan parches y nuevas versiones tan pronto como sea posible.

## 4.4 Selección de un lenguaje de programación para CGI's.

### 4.4.1 CGI. Common Gateway Interface

Uno de los elementos más potentes de HTML es la *Common Gateway Interface* (CGI). HTML contiene etiquetas que le permiten al usuario la elaboración rápida de documentos de aspecto profesional. Pero sería fabuloso que HTML tuviera etiquetas mediante las cuales los usuarios pudieran ejecutar programas con un navegador Web, procesar los resultados y presentarlos en un documento HTML. ¿Se puede hacer eso? La respuesta es no y sí. En vez de tratar de hacer un lenguaje de programación propio, habiendo tantos tan buenos, los diseñadores del estándar HTML hicieron algo mayor al diseñar la *Common Gateway Interface*.

Los diseñadores de HTML crearon una entrada en el estándar HTML que permite a los programadores ejecutar un programa escrito en cualquier lenguaje que deseen usar. Esto se llama *Common Gateway Interface*. Un *gateway* es una conexión con el sistema operativo externo. CGI proporciona a los programadores un modo de que las páginas Web en HTML puedan ejecutar programas externos y presentar los resultados.

#### El proceso detrás del CGI

La acción de llamar un programa CGI desde un navegador Web es muy sencilla para el usuario, lo cual es uno de los principales atractivos del *Common Gateway Interface*. Sin embargo, desde la perspectiva del programador, el proceso es un poco más complicado. Existen varios pasos a seguir para que un programa CGI funcione adecuadamente:

- 1) El usuario llama a un programa CGI haciendo *click* sobre un vínculo u oprimiendo un botón.
- 2) El navegador solicita autorización al servidor de Web para ejecutar el programa CGI.
- 3) El servidor Web revisa la configuración y los archivos de acceso para asegurarse que el solicitante tiene permitido el acceso al programa CGI.
- 4) El servidor Web se asegura que exista el programa CGI.

- 5) Si existe el programa, éste se ejecuta.
- 6) Cualquier resultado producido por el programa CGI se devuelve al navegador Web.
- 7) El navegador Web despliega el resultado.

El navegador Web puede transferir información al programa CGI de varias maneras, y el programa puede devolver los resultados con etiquetas HTML incluidas, como texto sencillo o como una imagen. El navegador Web interpreta los resultados de la misma manera como lo hace con cualquier otro documento. Con un *click* del ratón, proporciona una herramienta muy poderosa para ejecutar prácticamente cualquier programa, y permite a los programadores el acceso a cualquier base de datos externa, que proporcione una interfaz de programación. Además, en vez de diseñar un *front end* para una base de datos específica, el *front end* queda igual, sin importar el tipo de base de datos que se utilicen abajo.

Para que los programas CGI puedan funcionar en un sistema, el servidor Web se debe configurar de manera que permita el acceso a éstos. Algunos administradores de sistemas consideran el acceso con CGI como un posible riesgo de seguridad, y niegan a todos los usuarios la posibilidad de utilizarlo. En otros sistemas, el acceso con CGI se otorga a ciertos usuarios de manera exclusiva, y sólo los programas en CGI escritos por usuarios autorizados puede ejecutarlos el servidor.

### **Transferencia de datos a través de CGI**

CGI es un *gateway* de doble sentido. Los datos pueden transferirse al programa CGI para su procesamiento y los programas CGI también pueden devolver información al navegador Web para su presentación. Los programas CGI proporcionan a los programadores un modo de hacer páginas de Web interactivas. La información introducida por el usuario puede afectar el comportamiento del programa CGI, y los resultados devueltos por el programa son resultado directo de lo que introduce el usuario.

Un navegador Web carga un documento HTML y lo despliega. En algunos casos se ejecuta un programa CGI cuando se carga la página, y los resultados se despliegan como parte de la página

Web. En otros casos, el usuario oprime un botón o hace *click* en un vínculo para activar un programa CGI y los resultados se despliegan como parte de otra página Web.

Los programas CGI no tienen que utilizarse como programas tradicionales. Por ejemplo, un contador en una página Web que indica cuántos visitantes la han abierto puede ser un programa CGI. A menudo, los contadores son programas CGI que devuelven números en forma de imágenes en formato GIF, lo cual no se espera de un programa tradicional.

#### 4.4.2 ¿Cómo se eligió el lenguaje de programación para CGI's?

Existen muchos lenguajes disponibles para escribir programas CGI. No hay un sólo lenguaje que satisfaga las necesidades de cada programa, pero si se piensa en los lenguajes de programación como conjuntos de herramientas, puede seleccionar la herramienta apropiada para la tarea a realizar.

Los siguientes son los tres principales lenguajes utilizados para la programación CGI:

- Si el programa CGI se utiliza fundamentalmente para efectuar comandos del sistema, un *script shell* de UNIX puede cumplir el objetivo.
- Un programa ejecutable escrito en un lenguaje como C es una buena opción si la velocidad de ejecución o la seguridad del código fuente es un aspecto importante.
- Si la facilidad de operación o un conjunto enriquecido de funciones es lo que usted desea, *Perl* será la mejor elección.

Cada herramienta de programación tiene muchas cosas que ofrecer, todo depende del tipo de trabajo a realizar, para elegir el tipo de herramienta. Todos esos lenguajes se utilizan para crear programas CGI como se muestra en las siguientes secciones.

## ¿Por qué se decidió por PHP?

PHP cubre todas las características de los principales lenguajes de programación de CGI's mencionadas con anterioridad, es un *software* gratuito y trabaja de manera transparente con Apache además de permitir la comunicación a través de CGI's con varios manejadores de bases de datos. PHP es un **pre-procesador** que se incrusta en el *daemon* de apache (httpd) lo cual lo convierte en un poderoso lenguaje de interpretación y es altamente compatible con el servidor de Web Apache.

Es básicamente una aplicación que analizará los archivos HTML, que contienen información "extra" que el PHP interpreta y utiliza para: asignar variables, interactuar con bases de datos, actúa de manera condicional, escribe en archivos, maneja correo electrónico, y mucho más. PHP puede hacer que las páginas Web sean muy interactivas. Quizá la mayor y más significativa característica de PHP es su capacidad de integrar soporte para bases de datos. Escribir páginas Web con soporte para bases de datos es muy sencillo. La siguiente lista de manejadores de bases de datos son actualmente soportados por PHP:

- Oracle
- Adabas D
- Sybase
- FilePro
- mSQL 1.x y 2.x
- Velocis
- MySQL
- Solid
- DBase
- Generic ODBC
- Unix dbm
- PostgreSQL
- Todos los manejadores de bases de datos con una interfase ODBC.

Ahora, con relación a nuestro servidor de bases de datos ORACLE que se tiene instalado en nuestro servidor, PHP tiene soporte "integrado" para el servidor ORACLE. En otras palabras, PHP utiliza su propio lenguaje para simplificar y hacer más accesible la interacción de datos, como es la lectura, modificación, borrado y actualización de las tablas del servidor ORACLE.

Pensamos que el interprete de PHP es una excelente forma de usar el servidor ORACLE y de sacarle más provecho de una manera más amigable a través de un navegador, y en la documentación a la que se hará referencia en el siguiente capítulo se hace una demostración acerca de su funcionalidad.



## *CAPÍTULO V*

### *CASO DE APLICACIÓN: INTRANET DEL INSTITUTO NACIONAL DE ECOLOGÍA*

---

#### **5.1 Delimitación y justificación del proyecto**

Dentro del Programa de Medio Ambiente 1995-2000 se tiene contemplado el desarrollo del Sistema Nacional de Información Ambiental (SINIA).

El SINIA es un sistema diseñado en forma modular, lo cual permitirá efectuar desarrollos independientes que sean capaces de compartir información. De esta forma se podrán mejorar los procesos de acceso y distribución de la información generada por las instancias encargadas de la política en materia de medio ambiente, recursos naturales y pesca, en forma paulatina.

Dentro de este esquema se pretende desarrollar un mecanismo que permita que la información contenida en el Instituto Nacional de Ecología (INE) sea consultada por personas del mismo Instituto de una manera fácil y sencilla. Para esto es necesario contar con un sistema que permita agilizar y controlar la difusión de la información y ofrecer servicios de recuperación, consulta y opinión, así como contar con herramientas, procesos y conocimientos necesarios y suficientes para poder ofrecer esta información a todo el personal dentro del INE.

Lo anterior, conlleva a poner en marcha este instrumento de apoyo automatizado que proporciona información siempre actualizada y confiable que permite tomar las decisiones pertinentes que eleven la calidad de atención y logre ejecutar en los tiempos establecidos las acciones que consoliden el cumplimiento de los compromisos dentro del Instituto.

Esta herramienta tecnológica proporcionará importantes beneficios, ya que dará oportunidad de cumplir con los requerimientos de Ley a la vez que ofertar servicios que permitan interactuar con el personal del INE.

Considerando también que el Instituto Nacional de Ecología está interesado en participar en las tecnologías de punta y tiene claro que debe incorporarse a la tendencia mundial de usar una Intranet como un medio para difundir información e interactuar con usuarios internos se justifica el interés manifiesto del Instituto en la construcción de una herramienta de calidad interna.

La Intranet del INE aprovechará la filosofía y metodología de trabajo de Internet, así como su flexibilidad y eficacia, los niveles de comunicación y productividad se incrementarán considerablemente. La Intranet del INE estará constituida básicamente por un sitio local de Web donde se podrá diseminar información y proveer acceso a aplicaciones y grupos de discusión entre otras cosas. Además ofrecerá herramientas para la distribución de información, es decir, será un vehículo para que la información, que antes residía en sistemas a los que pocos tenían acceso, pueda ser accesada, desde su escritorio, por prácticamente todos los interesados dentro del INE.

El resultado de esto será un ambiente de acceso sencillo a la información y desde cualquier punto del Instituto, sin dejar de tomar en cuenta los niveles de seguridad que se requieren para el manejo de información confidencial o delicada, la cual no podrá ser accesada por todos los usuarios; se espera tener resultados gratificantes ya que un aspecto importante que se tiene que afrontar es introducir esta cultura de trabajo en los empleados del INE.

La Intranet del INE sólo será accedida por personal autenticado con una cuenta y una contraseña, y con ciertos criterios de seguridad en cuanto a quien puede o no ver cierta información además deberá de ofrecer la gran ventaja de utilizar las mismas herramientas de comunicación que se usan para conectarse a Internet y de las ya instaladas como compiladores (GCC) y sistemas más complejos como lo es el manejador de bases de datos ORACLE por lo cual no será necesario instalar de más *software*.

## 5.2 Planteamiento de la necesidad

Actualmente la información que es de interés común para el personal del Instituto Nacional de Ecología es consultada en oficinas y comunicados que son enviados a cada una de las personas interesadas dando como consecuencia perdida de tiempo y desperdicio de recursos del Instituto como es el caso del papel y de toner de impresora.

En un día típico docenas de empleados transfieren gráficos, aplicaciones, documentos, en resumen, información. Por experiencia y observación se ha visto que el flujo de datos es unidireccional. El instituto trabaja con un gran volumen de datos, y en forma jerárquica. Por ejemplo, directores generales, o directores de área, que tienen personas a su cargo, por dar una cifra razonable, aproximadamente diez personas trabajando con él, cada vez que el jefe tiene que pasar información “hacia abajo” a través de la pirámide que es la empresa, tiene que pasársela a todos los empleados bajo su responsabilidad. En este caso, cada director ha de enviar la información en cuestión diez veces. La relación que se establece entre jefes y empleados está desequilibrada, ya que supone un cuello de botella para la transferencia de datos entre personas de la empresa. Es decir si el tiempo que tarda un director en enviar la información a un empleado es  $x$ , habrá un empleado de los diez que recibirá su información al cabo de diez veces  $x$  tiempo, ya que el jefe va enviando los datos de uno en uno.

Otro problema muy común es que las máquinas del INE no están todas totalmente estandarizadas en cuanto a paquetería, es decir, procesadores de texto, visualizadores de gráficos, hojas de calculo, etc. por lo cual es un problema para algunos usuarios poder dar lectura al instante a un documento en un formato que no es soportado por su computadora personal y como consecuencia una perdida de tiempo para el usuario al tratar de conseguir una máquina con mayores capacidades.

## 5.3 Instalando y configurando PHP para UNIX

En esta sección se verán los pasos que se deben de seguir para la instalación de la herramienta que nos permitirá la escritura de CGI's así como la comunicación con la base de datos Institucional para la creación de páginas interactivas en tiempo real.

### 5.3.1 Obteniendo PHP

Para obtener información y la última versión de PHP se consultó el siguiente URL: <http://www.php.net/> que es el sitio Web oficial en Internet y en el cual también podemos encontrar la más reciente documentación, ayuda, ejemplos y solución a posibles *bugs* encontrados.

Algunos requisitos de *software* para la instalación de PHP son:

- Herramientas básicas de UNIX
- Un compilador ANSI C
- Y un Web server (sólo la estructura)

### 5.3.2 Instalación de PHP

La instalación de PHP se hace en 3 sencillos pasos:

#### 1.- Desempacar el archivo de distribución

El archivo fuente es llamado comúnmente **php3xn.tar.gz** y tiene un tamaño aproximado de 997 KB y descompactado necesita un espacio cercano a los 12 MB. Para obtener la estructura y los archivos fuentes se deben seguir los siguientes pasos:

- Lo primero que se necesita es tener un directorio disponible, para este caso se utilizó el directorio **/opt/src**

- Descompactar el archivo de distribución con el siguiente comando:

```
gunzip php-3.0.tar.gz
```

- Al acabar de ejecutarse queda un archivo llamado **php-3.0.tar**
- Para quitar la extensión *tar* se utiliza el siguiente comando:

```
tar xvf php-3.0.tar
```

Automáticamente se creará el directorio **/opt/src/php-3.0** como base de la estructura de archivos y directorios que se muestra en el anexo 1.

## **2.- Configurar PHP**

Se debe utilizar el script llamado *setup* que viene incluido en la distribución, el cual hará una serie de preguntas, referentes a opciones y localización de *software*, y automáticamente correrá el script *configure*.

Como requisitos previos a correr el script *setup* es conocer y tener definida la estructura y localización de los directorios de Apache, esto es porque el script preguntará por el directorio raíz en el cual está Apache y buscará el archivo *httpd.h* a partir de la ruta indicada dependiendo de la versión de Apache que se esté utilizando. El directorio raíz de Apache es el siguiente: **/opt/src/apache\_1.3.0**

Para poder tener el soporte para ORACLE con PHP es el mismo caso, antes debemos saber cual es la ruta donde residen las librerías de ORACLE que utilizará PHP para que lo pueda reconocer. En éste caso las librerías se encuentran en la siguiente ruta: **/oracle/app/oracle/product/7.3.2/lib**. El script *setup* sólo preguntará por el directorio base de Oracle el cual es **/oracle/app/oracle/product/7.3.2**.

El listado de la ejecución del script *setup* se lista en el anexo 2.

### 3.- Compilar e instalar los archivos

Ya que PHP se va a utilizar como módulo de Apache en éste paso se copiarán los archivos apropiados al directorio **src/modules/php3** de la estructura de directorios de Apache. Primero se debe ejecutar el comando *make* para compilar y luego *make install* para que se instalen los archivos.

La ejecución del comando *make* se muestra en el anexo 3.

Al terminar de compilar PHP tendremos como resultado una librería llamada **libmodphp3.a** la cual será utilizada al compilar Apache.

Ejecutando el comando *make install*:

```
# make install
cp libmodphp3.a mod_php3.* php_version.h /opt/src/apache_1.3.0/src/modules/php3;
cp apMakefile.tpl /opt/src/apache_1.3.0/src/modules/php3/Makefile.tpl;
cp apMakefile.libdir /opt/src/apache_1.3.0/src/modules/php3/Makefile.libdir;
cp libphp3.module /opt/src/apache_1.3.0/src/modules/php3
```

**Listado 1.** Ejecución del comando *make install*

Una vez ejecutado éste comando tendremos en el directorio **src/modules/php3** los archivos necesarios para ser utilizados cuando se compile Apache. El siguiente paso ahora es instalar y configurar Apache para que reconozca como módulo a PHP.

## 5.4 Instalando y configurando Apache para UNIX

La instalación de Apache se llevó a cabo en el mismo servidor SUN donde se instaló PHP y en el cual también reside el manejador de bases de datos ORACLE.

### 5.4.1 Obteniendo Apache

Para obtener información y la última versión de Apache se consultó el siguiente URL: <http://www.apache.org/> que es el sitio Web oficial en Internet y en el cual además de obtener la versión actual, se encuentra cualquier liberación de pruebas recientes, ayuda, detalles de páginas espejo y sitios de ftp anónimo para bajar el *software*.

Los archivos fuente por lo regular se distribuyen en un formato *tar* y compactados en un archivo que por lo general se llama *apache\_1.3.x.tar.gz* y tiene un tamaño aproximado de 1.8 MB y descompactado se necesita un espacio poco mayor de los 16 MB, para lograr obtener los fuentes se hacen los siguientes pasos:

- Lo primero que se necesita es tener un directorio disponible, para este caso se utilizó el directorio */opt/src*
- Descompactar el archivo de distribución con el siguiente comando:

```
gunzip apache_1.3.0.tar.gz
```

- Al acabar de ejecutarse quedará un archivo llamado *apache\_1.3.0.tar*
- Para quitar la extensión *tar* se utiliza el siguiente comando:

```
tar xvf apache_1.3.0.tar
```

Automáticamente se crea<sup>28</sup> el directorio `/opt/src/apache_1.3.0` como base y la siguiente estructura de archivos y directorios:

Total 162								
Drwxr-xr-x	8	25	root	512	Jul	9	15:04	.
Drwxrwxrwx	21	1005	root	1024	Jul	9	13:43	..
-rw-r--r--	1	125	root	12650	May	18	19:43	ABOUT_APACHE
-rw-r--r--	1	125	root	811	Jun	1	16:18	Announcement
-rw-r--r--	1	125	root	9720	Apr	14	02:40	CHANGES
-rw-r--r--	1	125	root	19718	May	22	01:25	INSTALL
-rw-r--r--	1	125	root	14545	Mar	25	12:39	KEYS
-rw-r--r--	1	125	root	2848	Mar	31	06:52	LICENSE
-rw-r--r--	1	root	other	20165	Jul	30	17:45	Makefile
-rw-r--r--	1	125	root	20062	May	12	05:52	Makefile.tmp
-rw-r--r--	1	125	root	2046	Apr	1	06:59	README
-rw-r--r--	1	125	root	2898	May	30	06:57	README.NT
-rw-r--r--	1	125	root	10702	May	30	09:24	README.configure
drwxr-xr-x	2	125	root	512	Jun	1	16:25	cgi-bin
drwxr-xr-x	2	125	root	512	Jun	1	16:27	conf
-rwxr-xr-x	1	125	root	35703	May	29	14:45	configure
drwxr-xr-x	3	125	root	512	Jun	1	16:27	htdocs
drwxr-xr-x	2	125	root	2048	Jun	1	16:25	icons
drwxr-xr-x	2	125	root	512	Jun	1	16:27	logs
drwxr-xr-x	10	125	root	1024	Jul	30	17:45	src

Listado 2. Estructura de archivos y directorios de Apache

### 5.4.2 Compilando Apache

La compilación de Apache consiste de tres pasos básicamente:

- En primer lugar seleccionar que módulos de Apache se quieren incluir para el servidor que se va a instalar
- Como segundo paso es crear la configuración de acuerdo al sistema operativo en el cual se va instalar el servidor
- Por último compilar para generar el archivo ejecutable

<sup>28</sup> Este paso se hizo con anterioridad porque PHP necesita conocer la ruta de Apache (`/opt/src/apache_1.3.0/src/modules`) para colocar la librería y los archivos que deben compilarse.



Toda la configuración de Apache se encuentra localizada en el directorio raíz de la estructura de directorios que se creó. Es necesario colocarse en este directorio para seguir los siguientes pasos:

### 1.- Seleccionar en el archivo *Configuration* los módulos que se desean compilar

Descomentar las líneas correspondientes a módulos opcionales que se desean incluir, o también se pueden agregar nuevas líneas que correspondan a módulos adicionales que se deseen incluir. Para este caso no se agregó ningún módulo adicional para PHP ya que esto lo haremos más adelante con el script *configure*.

### 2.- Configurar Apache para el sistema operativo.

Normalmente sólo se debe ejecutar el script *configure* de la forma que se muestra más adelante. Sin embargo, si este proceso falla o se desea un requerimiento especial, incluir una librería o agregar un módulo, se deben de editar las siguientes opciones: *EXTRA\_CFLAGS*, *LIBS*, *LFLAGS*, *INCLUDES*. Para nuestro caso se tuvieron que agregar las opciones *--prefix* y *--activate-module* para indicar el directorio raíz de Apache y la localización de la librería de PHP respectivamente.

Ejecutando el script *configure*:

```
# ./configure --prefix=/opt/src/apache_1.3.0 --activate-module=src/modules/php3/libphp3.a
Configuring for Apache, Version 1.3.0
+ activated php3 module (modules/php3/libphp3.a)
Creating Makefile
Creating Configuration.apaci in src
Creating Makefile in src
+ configured for Solaris 251 platform
+ setting C compiler to gcc
+ adding selected modules
  o php3_module uses ConfigStart/End
+ doing sanity check on compiler and options
Creating Makefile in src/support
Creating Makefile in src/main
Creating Makefile in src/ap
Creating Makefile in src/regex
Creating Makefile in src/os/unix
Creating Makefile in src/modules/standard
Creating Makefile in src/modules/php3
```

Listado 3. Listado de la ejecución del script *configure*

Esto genera un archivo *Makefile* que será usado en el tercer paso. También se crea un archivo *Makefile* en varios directorios, para compilación de programas de soporte adicionales.

### 3.- Compilar ejecutando el comando *make* de UNIX

Como se mencionó anteriormente, todos estos pasos deben de ser ejecutados dentro del directorio raíz. Después de la compilación tendremos un archivo ejecutable llamado *httpd* dentro del directorio *src*.

El listado que produjo el comando *make* se muestra en el anexo 4.

El siguiente paso es instalar el programa y configurarlo. Apache está diseñado para ser configurado y ejecutado desde la misma estructura de directorios donde fue compilado. En caso de ejecutarlo desde otro directorio es necesario copiar los directorios *conf*, *logs* e *icons* al lugar donde se cambio el ejecutable. Para nuestro caso la estructura de directorios quedó de la siguiente manera:

```
/usr/local/etc/httpd/conf
```

```
/usr/local/etc/httpd/logs
```

```
/usr/local/etc/httpd/icons
```

y el archivo ejecutable *httpd* fue puesto en el siguiente directorio:

```
/usr/local/etc/httpd/bin
```

Es necesario editar los archivos de configuración del servidor, esto consiste en poner en habilitado varias opciones de los tres principales archivos de configuración. Por omisión estos archivos están localizados en el directorio *conf* y son los siguientes: *srm.conf*, *access.conf* y *httpd.conf*. Para ayuda de cómo configurarlos hay unos archivos que se encuentran en el directorio *conf* que vienen con la distribución de Apache, estos son llamados de la siguiente forma: *srm.conf-dist*, *access.conf-dist* y *httpd.conf-dist*. Se pueden copiar o renombrar estos archivos pero sin la terminación *-dist*, editarlos y leer los comentarios que aparecen en cada uno

de ellos. También existe un archivo adicional en el directorio *conf* llamado *mime.types*, este archivo generalmente no necesita ser editado.

Primero es necesario editar el archivo *httpd.conf*, que contiene los atributos generales acerca del servidor como es el número de puerto, con que usuario se ejecuta, etc. después se edita el archivo *srm.conf* donde se da la raíz de la estructura del árbol de los documentos, y funciones especiales como interpretación de HTML por parte del servidor o interpretación interna de mapas sensitivos, etc. y finalmente el archivo *access.conf* para ver los tipos de servicios que son permitidos y en que circunstancias.

### 5.4.3 Configuración del servidor Apache

El siguiente paso fue modificar los archivos de configuración. Los archivos de configuración siguen algunas reglas:

1. Es indistinto el uso de mayúsculas o minúsculas (excepto directorios, que en UNIX si lo es)
2. Los comentarios empiezan con #
3. Los espacios extras se ignoran
4. Cada línea debe contener a lo más una directiva

A continuación se detallan las modificaciones mínimas que deberán llevarse a cabo en los archivos de configuración para que el servidor pueda arrancar. Estas modificaciones se realizarán en los tres archivos antes mencionados:

#### Configuración del servidor

Tomar el archivo *httpd.conf-dist* y copiarlo a *httpd.conf* que será el que utilizará el servidor.

**ServerType (tipo de servidor).** Si bien el servidor puede correr a través de *inetd*, es mejor utilizar *standalone*, pues en el primero ejecutará una copia del servidor por cada requisición y en la segunda se ejecutará una sola vez y permanecerá siempre en memoria.

**Port (puerto).** El estándar es utilizar el 80. A menos que se desee que se ejecute en otro puerto se cambiará, esto no se modificó. HTTPd puede ser ejecutado por cualquier usuario sólo si se utiliza un puerto arriba del 1024.

**User (usuario).** Para maximizar la seguridad es importante tener un usuario en el servidor llamado *nobody* (común en UNIX). Se verifica que así sea en el */etc/passwd*. La línea de *nobody* se ve así (notesé el valor -1).

```
nobody:*:-1:100:nobody:/dev/null:
```

**Group (grupo).** Igual que *user*. Se modifico a:

```
group nobody
```

Es necesario asegurarse que se tiene el siguiente registro en */etc/groups* (igual, su id tiene que ser -1):

```
nobody:-1:
```

**ServerAdmin.** Deberá ser el nombre de la cuenta de correo electrónico de quien será el administrador del sitio de WWW. Muchos optan por crear un alias llamado Webmaster. Para este caso es *admonweb*.

```
ServerAdmin admonweb@ine.gob.mx
```

**ServerRoot.** Es recomendable que se utilice el directorio que HTTPd recomienda para instalar el servidor (*/usr/local/etc/httpd*).

El archivo *httpd.conf* ya modificado quedó como se muestra en el anexo 5.

## Configuración de recursos del servidor

Tomar el archivo *srm.conf-dist* y copiarlo a *srm.conf* que será el archivo de configuración de recursos. Por omisión, HTTPd busca las páginas de WWW en el directorio */usr/local/etc/httpd/htdocs/*, los iconos en */usr/local/etc/httpd/icons/*, y los programas CGI en */usr/local/etc/httpd/cgi-bin/*. Para nuestro caso sólo se modificaron las rutas de los documentos y de los CGI's quedando de la siguiente forma:

Para los documentos:

*/opt/data*

Para los CGI's:

*/opt/data/cgi-bin*

El archivo *srm.conf* ya modificado quedó como se muestra en el anexo 6.

## Configuración de accesos del servidor

Tomar el archivo *access.conf-dist* y copiarlo a *access.conf* que será utilizado para restringir el acceso a las páginas. A menos que se hayan modificado los directorios en *srm.conf*, no será necesario hacer ninguna modificación a este archivo. Por lo tanto se hacen las mismas modificaciones que en el archivo *srm.conf*. El archivo *access.conf* ya modificado quedó como se muestra en el anexo 7.

El servidor está ahora listo para correr.

#### 5.4.4 Dando de alta y de baja el servidor

Para levantar o terminar el proceso del servidor Apache, en el directorio `/usr/local/etc/httpd/bin` se encuentra un script llamado *httpd.init* el cual viene incluido con la distribución y nos ayudará de la siguiente manera:

Para levantarlo sólo basta teclear el siguiente comando desde la ruta antes mencionada:

```
# httpd.init start
```

Si todo salió bien regresará el prompt del sistema casi inmediatamente. Esto indica que el servidor está “arriba” y “corriendo”. Si ocurre algún error durante la inicialización del servidor aparecerá un mensaje de error en la pantalla. Si el servidor está corriendo sin problemas, es necesario comprobarlo conectándose por medio de un navegador y se debe ver la página principal de la documentación de Apache.

Hay que tener en cuenta que cuando el servidor está activo se creará un número de proceso “hijo” para poder manejar las peticiones. Si se levanta el servidor de Apache como el usuario *root*, el proceso “padre” seguirá corriendo como *root* mientras que el proceso “hijo” o los procesos “hijos” cambiarán al usuario configurado en el archivo *httpd.conf*. El número de proceso padre está escrito en el archivo *httpd.pid* en el directorio *logs*.

Para dar de baja el servidor sólo se teclea el siguiente comando desde la misma ruta:

```
# httpd.init stop
```

Si se desea que el servidor de Apache se ejecute o se detenga automáticamente después de inicializar o dar de baja el servidor SUN donde reside, se debe agregar una llamada al script en los directorios de inicio y terminación de sistema, que para este caso tratándose de sistema operativo UNIX Solaris 2.5, el directorio de inicialización es el siguiente:

```
/etc/rc3.d/
```

y se teclea el siguiente comando para crear una liga blanda hacia donde esta una copia del script de inicialización del servidor Apache:

```
# ln -s ../init.d/httpd.init S20httpd
```

quedando de la siguiente forma:

```
lrwxrwxrwx 1 root other 20 Jul 2 14:10 S20httpd -> ../init.d/httpd.init
```

Y en el directorio de terminación donde residen los scripts para “matar” los procesos cuando se da de baja el servidor SUN es el siguiente:

```
/etc/rc0.d/
```

y se teclea el siguiente comando para crear de igual forma una liga blanda:

```
# ln -s ../init.d/httpd.init K10httpd
```

y queda de la siguiente forma:

```
lrwxrwxrwx 1 root other 20 Jul 2 14:16 K10httpd -> ../init.d/httpd.init
```

Con esto aseguramos que cuando se inicia o finaliza el servidor SUN, el proceso del servidor Apache terminará e iniciará automáticamente.

## 5.5 Creación de formas y aplicaciones en HTML y PHP

De entre todas las aplicaciones que puede tener la Intranet, específicamente podemos mencionar las siguientes:

- **Un directorio telefónico** con los números de cada empleado dentro del Instituto, correo electrónico, puesto y adscripción.
- Circulares diarias sobre **informes** del Instituto en los medios de comunicación: desde los acontecimientos que puedan afectar a la política del INE y/o SEMARNAP o la relación de los días festivos fijados en el calendario laboral.
- **Foros de discusión.** Para debate y noticias que permiten a los empleados contar con un centro de reunión y discusión de las actividades y asuntos relacionados con el Instituto o temas relacionados al medio ambiente.
- **Salas de discusión** y debate en tiempo real (*Chat*) para que los empleados y personal de mando puedan discutir asuntos de interés común de manera interactiva y al mismo tiempo.
- Una sección de **aplicaciones** en la cual se pueden obtener los programas de mayor uso en el Instituto, un ejemplo claro son los programas antivirus.
- Páginas en la cual esté la información relacionada a la **informática** del INE como son estándares, metodología, etc., utilizadas dentro del Instituto así como manuales de procedimientos para la realización de tareas internas.
- **Capacitación.** Los departamentos tienen el continuo desafío de mantener el material y cursos de capacitación actualizados. Una sección para este fin es una forma muy conveniente de solucionar el problema, entregando a los empleados un acceso a la información actualizada de capacitación, junto con la posibilidad de incorporar los manuales, temarios y calendarios de cursos.



- **Clarín informático.** Es una publicación trimestral con artículos de interés en el mundo de la informática y las telecomunicaciones.

En todos estos casos, la Intranet es una herramienta fácil de configurar, utilizar y administrar. Es posible crear secciones para cada departamento o área funcional. Además, puesto que el servidor de Web está configurado en una red de área local (LAN), es perfectamente adecuado para la instalación de aplicaciones de mayor tamaño.

El directorio en el cual residirán los archivos, formas y aplicaciones será */opt/data/intranet* y a partir de éste se ira creando la estructura de acuerdo a las necesidades de expansión. Las extensiones de los archivos pueden ser *.html* o *.phtml* ya que la configuración y la forma en que se compiló PHP permite interpretar archivos con ambas extensiones.

Antes que nada es necesario mencionar que se crearon dos archivos de uso común a ser incluidos en todas las páginas: uno para el encabezado y el otro para el pie de página.

El archivo *cabeza.phtml* es el archivo que va a ser llamado desde todas las páginas de la Intranet para agregar información de etiquetas HTML que deben de estar en todos los archivos escritos en este lenguaje como: `<HTML>`, `<BODY>`, `<HEAD>`, `<TITLE>`, etc. A continuación se muestra el contenido de este archivo:

```
<html>
<head>
<title>
<? echo $titulo; ?>
</title>
</head>
<body background="/logos/fondo.jpg" bgcolor="#ffffff" text="#000000"
link="#0000ff" vlink="#009933" alink="#ff0000">
<p align="center"></p>

```

Listado 4. Listado del archivo *cabeza.phtml*

El archivo *pie.phtml* el cual va a ser llamado en la parte inferior de todos los archivos y va a contener: una función de PHP para presentar la última fecha de actualización, el pie de página "Sistema Nacional de Información Ambiental" y las etiquetas de terminación de un archivo en HTML, es decir, `</BODY>` y `</HTML>`. A continuación se presenta el contenido de este archivo:

```
<font size=2>&Uacute;ltime actualizaci&oacute;n: <?echo Date(" d/m/y", getLastMod());?>
</font>
<br>
<font face="Arial" color="blue" size=1>Sistema Nacional de Informaci&oacute;n Ambiental</font>
</body>
</html>
```

Listado 5. Listado del archivo *pie.phtml*

### 5.5.1 Página principal de la Intranet del INE

De acuerdo a las aplicaciones antes mencionadas y teniendo en cuenta que el Instituto Nacional de Ecología es una dependencia de gobierno se realizó el diseño de la página principal de una manera seria y utilizando sólo las imágenes y fondos necesarios para tener una presentación de calidad. Las páginas internas siguen un mismo estándar en cuanto a la presentación de las pantallas así como una uniformidad y administración centralizada en cuanto a:

- Color y/o fondo de las páginas
- Tipografía
- Partes
  - Superior
  - Inferior
- Color de las letras
- Color de las ligas
- Color de las ligas visitadas

Lo cual se logra de una manera fácil y sencilla al incluir en todos los archivos el pie y el encabezado antes mencionado. También con esto tenemos la posibilidad y dinamismo de que en el momento que se desee cambiar alguna de las características anteriores sólo es necesario modificar el archivo *cabeza.phtml* o *pie.phtml* según lo que se requiera modificar, agregar o actualizar y automáticamente estos cambios se presentarán en todas y cada una de las páginas internas. El código se muestra en el anexo 8 y la pantalla de la página principal se presenta a continuación:

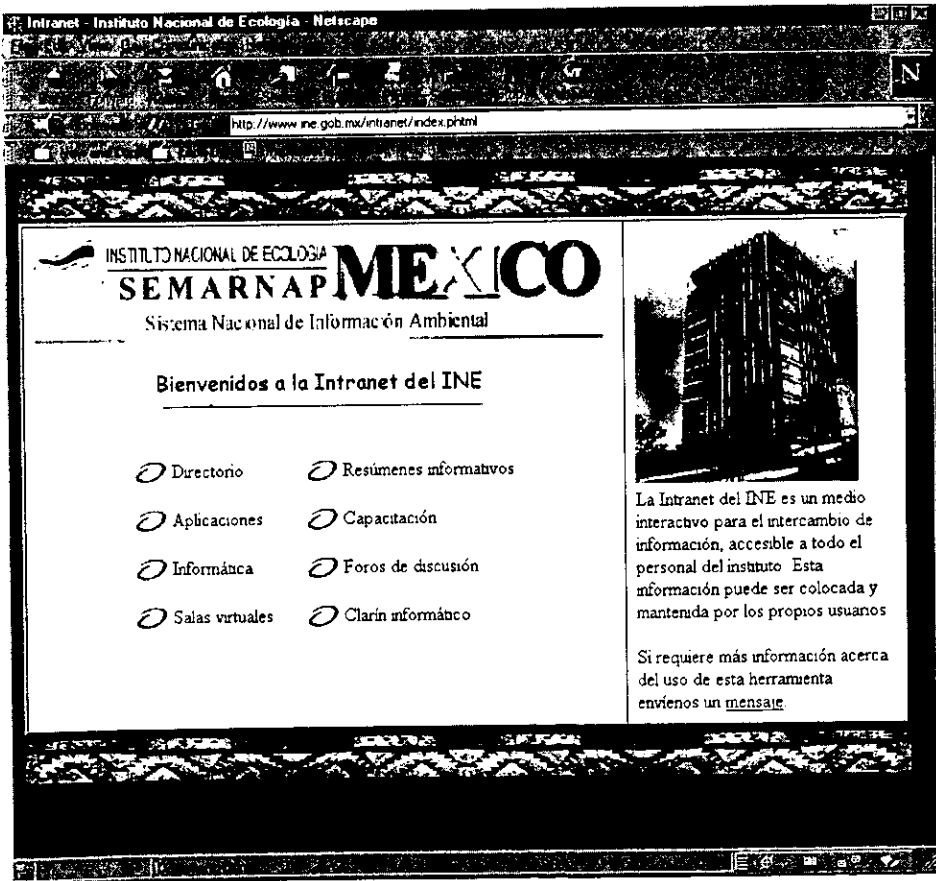


Figura 13. Pantalla principal de la Intranet del INE

### 5.5.2 Directorio telefónico

Para la realización del directorio telefónico, fue necesario el uso de una forma en HTML y PHP así como de lenguaje SQL ya que las consultas de la información se hacen directamente sobre la base de datos institucional la cual reside en el servidor de base de datos ORACLE. Cabe mencionar que dentro del mismo archivo en HTML viene el CGI en PHP, esta es una de las grandes ventajas que permite este lenguaje de programación de CGI's. La ruta del directorio es */opt/data/intranet/directorio* y el archivo se llama *directorio.phtml*.

Los datos que presentará como resultado esta forma son los siguientes:

1. **Nombre.** Es el nombre con el cual está dado de alta en recursos humanos.
2. **Correo electrónico.** Dirección electrónica en el servidor de correo institucional.
3. **Teléfono.** Número telefónico de la oficina donde labora.
4. **Puesto.** Puesto que ocupa en el instituto.
5. **Adscripción.** El nombre del área o adscripción donde labora.

El código fuente del CGI que presenta la información del personal del INE en la Intranet se lista en el anexo 9. La pantalla principal de la búsqueda en el directorio telefónico presentada al usuario es la siguiente:

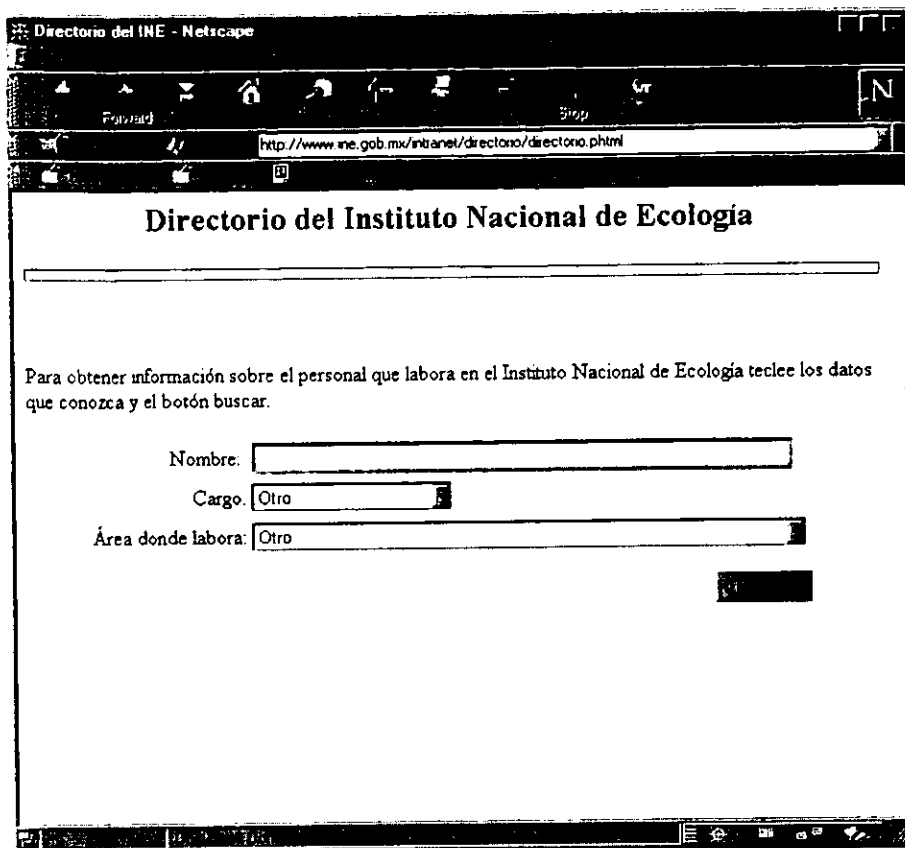


Figura 14. Pantalla principal del *directorio telefónico* del INE.

El *script* tiene la función de aceptar el nombre, el apellido o apellidos de la persona que se desea buscar así como el cargo:

- Director General
- Asesor
- Director de área
- Subdirector

- Jefe de departamento
- Otro

Y el área donde labora:

- Coordinación de Asesores
- Unidad Coordinadora de Áreas Naturales Protegidas
- Dirección General de Vida Silvestre
- Dirección General de Ordenamiento e Impacto Ambiental
- Dirección General de Regulación Ambiental
- Dirección General de Materiales, Residuos y Actividades Riesgosas
- Dirección General de Gestión e Información Ambiental
- Unidad de Asuntos Jurídicos
- Unidad de Crédito Externo
- Unidad de Administración
- Unidad de Sistemas e Informática
- Unidad de Cooperación y Convenios Internacionales
- Unidad de Sustancias Químicas y Evaluación Ambiental
- Unidad de Comunicación Social
- Unidad de Publicaciones
- Otro

Si se desconoce el cargo o el área por omisión se le puede dar "otro" y con esto hará la búsqueda por todos los cargos y en todas las áreas según sea el caso. A continuación se muestra una pantalla con los resultados obtenidos de una búsqueda con el nombre "**Rojas**":

**Directorio del Instituto Nacional de Ecología**

**Resultados de la Búsqueda**

Expresión buscada: Rojas

Nombre	Correo electrónico	Teléfono	Puesto	Adscripción
ALFONSO ROJAS	<a href="mailto:arojas@ine.gob.mx">arojas@ine.gob.mx</a>	624 35 55	Director de Area	UNIDAD COORD. DE AREAS NAT. PROTEG
CECILIA ROJAS	<a href="mailto:crojias@ine.gob.mx">crojias@ine.gob.mx</a>	624 12 90	Subdirector	DIRECCION GENERAL DE VIDA SILVESTRE
EDGARDO ROJAS	<a href="mailto:erojas@ine.gob.mx">erojas@ine.gob.mx</a>	624 15 39	Subdirector	DIR. GRAL. DE REGULACION AMBIENTAL
FRANCISCO JAVIER ROJAS	<a href="mailto:paco@ine.gob.mx">paco@ine.gob.mx</a>	624 36 31	Jefe de Departamento	UNIDAD DE SISTEMAS E INFORMATICA
GERSON ROJAS	<a href="mailto:grojas@ine.gob.mx">grojas@ine.gob.mx</a>	624 36 43	Director de Area	UNIDAD DE CREDITO EXTERNO
JESUS ROJAS	<a href="mailto:jarojas@ine.gob.mx">jarojas@ine.gob.mx</a>	624 15 20	Subdirector	UNIDAD DE ADMINISTRACION
MARIA GUADALUPE	<a href="mailto:mrojias@ine.gob.mx">mrojias@ine.gob.mx</a>	624 72 40	Jefe de Departamento	UNIDAD DE COOP. Y CONV INT

Figura 15. Resultados de la búsqueda con la expresión "Rojas".

A continuación se muestra otro ejemplo escribiendo la siguiente información en la pantalla principal de búsqueda del directorio:

**Nombre:** Francisco Javier Rojas

**Cargo:** Jefe de Departamento

**Área donde labora:** Unidad de Sistemas e Informática

Y el resultado de esta consulta se muestra a continuación:

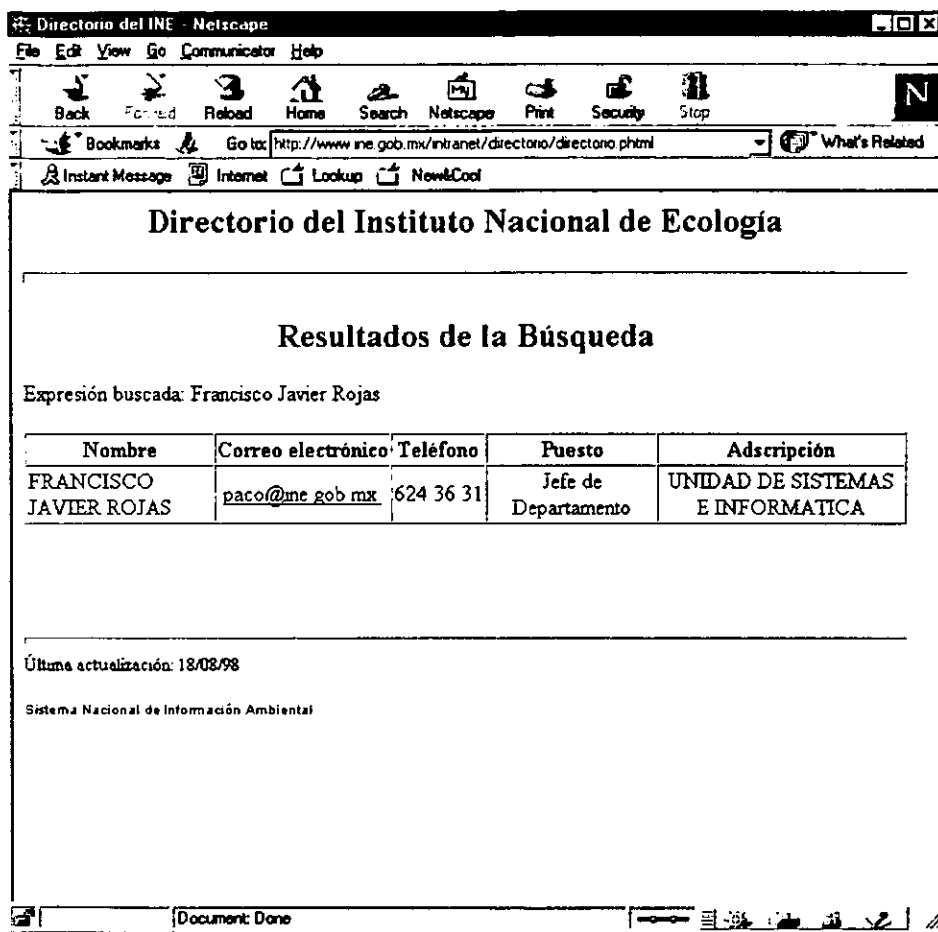


Figura 16. Resultados de una búsqueda exacta.



En caso de que la base de datos no esté disponible u ocurra algún error, automáticamente el CGI llamará a un archivo llamado *mantenimiento.phtml* el cual contiene el siguiente código:

```
<?
  Stitulo = "En mantenimiento";
  include "cabeza.phtml";
?>

<BR>
<BR>
<BR>
<center>
<h2>Mantenimiento</h2>
</center>
<BR>
<hr size=8>
<BR>

Por el momento nos encontramos d&acute;ndole mantenimiento a nuestro servidor.
Disculpe las molestias que esto le ocasiona. Esperamos tener la informaci&acute;n al aire lo m&acute;s
pronto posible.
<BR>
<BR>

<hr size=5>
<BR>
<BR>
</body>
</html>
```

Listado 6. Listado del archivo *mantenimiento.phtml*.

Y la pantalla de mantenimiento es la siguiente:

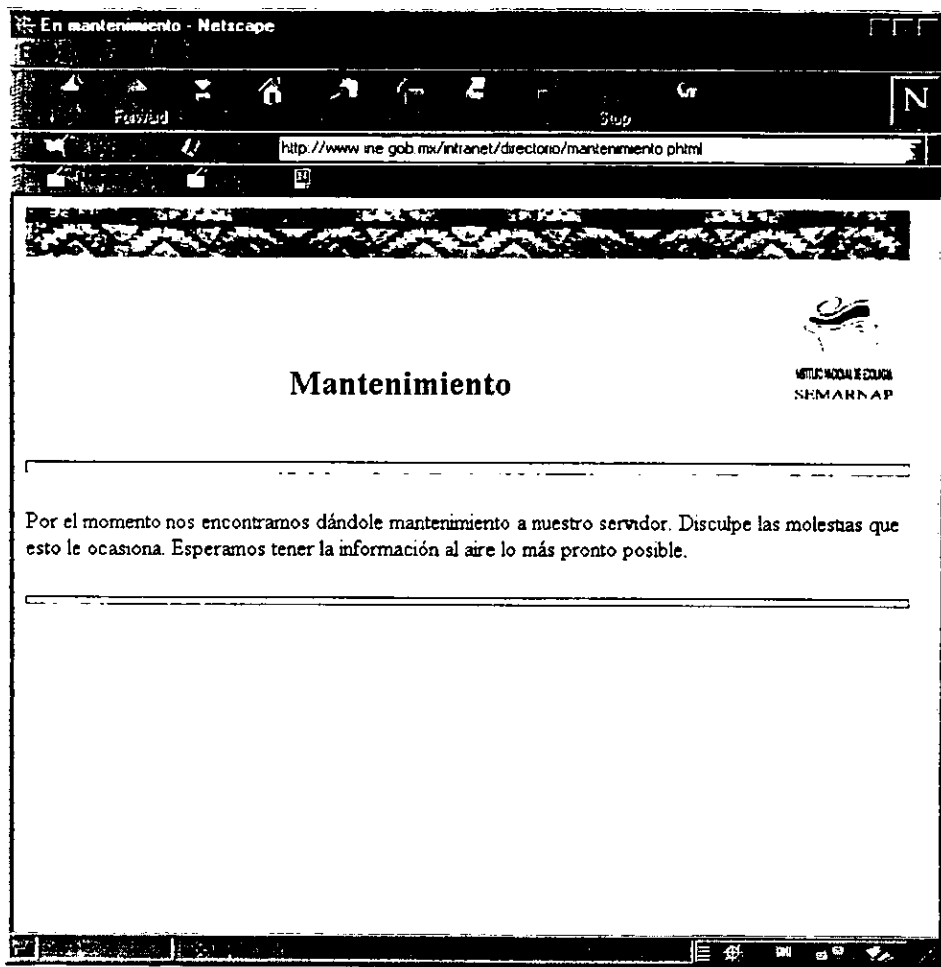


Figura 17. Pantalla de *mantenimiento*.

Es necesario aclarar que para el buen funcionamiento y despliegado de la información del directorio telefónico del INE, los datos de la base institucional residentes en ORACLE deben de estar correctos y actualizados.

### 5.5.3 Foros de discusión

El foro de discusión es un recurso que hace posible la distribución electrónica de mensajes, preguntas, artículos, etc., agrupados por temas. No requiere la presencia simultánea del emisor ni del receptor. Cualquiera que tenga acceso al foro puede participar, esto es, personas que pertenezcan al Instituto pueden ver y poner mensajes.

La variedad de los temas que se tratan en los foros son relativos y dependen de las necesidades de las áreas de INE. En esta sección sólo se hablará de cómo se instala y configura un foro y con esto el administrador de la Intranet puede crear los que le sean requeridos.

#### Características principales

Las razones por las cuales se optó por este conjunto de *scripts* para la creación de foros se listan a continuación:

- Múltiples foros pueden ser instalados
- Administración por medio de Web
- Mensajes relacionados
- Personalización de los mensajes (incluyendo botones, *frames* y formas)
- Envía un correo al administrador o a los integrantes del foro cuando este es actualizado
- Tiene una interfase con *swish* el cual es un poderoso indexador de páginas para formato HTML, y el cual ya se tiene instalado en el servidor.
- Permite a los participantes del foro modificar y/o suprimir sus mensajes por medio de un *password* personal

## Obteniendo el *software* para los foros

El *software* utilizado para la creación de los foros también es de dominio público y se llama **W-*agora***. Está completamente escrito en lenguaje de PHP por lo cual es 100% compatible con lo que se instaló anteriormente.

Para obtener información y la última versión de **W-*agora*** se consultó el siguiente URL: [http://www.araxe.fr/w-\*agora\*/](http://www.araxe.fr/w-<i>agora</i>/) que es un sitio Web en Internet donde se puede obtener la última versión, se encuentran las características, la documentación y demostraciones de esta poderosa herramienta utilizada para la creación de foros.

Los archivos fuente por lo regular se distribuyen en un formato *tar* y compactados en un archivo que por lo general se llama *w-*agora*-1\_2b2.tar.gz* y tiene un tamaño un poco mayor a los 21 Kb y descompactado se necesita un espacio aproximado de 96 Kb, el poco espacio que ocupa se debe a que se trata de archivos de texto o *scripts*. La forma de instalar los archivos es igual a la instalación de los anteriores, es decir:

1. Lo primero que se necesita es tener un directorio disponible, para este caso se utilizó el directorio */opt/data/intranet/foros/admin*

2. Descompactar el archivo de distribución con el siguiente comando:

```
gunzip w-agora-1_2b2.tar.gz
```

3. Al acabar de ejecutarse quedará un archivo llamado *w-*agora*-2b2.tar*

4. Para quitar la extensión *tar* se utiliza el siguiente comando:

```
tar xvf w-agora-2b2.tar
```

Automáticamente se creará el directorio */opt/data/intranet/foros/admin/w-*agora** como base, y la siguiente lista de archivos:

Total 97								
drwxrwxrwx	2	http	staff	512	Aug	20	16:20	.
drwxrwxrwx	3	http	staff	512	Aug	20	20:09	..
-rw-rw-rw-	1	http	staff	14345	Aug	20	16:19	admin.phtml
-rw-rw-rw-	1	http	staff	820	Aug	20	16:19	changelog
-rw-rw-rw-	1	http	staff	10216	Aug	20	16:20	doc.html
-rw-rw-rw-	1	http	staff	7780	Aug	20	16:20	editform.phtml
-rw-rw-rw-	1	http	staff	4248	Aug	20	16:20	editmsg.phtml
-rw-rw-rw-	1	http	staff	765	Aug	20	16:20	filelist
-rw-rw-rw-	1	http	staff	0	Aug	20	16:20	firsttime
-rw-rw-rw-	1	http	staff	498	Aug	20	16:20	form_en.html
-rw-rw-rw-	1	http	staff	188	Aug	20	16:20	form_en.var
-rw-rw-rw-	1	http	staff	511	Aug	20	16:20	form_fr.html
-rw-rw-rw-	1	http	staff	199	Aug	20	16:20	form_fr.var
-rw-rw-rw-	1	http	staff	6449	Aug	20	16:20	readme
-rw-rw-rw-	1	http	staff	1345	Aug	20	16:20	toolbar.phtml
-rw-rw-rw-	1	http	staff	2916	Aug	20	16:20	w-agera.msg.en
-rw-rw-rw-	1	http	staff	3147	Aug	20	16:20	w-agera.msg.fr
-rw-rw-rw-	1	http	staff	30169	Aug	20	16:20	w-agera.phtml

Listado 7. Listado de archivos del software utilizado para los foros.

Como se dijo anteriormente *W-agera* es totalmente dinámico y ajustable a las necesidades del usuario, por lo cual se modificaron las plantillas de lenguaje (*nombre\_foro.msg*), forma de petición de datos (*form.html*) y plantilla de presentación de los mensajes (*form.var*) para pedir y poner los mensajes en español. El archivo de mensajes ya modificado se muestra en el anexo 10. El archivo de la forma para ingresar los datos se lista a continuación:

```
<center><h2>
Por favor, llena esta forma para enviar tu mensaje
</center></h2>
<hr>
<b>Nombre:<b><br>
<input name=author value="<?echo $author>" size=40><p>
<b>Correo electr&oacute;nico:<b><br>
<input name=address value="<?echo $address>" size=40><p>
<b>Asunto:<b><br>
<input name=title value="<?echo $title>" size=80><p>
<b>Pon una contrase&ntilde;a:<b><br>
<input name=password type=password size=10><p>
<b>Escribe tu mensaje:<b><br>
<textarea name="text" rows=10 cols=70><?echo $text></textarea>
```

Listado 8. Forma para ingresar datos.

Y por último el archivo para la presentación de los mensajes en el momento de leerlos:

```
<?
$form_var[0]="author";
$form_lib[0]="<i>Autor</i>:<TD><b>$author</b>";
$form_var[1]="address";
$form_lib[1]="<i>Direccion de correo </i>:<TD><it><a href=mailto:$address>$address</a></it>";
?>
```

Listado 9. Archivo de presentación de mensajes.

### Instalación y configuración de un foro

El siguiente paso es abrir el archivo *w-agera.phtml?install* con un browser y seguir las instrucciones de instalación que se presentan. Para este caso el URL es el siguiente: *http://www.ine.gob.mx/intranet/foros/admin/w-agera/w-agera.phtml?install* y se presentará la siguiente pantalla de instalación:

[ List notes ] [ Post ] [ Administracion ]

## Install a new forum

Give the destination directory in both URL and "full pathname" formats

Full pathname of installation directory

URL of Directory

Give a name to this forum

choose language for messages and form file templates

Give a password for this forum

Please, reenter the password

Figura 18. Pantalla de instalación de foros.

A continuación se explican los parámetros que se le tienen que escribir para poder crear un foro:

**Directorio de instalación.** Es la ruta completa del directorio en el cual residirán los archivos utilizados por el foro.

**Nombre del foro.** Es el nombre dado al foro, el nombre del directorio donde residirá y la base de datos de los archivos relacionados.

**Lenguaje (francés o inglés).** Instalará el archivo de mensajes y la forma para poner mensajes en el lenguaje elegido. Cabe mencionar que estas plantillas (mensajes y forma) pueden ser modificadas al lenguaje español, lo cual se mostrará más adelante.

**Contraseña o password.** Será la contraseña utilizada para la administración de este foro.

Como siguiente paso se tienen que escribir los datos para lograr la creación del foro. Este mecanismo se mostrará con un ejemplo para que quede más claro. El foro de ejemplo se alimentará con los siguientes datos:

**Ruta del directorio:** /opt/data/intranet/foros/

**URL del directorio:** /intranet/foros/

**Nombre del foro:** ine

**Idioma:** inglés

Y la pantalla de instalación queda de la siguiente forma:



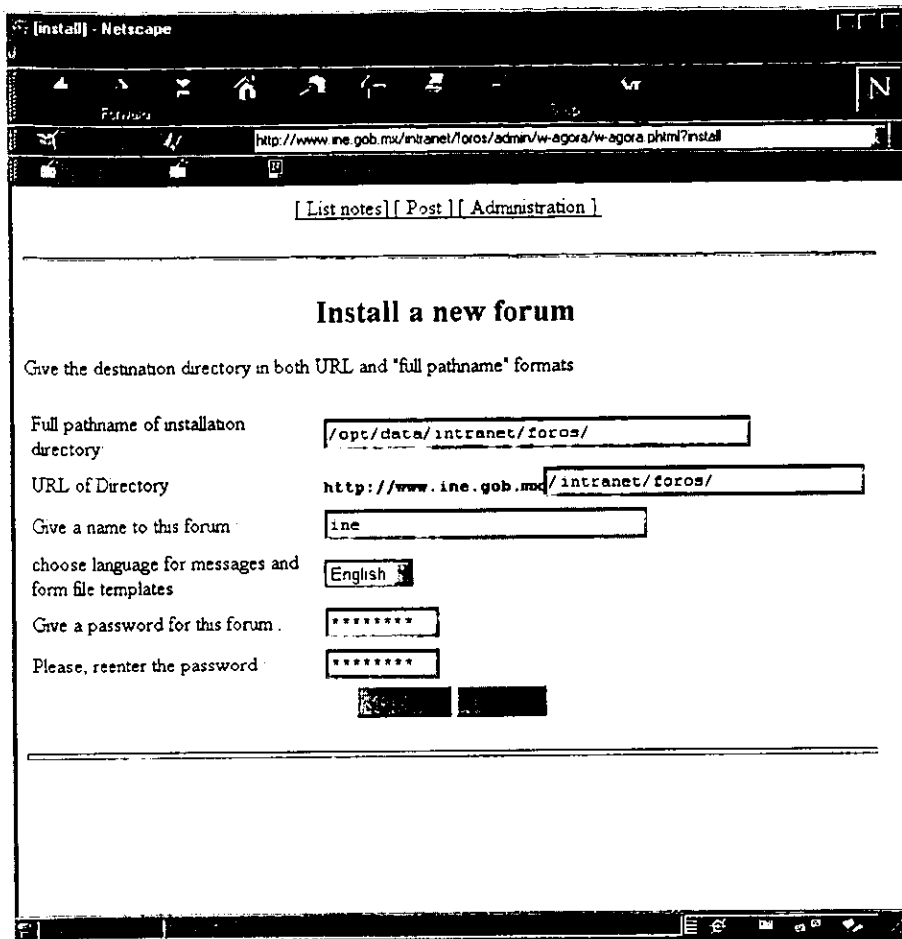


Figura 19. Pantalla de instalación con los datos para la creación de un foro llamado *ine*.

Una vez apretado el botón “*submit*” de la pantalla de instalación, el programa nos presentará una pantalla de confirmación de que el foro ha sido creado en los directorios especificados, los archivos de mensajes *ine.msg*, *form.html* y *form.var* así como la base de datos del foro.

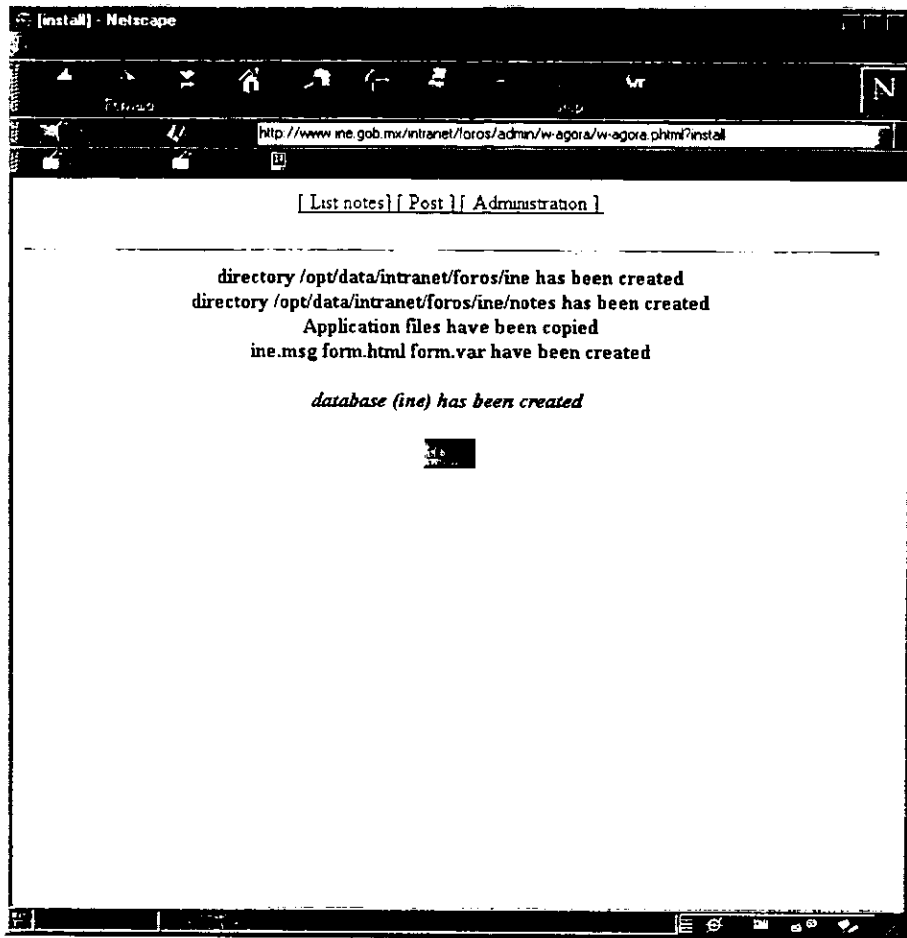


Figura 20. Pantalla de confirmación del foro creado

Al oprimir el botón de "ok" automáticamente nos presentará la pantalla de administración del foro que hemos creado. Los valores que aparecen son los que el programa da por omisión. Así que se explicarán los que fueron cambiados y por que valores. Como ya se dijo una de las características principales de *W-agera* es que es fácil de personalizar.

- Título de la página: *ine* se cambió por “*Foro del Instituto Nacional de Ecología*”
- Fondo (*background*): Se cambió el parámetro *bgcolor='#FFFFFF'* por *background='/logos/fondo.jpg'*
- Archivo de mensajes: *ine.msg* se cambió por *ine\_esp.msg*
- Archivo de la forma para ingresar datos: *form.html* por *form\_esp.html*
- Archivo de presentación de mensajes: *form.var* por *form\_esp.var*
- Dirección de correo electrónico: se escribió *admonweb@ine.gob.mx*
- Y por último se da el *password* para la administración de este foro

Esta pantalla, dado que son muchas las opciones que presenta, se muestra en las dos imágenes complementarias siguientes:

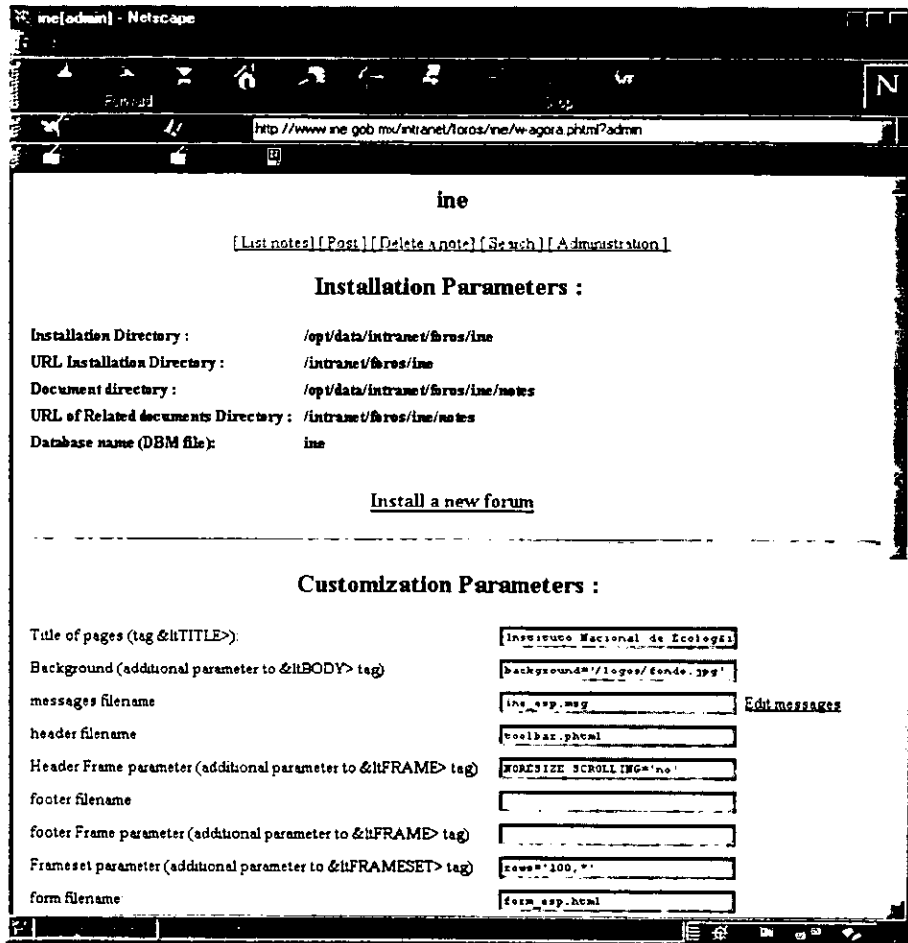


Figura 21. Página de administración(1) del foro ine con los nuevos valores asignados.

ine(admin) - Netscape

http://www.ine.gob.mx/intranet/foros/ine/w-ajora.phpml?admin

form filename:

filename containing form's variables:  [Edit variables](#)

Do you wish to use search (via swish) support?  Yes  No

Pathname:

Do you wish to use file upload support?  Yes  No

Do you wish to use server file browser feature?  Yes  No

Base directory for server file browser

Do you wish to send a mail to the administrator in case of update?  Yes  No

Administrator's mail address:

Message to send: 

Forum has been updated

Do you wish to allow replies (threaded documents)  Yes  No

Do you wish to allow users to modify (EDIT) documents  Yes  No

Do you wish to allow users to DELETE documents  Yes  No

Old Administrator password:

New Administrator password:

Figura 22. Página de administración(2) del foro *ine* con los nuevos valores asignados.

Una vez terminado de llenar los datos y oprimir el botón *submit*, nos presentará la pantalla de confirmación de que el nuevo foro ha sido modificado con los valores personalizados. Y por último la pantalla principal del foro en el cual ya se ven los cambios especificados.

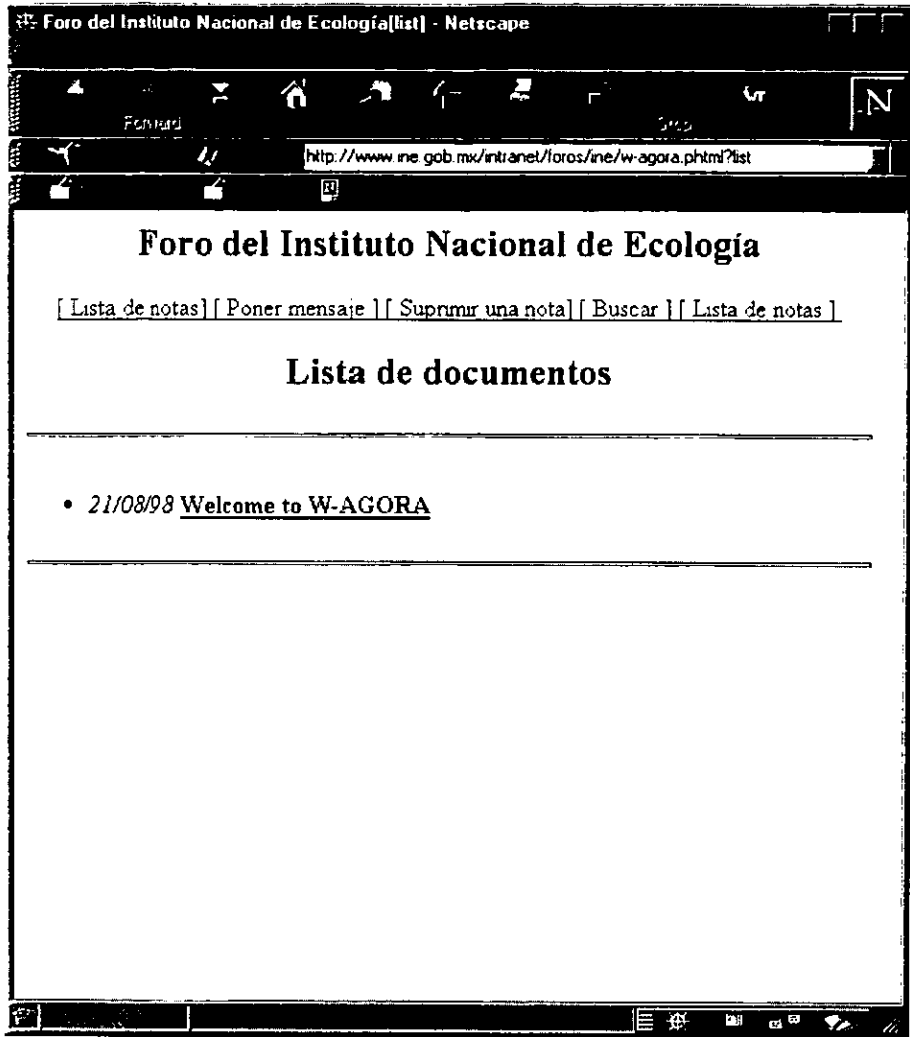


Figura 23. Pantalla principal del foro *ine*.

En lo que a administración se refiere este es el último paso, de aquí en adelante sólo depende de que los usuarios empiecen a introducir e intercambiar información. El funcionamiento es muy

sencillo y no requiere de mayor explicación debido a que la herramienta es muy amigable para leer y escribir mensajes.

#### 5.5.4 Salas de discusión en tiempo real (*Chat*)

Un *Chat* es ante todo un medio de comunicación en tiempo real. Esto es, un lugar donde la gente puede comunicarse, justo en el momento en el que escribe, con otros. La persona tiene la ventaja de saber que al otro lado de la comunicación está el receptor, y es tan sencillo como escribir con el teclado para comunicarse. Algunas definiciones que son necesarias conocer para el uso del *chat* son las siguientes:

- **Canales:** También habitaciones (*rooms*). Son los lugares donde se reúne la gente a comentar. Pueden tener una temática específica. Habitualmente un chat suele tener uno o más canales temáticos.
- **Nick:** También alias, *nickname*, *chatname*, etc. es el nombre con el que el usuario se identificará en el chat. Habitualmente son nombres diferentes del nombre propio (seudónimos), y generalmente se utiliza siempre el mismo.

Bien es sabido que la buena comunicación depende de saber comunicarse; no sólo cómo comunicarse, sino de que forma hacerlo. Así pues siempre es importante tener una actitud de respeto y seguir unas normas que, aunque no sean impuestas generalmente son preferibles por el bien de la comunicación entre todos.

PHP *Chat* es una sala interactiva de plática en tiempo real. PHP *Chat* tiene las siguientes características:

- Se pueden definir canales por medio de un archivo de texto
- Tiene una bitácora de usuarios
- A las oraciones que empiezan con *http*, *ftp* y *mailto* automáticamente les agrega la liga correspondiente

- No permite el ingreso de *tags*, estos son filtrados
- Presenta dos versiones, una con *frames* y otra sin *frames*
- Soporta tipos de letras como negritas, itálicas y fuentes más grandes

### Obteniendo PHP Chat

Para obtener información y la última versión de **PHP Chat** se consultó el siguiente URL: <http://www.cei.net/~jackson/> que es uno de los tantos sitios Web en Internet donde podemos encontrar los scripts y la documentación. El archivo fuente es llamado comunmente **phpchat.tar** o **phptar.zip** y tiene un tamaño aproximado de 41 KB y descompactado no cambia en gran medida el tamaño. Los pasos que se siguieron para obtener los archivos fueron los mismos utilizados en las otras aplicaciones:

- Lo primero que se necesitó fue tener un directorio disponible, para este caso se utilizó el directorio */opt/data/intranet/salas*
- Quitar la extensión *tar* utilizando el siguiente comando:

```
tar xvf phpcat.tar
```

A diferencia de las otras aplicaciones **PHP Chat** no creará ningún directorio, sólo se crearan los archivos automáticamente en el directorio donde se ejecutó el comando:

Total	40						
drwxr-xr-x	2	http	staff	512	Aug 26	13:13	.
drwxrwxr-x	8	http	staff	512	Aug 26	13:07	..
-rw-r--r--	1	http	staff	25	Aug 21	11:59	channels.txt
-rw-r--r--	1	http	staff	7297	Aug 21	11:59	chat.phtml
-rw-r--r--	1	http	staff	167	Aug 21	12:07	chat1.log
-rw-r--r--	1	http	staff	690	Aug 21	11:59	exitchat.gif
-rw-r--r--	1	http	staff	3490	Aug 21	11:59	index.html
-rw-r--r--	1	http	staff	6449	Aug 21	11:59	inout.phtml
-rw-r--r--	1	http	staff	7173	Aug 21	11:59	noframe.phtml
-rw-r--r--	1	http	staff	3506	Aug 21	11:59	pchat1.gif
-rw-r--r--	1	http	staff	560	Aug 21	11:59	pchat2.gif
-rw-r--r--	1	http	staff	2089	Aug 21	11:59	refresh.phtml
-rw-r--r--	1	http	staff	0	Aug 21	12:06	users.log

Listado 10. Archivos de PHP Chat.



PHP *Chat* trabaja leyendo y escribiendo en un archivo especificado, y utiliza un archivo por canal. Los canales se pueden definir en el archivo *channels.txt* asignando un nombre de canal y el nombre del archivo para almacenar la información de ese canal. La sintaxis es la siguiente:

*Nombre canal=nombre\_archivo*

Y se tienen que seguir algunas reglas muy sencillas para evitar problemas al momento que sea interpretado por los scripts de PHP *Chat*.

- Es importante que no se dejen líneas en blanco en este archivo ya que podría causar algún error
- El archivo sólo debe contener las líneas de los canales que se tengan
- No se debe incluir el signo de igual en el nombre del canal
- Y no incluir espacios entre el signo de igual y el nombre del archivo

El script del *chat* lee y agrega la información al nombre del archivo dado en *channels.txt* que está actualmente seleccionado. Si el archivo llega a crecer más de 20 KB, es cortado y reemplazado con las últimas 15 líneas que tenía. También hay otro archivo que se utiliza, *users.log*, que sirve para llevar una bitácora de los usuarios que han entrado, su dirección de correo electrónico y la dirección IP de la máquina desde donde se conectaron.

### **Instalación de las salas de discusión interactivas**

Como ya se dijo anteriormente una de las ventajas que presenta escribir y/o tener los códigos en lenguaje PHP es poder modificarlos de acuerdo a las necesidades del usuario sin tener que compilarlos para crear un archivo ejecutable. PHP *Chat* como ya vimos presenta dos tipos de versiones: *frames* y *no frames*. La que se optó por utilizar en el INE, fue la versión de *frames*; ya que presenta más opciones de uso que más adelante se comentarán con las pantallas de presentación. Los archivos de uso para la versión con *frames* son:

- *chat.phtml*
- *input.phtml*
- *refresh.phtml*

Estos tres archivos fueron modificados en su contenido para que se estandarizaran en los fondos y tipos de letras, así como también en la presentación de los comentarios para que aparezcan en idioma español. El listado del archivo **chat.phtml** después de ser modificado se muestra en el anexo 11.

El archivo **chat.phtml** es el archivo principal y es el encargado de pedir los datos más importantes al usuario así como llamar a los archivos **input.phtml** y **refresh.phtml**.

El listado del archivo **input.phtml** después de ser modificado se muestra en el anexo 12. Este archivo es el encargado de interactuar con el usuario ya que es la forma donde se escribirán los mensajes, el tiempo para refrescar la pantalla, el tipo de letra que se desea desplegar y algunas normas de comunicación en Internet llamados “*emoticones*” las cuales no son más que formas expresivas de una manera escrita.

El listado del archivo **refresh.phtml** después de ser modificado se muestra en el anexo 13. El archivo **refresh.phtml** es el encargado de presentar todos y cada uno de los mensajes escritos en la sala, presentar la información de quien entra y quien sale de la sala así como de “refrescar” o actualizar la pantalla de presentación de mensajes para poder leer los que se van escribiendo.

Con esto se da por terminado la instalación de **PHP Chat**, ahora el siguiente paso es crear las salas de discusión. Estas salas son relativas y se crearan de acuerdo a los requerimientos de las instancias del INE, pero a manera de ejemplo se presentará el archivo **channels.txt**, que como ya se dijo es donde se crean las salas, con algunas salas para poder ilustrar la facilidad de creación y la forma en que aparecerán en pantalla. A continuación se muestra el contenido del archivo **channels.txt**:

```
Canal por omision=chat1.log  
Canal de la Unidad de Sistemas=chat2.log  
Canal de la Unidad de Comunicacion Social=chat3.log  
Canal de Vida Silvestre=chat4.log
```

Listado 11. Listado del archivo *channels.txt*.

En el listado anterior se muestran 4 canales, el de omisión que es el que **PHP Chat** crea, y otros 3 que fueron creados editando el archivo y agregando las líneas correspondientes: Unidad de Sistemas, Unidad de Comunicación Social y Vida Silvestre. Como se puede observar cada nombre de sala tiene asignado un nombre de archivo diferente para llevar una bitácora para cada canal.

La pantalla principal es en la que debemos escribir el seudónimo y nuestra dirección de correo electrónico para poder ingresar a alguna de las salas de interacción. Cabe mencionar que si alguno de estos datos falta no nos permitirá el acceso a la siguiente pantalla.

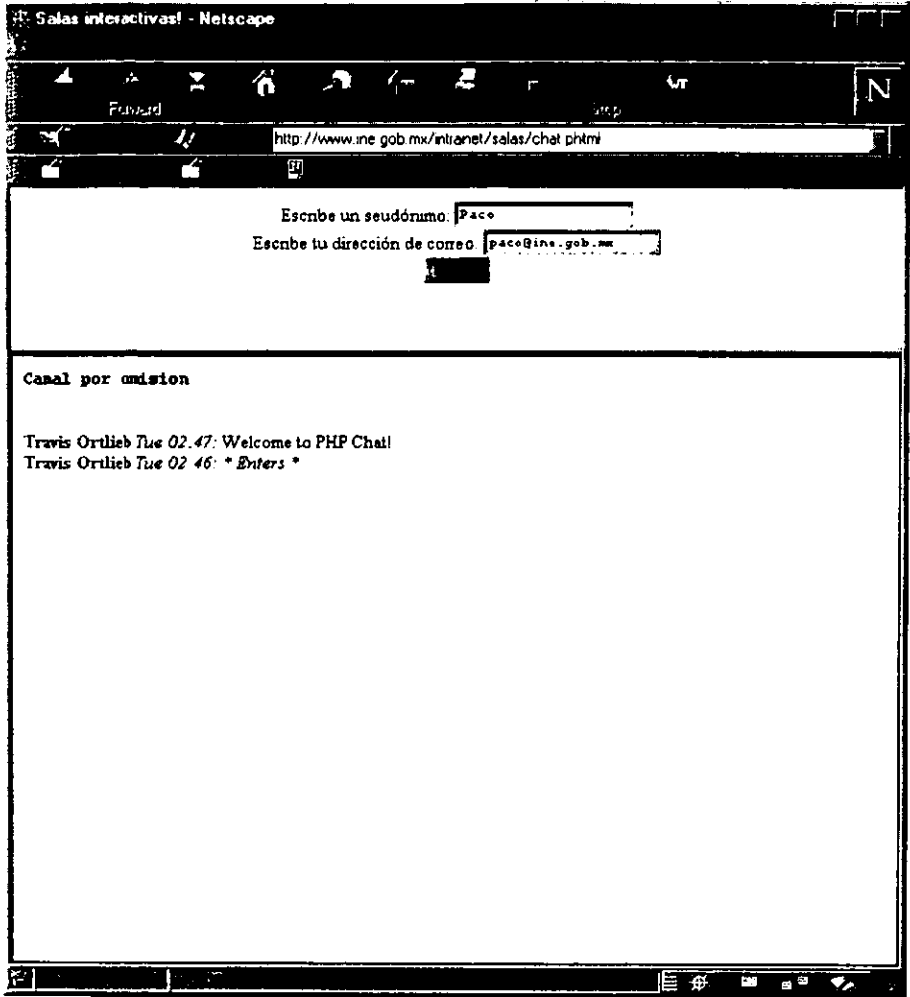


Figura 24. Pantalla principal de las salas de discusión en tiempo real (Chat).

La siguiente pantalla que se presenta es en la que debemos escoger la sala a la cual deseamos entrar, una vez seleccionada apretar el botón “Cambiar” para poder entrar a la sala seleccionada.

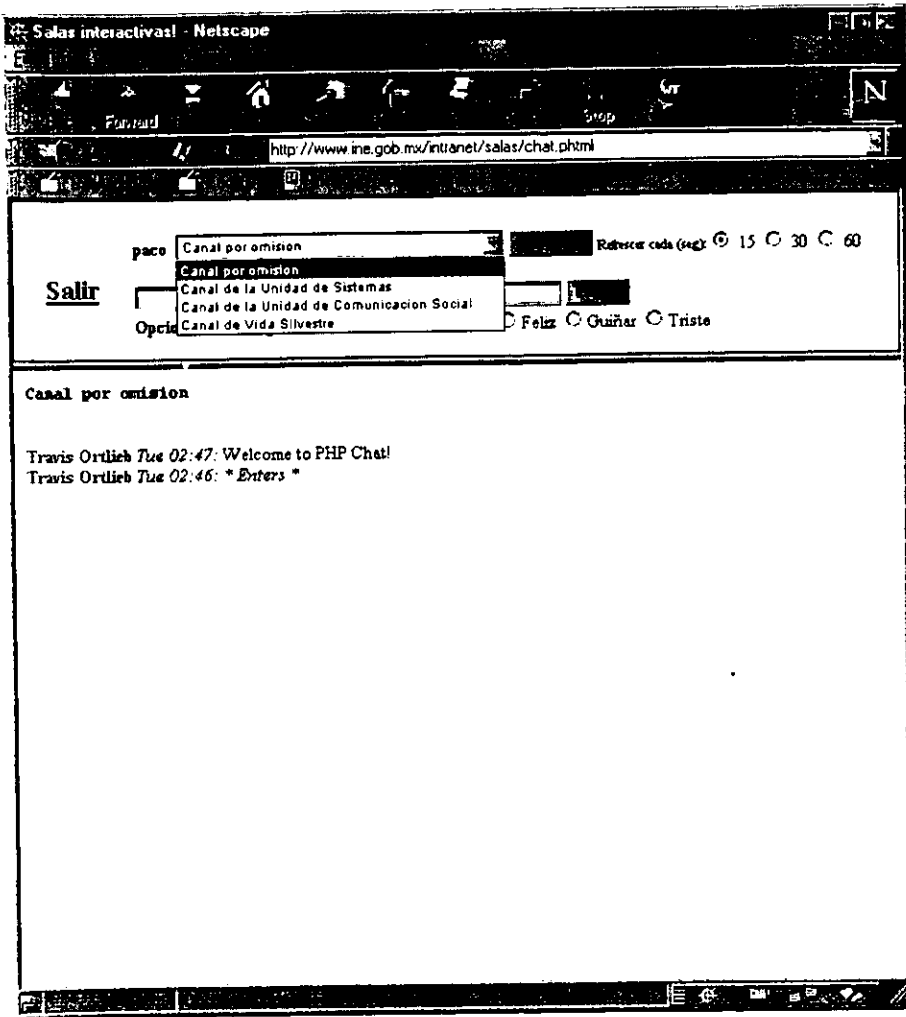


Figura 25. Pantalla de selección de canal.

De aquí en adelante sólo es necesario escribir los mensajes y automáticamente aparecerán en la parte baja de la pantalla. La pantalla se actualizará dependiendo del tiempo que hayamos elegido anteriormente para actualizar. Y para salir de la sala sólo es necesario dar *click* en la liga que dice "Salir".

### 5.5.5 Resúmenes informativos

Para la realización de la sección donde se presentan los resúmenes informativos, fue necesaria la creación de una forma en HTML y PHP. La ruta de la sección es */opt/data/intranet/informativos* y el archivo se llama *index.phtml*. Las consultas de los resúmenes las hace directamente sobre archivos localizados en el directorio en el cual reside. Cabe mencionar que para el funcionamiento de esta sección es necesario que el encargado de proporcionar la información tenga acceso al directorio antes mencionado para poder transferir los archivos vía FTP (*File Transfer Protocol*) y con un nombre de archivo basándose en el siguiente formato:

#### *añomesdia.html*

- *año*: los últimos dos dígitos del año
- *mes*: escribir el número de mes, si es de un dígito anteponer un cero
- *dia*: escribir el número del día, si es de un sólo dígito anteponer un cero

Con esto se automatiza la consulta de los archivos y se elimina la tarea de estar modificando las ligas en el archivo HTML para la llamada, ya que estos se actualizan diariamente y son demasiados.

En el anexo 14 se presenta el código fuente del archivo *index.phtml* que es el que presenta el contenido de cada uno de los resúmenes informativos en la Intranet.

Y la pantalla principal de la sección de resúmenes informativos que se presenta al usuario es la siguiente:

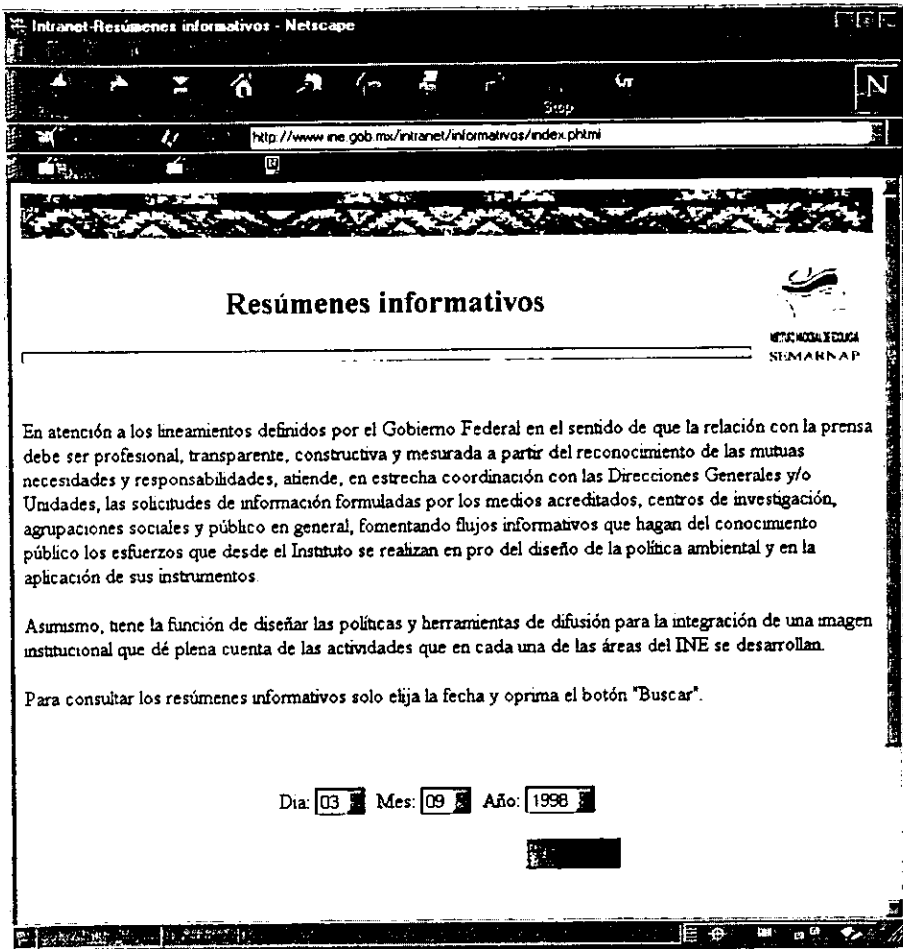


Figura 26. Pantalla principal de la sección de *Resúmenes informativos*.

Para las secciones restantes requeridas por el INE: Aplicaciones, Informática, Capacitación, y Clarín informático, la información que será puesta sólo tiene como objetivo principal la de informar a los usuarios del INE aspectos específicos, es decir, en estas secciones no hay interacción con el usuario, son sólo de consulta; por lo tanto únicamente se hizo uso del código PHP mencionado con anterioridad, para los encabezados (**cabeza.phtml**) y pies (**pie.phtml**) de página, la parte intermedia o sea el contenido de la página es información de tipo texto, con carácter meramente informativo y es relativa debido a que depende de los responsables para darles mantenimiento y tenerlas actualizadas.

Así que sólo se presentan a manera de ejemplo los códigos PHP y las pantallas de las páginas principales de cada sección.

#### 5.5.6 Aplicaciones

Para la realización de la sección de aplicaciones, se creó el directorio **/opt/data/intranet/aplicaciones** y el archivo principal se llama **index.phtml**. Cabe mencionar que en esta sección se encuentran los programas de mayor uso dentro de INE y los cuales presentan la opción de ser copiados a la máquina desde la cual se consulta. El listado se muestra en el anexo 15 y a continuación se presenta la pantalla:



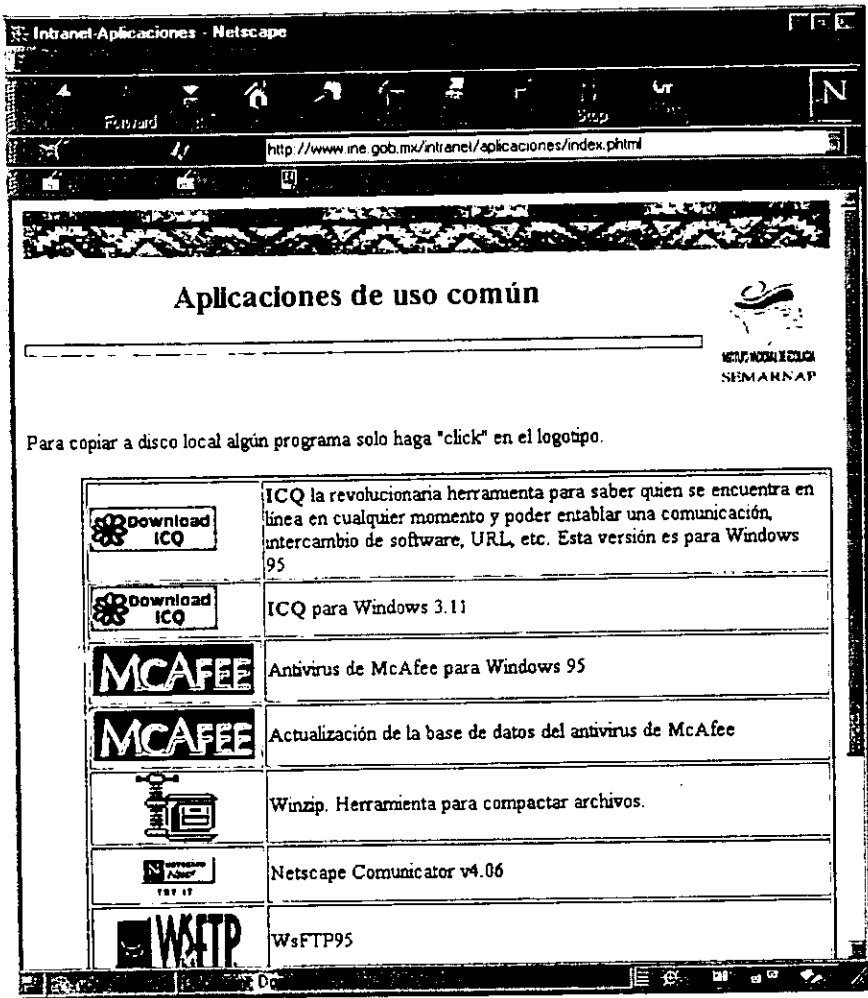


Figura 27. Pantalla principal de la sección de *Aplicaciones*.

### 5.5.7 Informática

La sección de informática está compuesta por la información referente a sistemas, manuales de procedimientos y metodologías utilizadas por la Unidad de Sistemas e Informática que deben de ser de conocimiento para todas las áreas. El directorio en el cual reside esta información es `/opt/data/intranet/metodologia` y el archivo principal es llamado también `index.phtml`. El listado del archivo principal de esta sección se presenta en el anexo 16 y la pantalla es la siguiente:

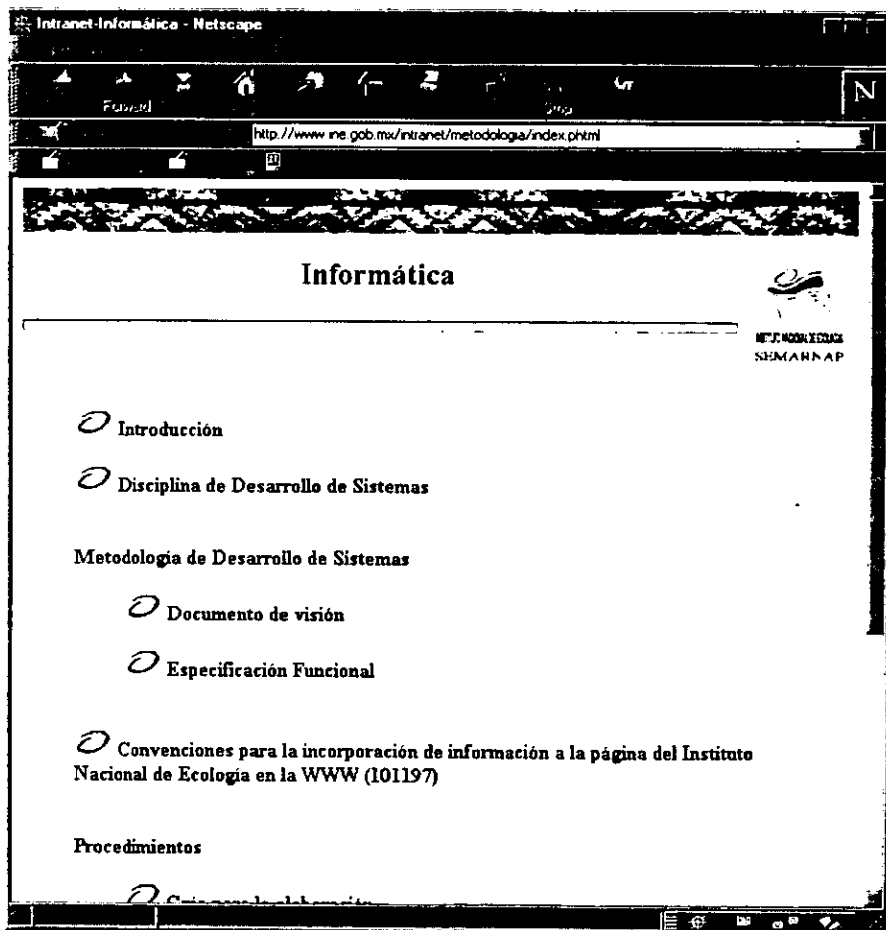


Figura 28. Pantalla principal de la sección de *Informática*.

### 5.5.8 Capacitación

Esta sección cuenta con toda la información relacionada a la capacitación que se imparte al personal del Instituto, en ella podemos encontrar programas, manuales, cursos, calendarios y fichas de inscripción. Esta sección se creó en el directorio `/opt/data/intranet/capacitacion` y el archivo principal se llama `index.phtml`. El listado del archivo principal de esta sección se presenta en el anexo 17 y la pantalla es la siguiente:

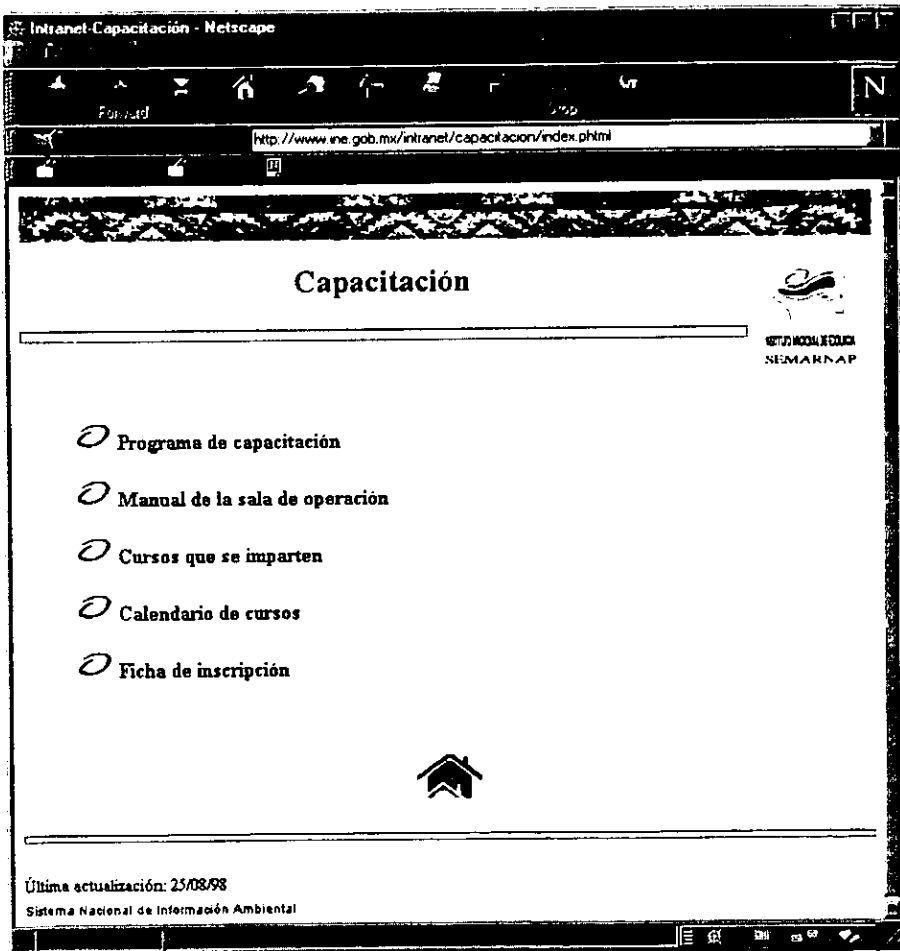


Figura 29. Pantalla principal de la sección de *Capacitación*.

### 5.5.9 Clarín informático

La sección del Clarín informático es un boletín de difusión de información referente a la informática dentro del INE y es una publicación trimestral, con esto se desea promover la cultura informática y difundir el conocimiento de tecnologías de vanguardia para todas las áreas. El directorio en el cual reside esta información es `/opt/data/intranet/clarin` y el archivo principal es llamado `index.phtml`. El listado de esta sección se muestra en el anexo 18 y la pantalla es la siguiente:

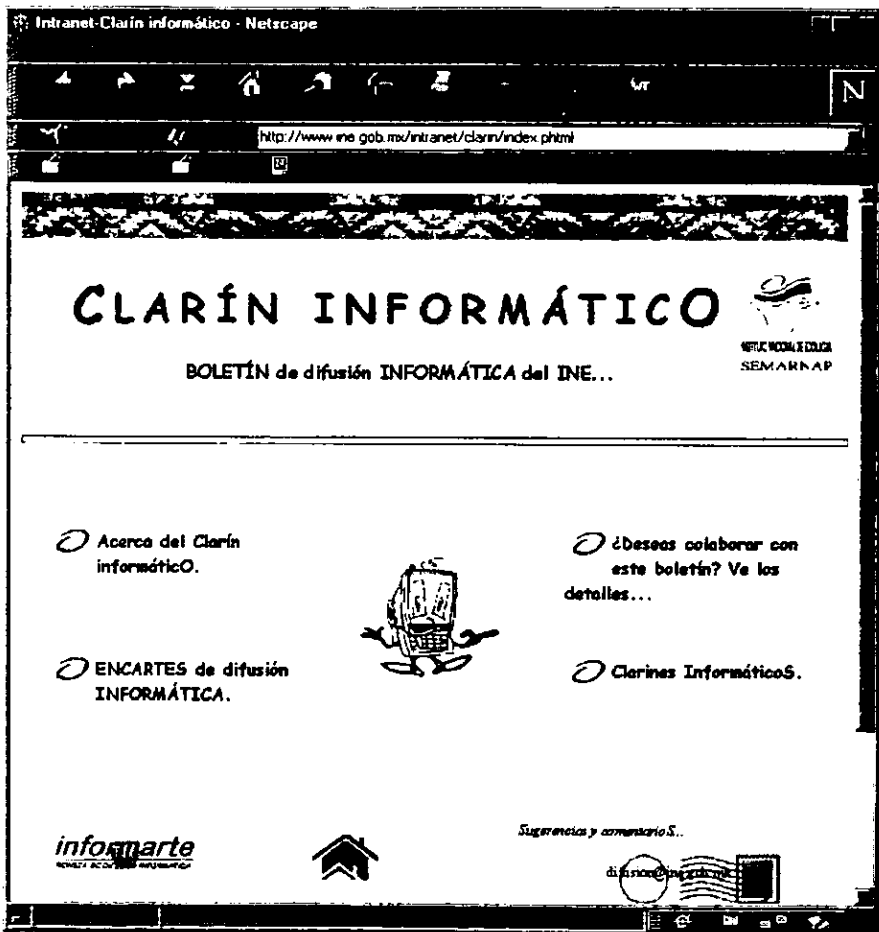


Figura 30. Pantalla principal de la sección del *Clarín informático*.

## 5.6 Seguridad y mantenimiento

### 5.6.1 Seguridad

El tema de la seguridad en WWW sería suficiente para escribir varios capítulos, pues son muchas las formas en las que podríamos comprometer la seguridad de una computadora al instalar un servidor de Web. Al instalar un servidor estamos creando una ventana a través de la cual la gente puede ver hacia el interior. Existen tres tipos de usuarios: los que visitan tranquilamente el servidor, los que intentan ver más de lo permitido, y los que intentan entrar al sistema sin autorización.

Como se explicó en el Capítulo I, la gran mayoría de los documentos en WWW están escritos en HTML (*Hyper Text Markup Language*), y son enviados a través de la red utilizando un protocolo conocido como HTTP (*Hyper Text Transfer Protocol*). Por lo tanto, todo servidor de Web debe estar corriendo algún tipo de servidor de HTTP. Es en las distintas implementaciones de estos servidores donde puede haber errores y problemas que permitan acceso no autorizado a los recursos.

Al instalar un servidor de Web se toman los siguientes riesgos:

- Que usuarios no autorizados tengan acceso a documentos confidenciales. El principal objetivo de un servidor es tener información disponible a quien desee leerla o a un público selecto, si el acceso es restringido. En el primer caso, podemos, por accidente, permitir acceso a archivos confidenciales; este problema se resuelve fácilmente, teniendo cuidado en que lo que sea disponible es exactamente lo que deseamos sea disponible; de la misma forma es conveniente tener políticas corporativas de que tipo de información puede ser puesta en un servidor de Web. Si deseamos que un documento sea leído sólo por algunas personas, el mecanismo de acceso puede ser violado por un intruso, robando el password, por ejemplo y con ello lograr acceso legítimo a páginas que deberían ser confidenciales. Actualmente este tema es de mucho interés a usuarios y se están desarrollando múltiples mecanismos para evitar este tipo de problemas.

- Que la información accesible a través del servidor permita a un extraño invadir el sistema. Supongamos que el archivo */etc/passwd* (donde UNIX guarda toda la información referente a los usuarios de un sistema y sus contraseñas encriptadas) es accesible a través de Web; esto permitiría a quien así lo deseara, y tuviera los recursos y conocimientos computacionales adecuados, intentar adivinar el de algún usuario. Con eso ganaría acceso al sistema. Por eso es importante escudar toda la información crítica del sistema y no permitirle a nadie el acceso a ella.
- Que si información confidencial es compartida entre el usuario y el servidor, ésta sea interceptada por terceros que la pueden mal utilizar. A menos que se tengan los cuidados necesarios, la información que comparten el servidor y el navegador no está cifrada y, por lo tanto, es legible por cualquiera que la intercepte. Es por eso que se recomienda no enviar números de tarjetas de crédito a través de Web (o de cualquier otro sistema en Internet).
- Que el servidor permita a un extraño ejecutar comandos que pongan en riesgo el sistema. Esto puede deberse a errores en el código del servidor o debido a errores de los programas CGI que el servidor ejecuta. Un programa CGI que acepta parámetros de un usuario es un riesgo potencial para el servidor que lo ejecuta. El problema se debe a que un programa CGI mal escrito o con errores puede permitir ejecutar otros programas, en ocasiones sin desearlo. Dicho programa puede permitir el acceso no autorizado o dañar el sistema (borrar archivos, por ejemplo). Se debe tener mucho cuidado al escribir programas CGI y verificar que todos aquellos que deseemos instalar en nuestro servidor sean seguros.

Debido a que WWW es un mercado de gran potencial dentro de Internet, hay mucho interés en desarrollar un medio ambiente en que se puedan realizar transacciones seguras, permitiendo que la información de un servidor sea accesible sólo por los que deben y que la información intercambiada entre el usuario y el servidor se mantenga confidencial. Al parecer la respuesta se encuentra en criptografía de llave pública, que permite cifrar (encriptar) información y además autenticar a usuarios, es decir, verificar que un usuario es quien asegura ser.

Las principales medidas que se deben tomar para incrementar la seguridad de un servidor de WWW se pueden resumir en las siguientes:

- Evitar usar un servidor con problemas conocidos. Muchos de los servidores de HTTP ampliamente utilizados han contenido errores que dan lugar a problemas de seguridad. En la gran mayoría de los casos, estos errores han sido corregidos en cuanto fueron detectados, y las correcciones incorporadas en versiones posteriores del servidor. Sin embargo, muchos sitios siguen utilizando versiones viejas, con el consiguiente riesgo de explotación de los huecos de seguridad.

Como dato histórico podemos mencionar que uno de los huecos más notorios ha sido el que fue encontrado en el servidor `httpd` de NCSA, versión 1.3, en febrero de 1995, y que permitía ejecutar comandos arbitrarios<sup>29</sup>. Este problema fue inmediatamente reportado y corregido. En términos generales, es recomendable utilizar siempre la última versión del servidor de HTTP que se haya elegido.

- Poner bien los permisos en los archivos del servidor. Tanto los archivos del servidor (el ejecutable, archivos de configuración, etc.), como los documentos almacenados en él deben tener permisos de acceso cuidadosamente establecidos, para impedir que alguna persona no autorizada, ya sea accidental o intencionalmente, haga modificaciones que pudieran causar problemas. En términos generales:

- ⇒ Crear un usuario y/o grupo dedicado al servidor de HTTP (posiblemente llamado `www`, `http` o algo semejante). Para este caso la cuenta es llamada `http`.
- ⇒ El programa del servidor de HTTP debe ser ejecutable solamente por el usuario autorizado (que puede ser `root` en caso de que sea necesario, por ejemplo, para poder otorgar el servicio en el puerto 80, que es el estándar, pero que es un puerto privilegiado).
- ⇒ Los archivos de configuración del servidor deben ser modificables solamente por el usuario autorizado.

---

<sup>29</sup> Revista Soluciones Avanzadas. Mayo 1996. "Seguridad en los servicios de Internet"

- ⇒ Los documentos deben ser modificables solamente por el usuario y/o grupo autorizado.
- ⇒ Revisar que no sea posible utilizar ligas simbólicas para dar acceso a archivos que están fuera del árbol de documentos del servidor.
- Reducir al mínimo las cuentas de usuarios existentes en la máquina que funciona como servidor.
- De ser posible, ejecutar el servidor en un ambiente *chroot* (con acceso solamente a una parte del sistema de archivos), para que sólo tenga acceso a los archivos estrictamente indispensables.
- Monitorear periódicamente las bitácoras del servidor para detectar comportamientos extraños.
- Desactivar la ejecución de programas CGI (*Common Gateway Interface*) salvo en casos necesarios y cuidadosamente controlados.
- Revisar cuidadosamente los programas CGI que se utilicen (para implementar formas de HTML, por ejemplo), para que no se ejecuten con ninguna clase de privilegios, no modifiquen nada en el servidor y no utilicen ninguna clase de información proporcionada por el usuario sin antes revisarla muy detalladamente.

También es importante tener cuidado del lado del cliente. Si el visualizador utilizado está configurado para ejecutar "ciegamente" cualquier documento de tipo aplicación que se encuentre, es posible ejecutar comandos arbitrarios en la máquina en la que se está ejecutando dicho visualizador.

### 5.6.2 Instalación de la autenticación de usuarios

Hay dos formas de restringir el acceso a los documentos: por restricción de la dirección IP donde está funcionando el navegador o por autenticación de usuario pidiendo cuenta y contraseña. El



primero puede ser usado, por ejemplo, para restringir documentos dentro de una compañía. Sin embargo si la gente que tendrá acceso a los documentos está ampliamente dispersada, o si el administrador de Web necesita habilitar el control sobre fundamentos individuales, es decir, a cierto grupo de personas, es posible que requiera una cuenta y una contraseña antes de permitir el acceso a un documento. Esto es llamado autenticación de usuario.

La instalación de la autenticación de usuarios se basa principalmente en dos pasos: primero, crear un archivo que contenga las cuentas y las contraseñas; y segundo, configurar el servidor para decirle que documentos estarán protegidos y que usuarios tendrán acceso (después de dar una contraseña válida) a ellos.

### **Crear la base de datos de usuarios**

Una lista de usuarios y contraseñas deberá ser creada en un archivo. Por razones de seguridad, este archivo no deberá estar bajo el directorio raíz de los documentos. La creación de este archivo se hizo sobre el directorio `/usr/local/etc/httpd` y se llama **usuarios**.

El archivo consistirá de un registro con cuenta y contraseña para cada uno de los usuarios que se creen. El formato es muy similar al archivo de contraseñas de UNIX, con la cuenta y la contraseña separadas por dos puntos. Sin embargo manualmente no se pueden dar estos datos porque las contraseñas están almacenadas en un formato encriptado para proporcionar mayor grado de seguridad. El programa usado para crear el archivo de usuarios, agregar y/o modificarlos es **htpasswd**.

**htpasswd** es un programa escrito en lenguaje C y está incluido en el directorio *src/support* de la estructura de directorios de Apache. El programa generalmente no se encuentra compilado, así que éste debe ser el primer paso. Para compilar este programa es necesario situarse en el directorio antes mencionado y ejecutar el siguiente comando:

***make htpasswd***

Después de la compilación, quedará un archivo ejecutable llamado *htpasswd*, que se puede dejar en el directorio actual o ponerlo en el directorio donde se acostumbra poner los binarios del

sistema. En este caso el archivo ejecutable se copió al directorio */usr/local/bin*. A continuación se muestra la compilación de *htpasswd*:

```
#make htpasswd
gcc -c -I./os/unix -I./include -DSOLARIS2=251 `./apaci` htpasswd.c
gcc -DSOLARIS2=251 `./apaci` htpasswd.o -o htpasswd -L../ap -lap -lsocket -lnsl
```

Listado 12. Compilación del programa *htpasswd*.

### Como usar *htpasswd*

Realmente el uso del programa *htpasswd* es muy sencillo. La sintaxis para usar este programa es la siguiente:

*Usar: htpasswd [-c] archivo\_de\_contraseñas cuenta\_de\_usuario*

*La opción -c es para crear un nuevo archivo de usuarios.*

Como se puede ver en la sintaxis del uso del programa la opción *-c* crea un nuevo archivo de usuarios. Por lo tanto esta opción sólo es utilizada al crear el primer usuario, para los siguientes no es necesario. A continuación se muestra la forma en la que se creó el usuario general que tendrá acceso a la Intranet. Cabe aclarar que los usuarios se irán creando de acuerdo a la accesibilidad dada por cada uno de los encargados de cada sección con respecto al grado de confidencialidad de la información.

*htpasswd -c /usr/local/etc/httpd/usuarios intranet*

Cuando se ejecute este comando aparecerá el *prompt* dos veces para escribir la contraseña y para confirmarla. Para agregar más usuarios o modificarlos no es necesario escribir la opción *-c*, sólo se escribirá como se muestra a continuación:

*htpasswd /usr/local/etc/httpd/usuarios paco*

Después de agregar algunos usuarios, el contenido del archivo `/usr/local/etc/httpd/usuarios` aparecerá de la siguiente manera:

```
#more /usr/local/etc/httpd/usuarios
imecas:9lIQlbwY3A/tA
paco:rx6Mzzx.Nk0fo
intranet:cV0qRdPrSgxFQ
```

Listado 13. Archivo *usuarios*.

Como se puede observar en el listado, el primer campo es el nombre del usuario y el segundo es la contraseña encriptada.

### Configuración del servidor

Para decirle al servidor que use los nombres de usuario y las contraseñas del archivo antes creado, es necesario configurarlo. Esta es una sección de la estructura de archivos que será restringida a algunos o a todos los usuarios listados en el archivo `usuarios`. Esto es típicamente hecho desde un directorio base, protegiendo un directorio y todos sus subdirectorios. Las directivas para crear un área protegida pueden ser puestas en un archivo llamado `.htaccess` en el directorio que se desea proteger, o en la sección `<Directory>` del archivo de configuración `access.conf`.

Para poner acceso restringido en un directorio con un archivo `.htaccess`, es necesario asegurarse que el archivo de configuración `access.conf` permite la autenticación de usuarios por medio de este archivo. Esto es controlado por la etiqueta `AuthConfig`. El archivo `access.conf` deberá incluir `AllowOverride AuthConfig` para permitir las directivas de autenticación a ser usadas en el archivo `.htaccess`.

Para restringir un directorio a cualquier cuenta listada en el archivo de usuarios creado, deberá crearse un archivo `.htaccess` conteniendo la siguiente información:

```
AuthName acceso restringido
AuthType Basic
AuthUserFile /usr/local/etc/httpd/usuarios
require valid-user
```

Listado 14. Directivas de un archivo *.htaccess*.

La primera directiva especifica el nombre que se le da al área a proteger, puede ser cualquier cosa. Una vez que el usuario ha dado una cuenta y contraseña valida, cualquier otro recurso con el mismo nombre de protección puede ser accesado con la misma cuenta y contraseña. Esto puede ser usado para crear dos áreas que compartan la misma cuenta y contraseña.

La directiva **AuthType** le dice al servidor que protocolo está siendo usado para la autenticación de usuarios. Por el momento, **Basic** es el único método disponible. Sin embargo un nuevo método, **Digets**, está muy próximo a ser estandarizado, y dará más seguridad que la autenticación **Basic**.

**AuthUserfile** le dice al servidor la localización del archivo de usuarios creado por **htpasswd**. Una directiva similar, **AuthGroupFile**, puede ser usada para decirle al servidor la localización del archivo de grupos de usuarios, si es que este se está usando, esto se explicará a detalle más adelante.

Entre estas cuatro directivas le dicen al servidor donde encontrar los usuarios, las contraseñas y el protocolo de autenticación a usar. El servidor con esto sabe que recursos están restringidos y sólo usuarios validos podrán acesar. La última directiva le dice al servidor que usuarios del archivo son validos para métodos particulares de acceso. Esto es hecho con la directiva **requiere**. En el siguiente ejemplo, el argumento **valid-user** le dice al servidor que cualquier cuenta en el archivo de usuarios puede ser usado. Pero puede ser configurado para permitir sólo el acceso a ciertos usuarios. Para el caso de la Intranet del INE sólo se tiene por el momento acceso con la cuenta Intranet en el directorio raíz de documentos (*/opt/data*), con la siguiente directiva:

***requiere user intranet***

Las demás cuentas se irán creando de acuerdo a las restricciones que desee cada uno de los responsables de las secciones. A continuación se presenta la ventana que se presenta para pedir la autenticación de usuario.

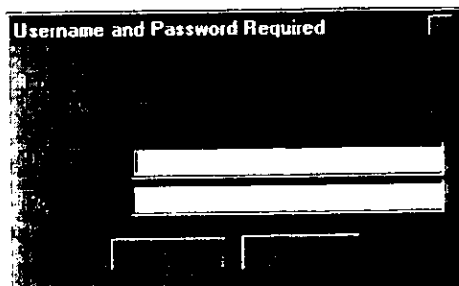


Figura 31. Ventana de autenticación de usuario.

Con esta directiva sólo se permite el acceso a la cuenta Intranet (después de haber escrito la contraseña correcta). Si cualquier otro usuario intenta acceder a este directorio aunque se encuentre en el archivo de usuarios y haya dado la contraseña correcta, le será denegado el acceso. Esto es útil para restringir diferentes áreas del servidor a diferentes usuarios con el mismo archivo de usuarios. Si un usuario tiene el acceso permitido a diferentes áreas, entonces sólo tendrá que recordar una sola contraseña. Hay que recordar que si la directiva **AuthName** es diferente en las distintas áreas, el usuario deberá escribir otra vez su contraseña.

### Usando grupos

Otra característica permitida para restringir el acceso a ciertos directorios es el uso de grupos de usuarios. Si se quiere permitir el acceso a sólo un grupo selecto de usuarios a cierta información, se pueden listar todas las cuentas permitidas en la línea de la directiva **requiere**. Sin embargo esto significa que la información de los usuarios en los archivos **.htaccess** está creciendo, y podría no ser conveniente si hay demasiados usuarios. Pero afortunadamente hay una forma de lograr esto, usando un archivo de grupos.

Esto opera muy similar a la forma en que operan los grupos en UNIX estándar: cualquier usuario puede ser miembro de cualquier número de grupos. Por lo tanto se puede usar la directiva **requiere** para restringir a los usuarios en uno más grupos en particular. Por ejemplo, se podría crear un grupo llamado **staff** conteniendo las cuentas de los usuarios que tienen permitido acceder paginas internas. Para permitir el acceso sólo a usuarios en el grupo **staff** se deberá usar la siguiente directiva:

*requiere group staff*

Múltiples grupos pueden ser listados, y una sola directiva **requiere user** también puede ser dada, en cualquier caso cualquier usuario en cualquiera de los grupos listados, o cualquier usuario listado específicamente, podrá acceder a los archivos de ese directorio. Por ejemplo:

*requiere group staff admin*

*requiere user paco*

Con esto se permite que cualquier usuario en el grupo **staff** o **admin**, o el usuario **paco**, accederá a los archivos en el directorio después de haber dado una contraseña valida.

Por último, el archivo de grupos consiste de líneas con un nombre de grupo seguida por la lista de usuarios separados por espacios. Por ejemplo:

*staff:imecas cesar*

*admin:intranet paco*

La directiva **AuthGroupFile** es usada para decirle al servidor la localización del archivo que contiene los grupos. Cabe señalar que la longitud máxima de una línea en el archivo de grupos es aproximadamente de 8,000 caracteres (8 Kb). Si se excede la capacidad de los caracteres permitidos, se puede tener más de una línea con el mismo nombre de grupo en el mismo archivo.

### 5.6.3 Mantenimiento

Como se ha visto el instalar un servidor de Web no es un proceso difícil, prueba de ello es la proliferación de organizaciones que tienen instalados ya servidores. El problema es, una vez instalado el servidor, mantenerlo funcionando, cuidar aspectos de seguridad y crear las páginas que le darán vida al servidor. Habrá arquitecturas para las que sea más difícil que el proceso aquí mostrado; sin embargo el código fuente del servidor de Apache y PHP viene ya preconfigurado para una amplia variedad de máquinas y sistemas operativos, lo que facilita el trabajo.

La Intranet instalada se puede separar en tres grandes tareas desde el punto de vista del mantenimiento:

- El servidor SUN donde reside el servidor Web

El mantenimiento específico de este servidor SUN está a cargo del Departamento de Administración de UNIX, por lo tanto, ellos son los que se encargarán de verificar que la máquina siempre se encuentre funcionando las 24 hrs. del día.

- Los programas instalados

Se instaló un servidor Web (Apache) con soporte para un lenguaje de programación de CGI's (PHP) con diversas utilidades. Como se ha visto a lo largo de la realización de este proyecto, cada proceso que necesita algún tipo de atención, se ha automatizado al máximo. Dada la naturaleza de Internet, el constante desarrollo de las Intranets y de los servicios que ofrecen, los programas que se han instalado siguen en desarrollo constante, por lo que aparecen nuevas versiones, nuevas normas y nuevas especificaciones e implantaciones de los protocolos. El servidor de Web y el lenguaje de programación instalados han ido cambiando de versión y actualizándose constantemente, ya que se incluyen mejoras de rendimiento y nuevas utilidades. Todo esto indica que en un plazo de tiempo estimado por el administrador de la Intranet el *software*, tanto del servidor de Web como del lenguaje de programación, deberán actualizarse para incorporar las nuevas versiones. Para ello sólo será necesario visitar cada cierto tiempo las páginas de donde se obtuvo el *software* de Apache y de PHP para conseguir la última versión. Todo lo explicado hasta

el momento es válido y funciona correctamente pero debemos de conocer las características nuevas si queremos aprovechar el *software* al máximo.

- La información publicada

La información por omisión debe ser actualizada constantemente, cualquier cambio debe verse reflejado en las páginas, ya que de lo contrario no se cumple uno de los objetivos de este proyecto, el de tener información actualizada. También es necesario aclarar que la actualización de las páginas estará dada por cada uno de los responsables y encargados de proporcionar la información de cada una de las secciones.



## *CONCLUSIONES*

---

El plan de estudios de la carrera de Ingeniería en Computación de la Escuela Nacional de Estudios Profesionales "Aragón" de la UNAM es muy completo, el profesorado y las instalaciones permiten realizar prácticas que contribuyen a reforzar los conocimientos adquiridos en los salones de clase, pero lo más importante es que la formación profesional se adquiere con la práctica. Gracias a este proyecto se pudieron poner a prueba los conocimientos adquiridos, los cuales dieron la base teórica necesaria para tener una formación ingenieril amplia.

Este ambicioso proyecto requirió también de los conocimientos y experiencias adquiridas en la Dirección de Computo para la Administración Académica (DCAA) a través de la Dirección General de Servicios de Cómputo Académico (DGSCA) y del apoyo del personal de la Unidad de Sistemas e Informática (USI) del Instituto Nacional de Ecología.

Gracias a lo investigado y a los conocimientos adquiridos al desarrollar la Intranet del INE, se puede decir que las tecnologías que se están desarrollando están enfocadas a la aplicación de redes internas, debido a que los protocolos TCP/IP permiten la interacción de distintos sistemas operativos y arquitecturas. Macintosh, IBM, o las estaciones de trabajo SUN pueden todas estar conectadas a la misma red Intranet y la comunicación es transparente. Actualmente, TCP/IP está integrado en los sistemas operativos Windows 95/NT, OS/2, OpenStep (NeXT), MacOS y cualquiera de las versiones UNIX (Solaris, Linux, etc.), además de versiones shareware para DOS. Las redes Novell con sistema operativo Netware también soportan TCP/IP. Esto significa que teniendo instalada una red con acceso a Internet, una red interna es mucho más fácil de implantar y satisface todas las necesidades de conectividad e interoperabilidad.

Como es posible notar a través del desarrollo de este proyecto, TCP/IP se ha convertido en el protocolo estándar de redes de comunicación. Esto le permite acceso a un mayor número de aplicaciones, sistemas y arquitecturas para la Intranet, lo que da flexibilidad en planes futuros y la

seguridad de que todo lo hecho será compatible con aplicaciones que se desarrollen a corto y largo plazo.

Una ventaja fundamental de la Intranet es que ahora las empresas pueden construir con total facilidad aplicaciones personalizadas a las que tienen un acceso inmediato los usuarios de cualquier parte de la Intranet, en cualquier plataforma. Puesto que el desarrollo y el despliegue son mucho más rápidos, el ahorro de costos con respecto a los antiguos modelos centrados en el escritorio es considerable. Más aún, los usuarios no necesitan una extensa formación. Las aplicaciones se ejecutan en la familiar interfaz de un navegador y los usuarios ya conocen las reglas: "¡señalar y hacer click!"

Se puede destacar que la revolución de Internet/Intranet ha abierto el camino para que las empresas usen sistemas abiertos y tecnologías para poner en marcha sus entornos de red sin sacrificar funciones ni robustez, en comparación con las alternativas propietarias tradicionales, gracias a lo cual tenemos productos y sistemas que hacen que esto sea posible.

Recordando cual fue la problemática del INE, y no sólo de este caso en particular, se puede decir que muchas empresas tienen similares necesidades de intercambio de información y aplicaciones, y se dan cuenta que, con el uso de herramientas sencillas y de dominio público implantadas en un entorno de Intranet, resuelven fácilmente estas carencias; esto es, utilizan las características y ventajas que proporciona una Intranet como un medio para desarrollar y desplegar nuevas aplicaciones de una forma mucho más rápida, sin problemas de sincronización de información y de una manera universal.

El desarrollo de este proyecto es un ejemplo de cómo una organización puede utilizar eficazmente las tecnologías Internet existentes para implantar soluciones prácticas, es decir, una Intranet. Todos los empleados del Instituto Nacional de Ecología tendrán a su alcance un procedimiento sencillo para acceder a la información interna, que podrán consultar, editar, imprimir, etc., mediante una interfaz eficaz y cómoda, como es el cliente WWW que está disponible en cada puesto de trabajo.

La Intranet del INE y en general las Intranets no están restringidas a redes locales, pues los protocolos utilizados por Internet están diseñados para funcionar tanto en LAN's como en WAN's. Si las delegaciones de la SEMARNAP en el D.F. y otras ciudades del interior del país construyen sus propias Intranets, con el tiempo se pueden coordinar para llegar a tener una Intranet a nivel más amplio, es decir, independientemente del lugar de la república en donde se encuentren las dependencias de la SEMARNAP podrán acceder a la información.

Se está consciente de que a los usuarios que por años han usado otro medio para el intercambio de información, les costará un poco más de esfuerzo adaptarse a consultar la información, que antes les llegaba en papel, en un visualizador de páginas Web; ya que no están familiarizados con el empleo de este nuevo mecanismo. Por lo tanto, es necesario inculcar esta nueva cultura informática en los procesos de información, haciéndolo de la manera más transparente y amigable posible; logrando con esto, una mejor y más rápida adaptación del personal al uso de la Intranet.

Cabe mencionar que también existen los usuarios que ya han tenido alguna experiencia o contacto con navegadores, y la migración hacia la Intranet, les será mucho más sencilla; pues ya han consultado información a través de un paginador, con la única diferencia que ésta era externa pero utilizando el mismo procedimiento. A pesar de todas las ventajas y desventajas que puedan argumentarse, la Intranet del Instituto Nacional de Ecología funciona satisfaciendo todas las necesidades manifestadas por la comunidad del Instituto. Los alcances de este proyecto están dados por los límites en *software* y *hardware* vigentes del Instituto, pero no por esto podemos decir que no sean adaptables a las nuevas tecnologías que adquiera el Instituto.

Queda claro que la Intranet del Instituto brinda más y mejores servicios, sólo se requiere de la difusión adecuada a la comunidad institucional para que con la participación de cada uno de ellos este nuevo mecanismo gane auge, reconocimiento y confianza para que la información publicada sea veraz y oportuna para cumplir y agilizar más sus tareas y nivel de desempeño.

Aunque el proyecto que se presenta está terminado, aún puede crecer a medida de que los recursos tecnológicos y las necesidades del Instituto Nacional de Ecología lo demanden, y como

## *BIBLIOGRAFÍA*

### **Libros y revistas**

- Evans, Tim  
Construya su propia Intranet  
Mexico : Prentice-Hall, 1997
- Frisch, Aeleen  
Essential System Administration  
O'Reilly & Associates, Inc.
- McCarthy, Linda  
Intranet security: stories from the trenches  
California : Sun Microsystems, c1998
- Eckel, George y Steen, William  
Intranet working  
Indianapolis, Ind. : New Riders, c1996
- Carabias Lilo, Julia y Quadri de la Torre, Gabriel  
Principios, orientaciones y agenda de trabajo del Instituto Nacional de Ecología  
México, D.F. primera edición, octubre 1995
- Contreras Alcalá, Felipe H.  
CGI: La programación para HTML  
Revista Soluciones Avanzadas  
No. 37, septiembre 1995
- Iturriaga Velázquez, Claudia y M. Germán, Daniel  
El conjunto de protocolos TCP/IP  
Revista Soluciones Avanzadas  
No. 33, mayo 1996
- Aguilera Sierra, Alejandro  
GNU no es UNIX  
Revista Soluciones Avanzadas  
No. 27, noviembre 1995

- Aguilera Sierra, Alejandro  
GNU no es UNIX 2ª parte: Los programas  
Revista Soluciones Avanzadas  
No. 37, septiembre 1996
- López-Ortiz, Alejandro  
Intranet: Internet en su organización  
Revista Soluciones Avanzadas  
No. 33, mayo 1996
- Martín Zamboni, Diego  
Seguridad en UNIX  
Revista Soluciones Avanzadas  
No. 35, julio 1996
- Casselberry, Rick  
Runing a perfect Intranet  
Indianapolis, Indiana : Que, c1996
- Hunt, Craig  
TCP/IP Network administration  
O'Reilly & Associates, Inc. 1992

#### URL's

- ANU Epub – PHP/FI Help Page  
<http://www.anu.edu.au/web/authors/php/>  
Agosto 1998
- Apache Project  
<http://www.apache.org/>  
Agosto 1998
- Apache Week  
<http://www.apacheweek.com/>  
Septiembre 1998
- Apache Week: 18<sup>th</sup> October 1996  
<http://www.apacheweek.com/issues/96-10-18>  
Septiembre 1998

- Aprenda la red  
<http://www.learnthenet.com/spanish/html/41intra.htm>  
Agosto 1998
- Artículos – Intranet  
<http://sunserv.fei.uv.mx/gaceta/principal.html>  
Agosto 1998
- Bit 99. Intranet: Un nuevo concepto para el manejo de la información corporativa  
<http://www.iies.es/teleco/bit/bit99/intranet.htm>  
Agosto 1998
- PHP scripts  
<http://www.bi.org/php/>  
Septiembre 1998
- Breve historia del Internet  
<http://mayaweb.upr.clu.edu/sysnet/internet/internet.htm>  
Julio 1998
- Cabletron's MMAC Products  
<http://www.cabletron.com/mmac/>  
Julio 1998
- CGI with PHP  
<http://triton.towson.edu/~schmitt/671/labs/php.html>  
Agosto 1998
- Construcción de una Intranet  
<http://www.redestb.es/inode/construc.htm>  
Agosto 1998
- Construyendo una Intranet  
<http://www.interplanet.es/enterprise/Soluciones/intranet/welcome.htm>  
Agosto 1998
- Curso de HTML en Brockman y Schunh  
<http://www.brockman.com.mx/cursoshtml/curhtml.htm>  
Julio 1998

- Documento Intranet  
<http://www.esi.us.es/~godo/intra.html>  
Agosto 1998
- Documentos para administración de sistemas  
<http://www.users.red3i.es/~efraim/docs.tecnicos/>  
Julio 1998
- Dynamic Web Pages with PHP3 (Web Techniques, February 1998)  
<http://www.webtechniques.com/features/1998/02/lerdorf/lerdorf.shtml>  
Agosto 1998
- Especial BIT 105: Impacto de Internet en las corporaciones: Intranet y Extranets  
<http://www.iies.es/teleco/bit/bit105/jsaras.htm>  
Julio 1998
- Estudios de casos Intranet  
[http://www25.netscape.com/es/comprod/at\\_work/customer\\_profiles/](http://www25.netscape.com/es/comprod/at_work/customer_profiles/)  
Septiembre 1998
- Frequently asked Questions About Intranets  
<http://www.innergy.com/ifaq.html>  
Julio 1998
- Glosario de términos en el Internet  
<http://www.computextos.com.pe/Manuales/glosa1.htm>  
Julio 1998
- GNU's Not Unix! – The project and the Free Software Foundation (FSF)  
<http://www.gnu.org/>  
Julio 1998
- Guía de Internet: CYBERSPACE  
<http://www.nauta.es/cd/guia/guide.toc.htm>  
Julio 1998
- Internet: Conceptos básicos  
<http://www.utralca.cl/InsMF/Cursos-Internet/seccion1.html>  
Julio 1998

- Inter@merica /Redes Intranet  
<http://www.interamerica.co.cr/espanol/intranet/>  
Agosto 1998
- Instituto Nacional de Ecología  
<http://www.ine.gob.mx/>  
Julio 1998
- Intranet  
<http://www.geocities.com/WallStreet/1962/intranet.html>  
Agosto 1998
- Intranet  
<http://www.make.com.ar/intranet.htm>  
Agosto 1998
- Introducción a TCP/IP  
<http://cimmerica.uc3m.es/~tatou/proyecto/node7.html>  
Julio 1998
- Manual de HTML  
<http://sunserv.fei.uv.mx/tips/manual/principal.html>  
Julio 1998
- Manual de HTML  
<http://www.smcoe.k12.ca.us/ssfusd/as/priscilla/manual.htm>  
Julio 1998
- MentorWeb wForum  
<http://www.mentorweb.net/wforum/index.html>  
Agosto 1998
- Netcraft Web Server Survey  
<http://www.netcraft.com/Survey/>  
Septiembre 1998
- Nuevas Tecnologías  
<http://www.spider-net.com/sinapsys/Tecnologia.htm>  
Agosto 1998



- PC News & Report  
<http://204.215.214.198/>  
Agosto 1998
- Preguntas y respuestas más comunes sobre el freeware  
<http://main.omni.net.mx/sitiomni/freeware/faqfree.html>  
Julio 1998
- PHP Chat  
<http://www.cei.net/~jackson/>  
Agosto 1998
- PHP/FI resource page  
<http://sighbear.com/Components/index.shtml>  
Julio 1998
- PHP3: PHP: Hypertext Preprocessor  
<http://www.php.net/>  
Septiembre 1998
- Qué es Intranet  
<http://www.redestb.es/inode/quees.htm>  
Agosto 1998
- ¿Qué es el Shareware?  
<http://www.uned.es/csi/software/freeinfo.htm>  
Julio 1998
- Scripts e exemples de PHP  
<http://setubal.com/iwa/info/phpscripts.phtml>  
Agosto 1998
- Seguridad en redes y sistemas  
<http://www.rediris.es/rediris/boletin/35/enfoque2.html>  
Julio 1998
- Server Watch Top 10 Web Servers  
<http://serverwatch.internet.com/top10.html>  
Julio 1998

- Sociedad Internet de México  
<http://www.isocmex.org.mx/>  
Julio 1998
- Software libre  
<http://www.ati.es/PUBLICACIONES/novatica/1997/126/nv126pres.html>  
Julio 1998
- Sun Microsystems  
<http://www.sun.com/>  
Julio 1998
- The Intranet Journal (sm) – Intranet 101  
<http://www.intranetjournal.com/newbie.html>  
Julio 1998
- W-Agora: Web publishing and forum software  
<http://www.araxe.fr/w-agera/>  
Agosto 1998
- WebServer Compare: The definitive guide to HTTP specs  
<http://webservercompare.internet.com/>  
Julio 1998
- Welcome to NCSA  
<http://www.ncsa.uiuc.edu/>  
Julio 1998
- World Wide Web  
<http://www.vinet.net.mx/soporte/faq/www/>  
Julio 1998

## ANEXOS

## Anexo 1. Estructura de archivos y directorios de PHP

Total 3548								
Drwxr-xr-x	15	root	other	3584	Aug	5	18:24	.
Drwxr-xr-x	10	root	other	512	Jun	12	20:41	..
-rw-rw-rw-	1	root	other	331	May	8	11:22	BUGS
-rw-rw-rw-	1	root	other	20150	Jun	6	09:05	CHANGES
-rw-rw-rw-	1	root	other	3705	May	26	14:56	CODING_STANDARDS
-rw-rw-rw-	1	root	other	17982	Sep	10	1997	COPYING
-rw-rw-rw-	1	root	other	2830	Jun	5	08:37	CREDITS
-rw-rw-rw-	1	root	other	33579	Jun	5	10:53	ChangeLog
-rw-rw-rw-	1	root	other	2980	May	29	15:53	EXTENSION_STATUS
-rw-rw-rw-	1	root	other	7777	Feb	23	10:20	FUNCTION_LIST.txt
-rw-rw-rw-	1	root	other	5173	Jun	2	08:44	INSTALL
-rw-rw-rw-	1	root	other	2873	May	23	05:30	LICENSE
-rw-r--r--	1	root	other	31201	Jun	8	20:59	Makefile
-rw-rw-rw-	1	root	other	31455	Jun	6	10:42	Makefile.in
-rw-rw-rw-	1	root	other	1801	Jun	5	08:27	README.QNX
-rw-rw-rw-	1	root	other	2378	May	22	10:07	README.WIN32
-rw-rw-rw-	1	root	other	3730	Mar	22	08:02	TODO
-rw-rw-rw-	1	root	other	914	Oct	23	1997	WISHLIST
-rw-rw-rw-	1	root	other	2750	May	30	12:04	WISHLIST-3.1
-rw-rw-rw-	1	root	other	4787	Jun	4	07:13	acconfig.h
-rw-rw-rw-	1	root	other	9190	May	14	09:53	aclocal.m4
-rw-rw-rw-	1	root	other	9763	May	29	10:25	alloc.c
-rw-rw-rw-	1	root	other	4911	May	22	13:18	alloc.h
-rw-rw-rw-	1	root	other	252	Mar	29	10:58	apMakefile.libdir
-rw-rw-rw-	1	root	other	2246	May	25	06:08	apMakefile.tmpl
-rw-rw-rw-	1	root	other	12956	May	20	11:53	apache.dsp
-rw-rw-rw-	1	root	other	20174	May	15	04:56	apidoc.txt
drwxr-xr-x	2	root	other	512	Jun	6	10:47	benchmarks
-rw-rw-rw-	1	root	other	17592	May	15	05:00	bison.simple
-rw-r--r--	1	root	other	4332	Jun	8	20:59	build-defs.h
-rw-rw-rw-	1	root	other	4043	Apr	29	07:22	build-defs.h.in
-rw-rw-rw-	1	root	other	4376	Mar	26	23:49	calendar.dsp
-rw-r--r--	1	root	other	5804	Jun	8	20:59	config.cache
-rwxrwxrwx	1	root	other	21543	Nov	1	1997	config.guess
-rw-r--r--	1	root	other	10980	Jun	8	20:59	config.h
-rw-rw-rw-	1	root	other	10514	Jun	5	08:27	config.h.in
-rw-r--r--	1	root	other	23637	Jun	8	20:59	config.log
-rwxr-xr-x	1	root	other	30492	Jun	8	20:59	config.status
-rwxrwxrwx	1	root	other	19461	Nov	1	1997	config.sub
-rw-rw-rw-	1	root	other	9126	Jun	5	08:57	config.w32.h
-rw-r--r--	1	root	other	32199	Jun	6	10:47	configuration-parser.tab.c
-rw-r--r--	1	root	other	180	Jun	6	10:47	configuration-parser.tab.h
-rw-rw-rw-	1	root	other	9865	Jun	1	14:34	configuration-parser.y
-rw-r--r--	1	root	other	43694	Jun	6	10:47	configuration-scanner.c
-rw-rw-rw-	1	root	other	3760	Jun	3	15:22	configuration-scanner.lex
-rwxr-xr-x	1	root	other	155603	Jun	6	10:47	configure
-rw-rw-rw-	1	root	other	37756	Jun	6	07:46	configure.in

-rw-rw-rw-	1	root	other	8050	May	22	13:18	constants.c
-rw-rw-rw-	1	root	other	2118	May	22	13:18	constants.h
-rw-rw-rw-	1	root	other	3302	May	16	16:03	control_structures.h
-rw-rw-rw-	1	root	other	52295	May	25	14:56	control_structures_inline.h
drwxr-xr-x	2	root	other	512	Jun	6	10:47	convertor
-rw-rw-rw-	1	root	other	4112	Mar	28	14:44	crypt.dsp
-rw-rw-rw-	1	root	other	1905	May	29	19:59	cvsusers
drwxr-xr-x	2	root	other	1024	Jun	8	20:59	dbase
-rw-rw-rw-	1	root	other	4266	Mar	26	23:49	dbase.dsp
-rw-rw-rw-	1	root	other	4173	Mar	26	23:49	dbm.dsp
-rw-rw-rw-	1	root	other	12383	May	30	17:18	debugger.c
drwxr-xr-x	8	root	other	512	Jun	6	10:47	dl
-rwxr-xr-x	1	root	other	120	Jun	8	20:57	do-conf
drwxr-xr-x	5	root	other	1024	Jun	8	20:59	doc
drwxr-xr-x	3	root	other	512	Jun	6	10:47	examples
drwxr-xr-x	5	root	other	512	Jun	6	10:47	extra
-rw-rw-rw-	1	root	other	4011	Mar	26	23:49	filepro.dsp
-rw-rw-rw-	1	root	other	73	Sep	25	1997	footer
-rw-rw-rw-	1	root	other	22071	Jun	1	01:06	fopen-wrappers.c
-rw-rw-rw-	1	root	other	3467	May	23	11:18	fopen-wrappers.h
drwxr-xr-x	2	root	other	3584	Aug	5	18:23	functions
-rw-rw-rw-	1	root	other	4154	May	11	17:01	gd.dsp
-rw-rw-rw-	1	root	other	3851	May	21	17:56	getopt.c
-rw-rw-rw-	1	root	other	200	May	11	14:17	getopt.h
-rw-rw-rw-	1	root	other	24574	May	23	15:10	hash.c
-rw-rw-rw-	1	root	other	6769	May	16	16:03	hash.h
-rw-rw-rw-	1	root	other	2059	Apr	29	07:22	header
-rw-rw-rw-	1	root	other	3758	May	25	14:56	highlight.c
-rw-rw-rw-	1	root	other	419	Nov	21	1997	highlight.h
-rw-rw-rw-	1	root	other	4203	Mar	31	18:24	imap4.dsp
-rw-rw-rw-	1	root	other	4772	Sep	10	1997	install-sh
-rw-rw-rw-	1	root	other	18059	May	29	15:53	internal_functions.c
-rw-rw-rw-	1	root	other	5655	May	21	11:28	internal_functions.h
-rw-rw-rw-	1	root	other	2620	May	15	04:56	internal_functions_registry.h
-rw-r--r--	1	root	other	134424	Jun	6	10:47	language-parser.tab.c
-rw-r--r--	1	root	other	1868	Jun	6	10:47	language-parser.tab.h
-rw-rw-rw-	1	root	other	31387	May	30	12:01	language-parser.y
-rw-r--r--	1	root	other	166434	Jun	6	10:47	language-scanner.c
-rw-rw-rw-	1	root	other	2586	Apr	29	07:22	language-scanner.h
-rw-rw-rw-	1	root	other	24042	May	25	14:56	language-scanner.lex
-rw-rw-rw-	1	root	other	4051	Mar	26	23:49	ldap.dsp
-rw-r--r--	1	root	other	322	Jun	8	20:59	libphp3.module
-rw-rw-rw-	1	root	other	465	Jun	1	03:58	libphp3.module.in
-rw-rw-rw-	1	root	other	3939	May	11	14:17	list.c
-rw-r--r--	1	root	other	0	Aug	5	18:24	listaphp.txt
-rwxrwxrwx	1	root	other	28502	Nov	1	1997	ltconfig
-rw-rw-rw-	1	root	other	47164	Nov	1	1997	ltrimain.sh
-rw-rw-rw-	1	root	other	66786	Jun	4	11:00	main.c
-rw-rw-rw-	1	root	other	3503	May	23	07:06	main.h
-rwxrwxrwx	1	root	other	2868	Mar	22	08:02	makedist
-rwxrwxrwx	1	root	other	201	May	9	16:39	makeparser.bat
-rw-rw-rw-	1	root	other	17766	Jun	2	08:28	mod_php3.c
-rw-rw-rw-	1	root	other	3531	May	9	17:17	mod_php3.h
-rw-rw-rw-	1	root	other	3878	May	16	16:03	modules.h

-rw-rw-rw-	1	root	other	4088	Apr	5	13:25	msql.dsp
-rw-rw-rw-	1	root	other	3967	Apr	5	13:25	msql1.dsp
-rwxrwxrwx	1	root	other	210	Jan	24	1998	mtparser.bat
-rw-rw-rw-	1	root	other	4194	Mar	28	14:44	mysql.dsp
-rw-rw-rw-	1	root	other	30189	Jun	5	01:11	operators.c
-rw-rw-rw-	1	root	other	4765	May	15	04:56	operators.h
-rw-rw-rw-	1	root	other	16602	Jun	4	11:00	php.h
-rw-rw-rw-	1	root	other	12862	May	22	09:52	php3.dsp
-rw-rw-rw-	1	root	other	3119	May	23	11:18	php3.dsw
-rw-rw-rw-	1	root	other	9284	May	22	08:39	php3.ini-dist
-rw-rw-rw-	1	root	other	6	Dec	21	1997	php3.opt
-rw-rw-rw-	1	root	other	2995	May	15	04:56	php3_debugger.h
-rw-rw-rw-	1	root	other	3534	May	16	16:03	php3_list.h
-rw-rw-rw-	1	root	other	2462	Apr	29	07:22	php3_sprintf.c
-rw-rw-rw-	1	root	other	1684	May	21	17:56	php3_threads.c
-rw-rw-rw-	1	root	other	1217	Mar	26	19:42	php3_threads.h
-rw-rw-rw-	1	root	other	757	May	22	10:07	php3extra.dsw
-rw-rw-rw-	1	root	other	1540	May	27	18:28	php3sapi.dsw
-rw-rw-rw-	1	root	other	26	Jun	5	10:53	php_version.h
-rw-rw-rw-	1	root	other	12990	May	11	14:17	phpmysql.dsp
-rw-rw-rw-	1	root	other	3793	May	23	11:18	phpsapi_cgi.dsp
-rw-rw-rw-	1	root	other	13183	May	25	13:05	phpsapi_core.dsp
-rw-rw-rw-	1	root	other	4252	May	27	18:28	phpsapi_isapi.dsp
-rw-rw-rw-	1	root	other	3732	May	27	18:28	phpsapi_nsapi.dsp
drwxr-xr-x	2	root	other	1024	Jun	8	20:59	regex
-rw-rw-rw-	1	root	other	9350	May	21	17:56	request_info.c
-rw-rw-rw-	1	root	other	2597	Apr	29	07:22	request_info.h
-rw-rw-rw-	1	root	other	4465	May	23	11:18	safe_mode.c
-rw-rw-rw-	1	root	other	169	May	23	11:18	safe_mode.h
drwxr-xr-x	3	root	other	512	Jun	6	10:47	serverapi
-rwxrwxrwx	1	root	other	17987	Jun	5	08:27	setup
-rw-rw-rw-	1	root	other	4047	Mar	31	18:24	snmp.dsp
-rw-rw-rw-	1	root	other	21629	May	11	14:17	snprintf.c
-rw-rw-rw-	1	root	other	2502	Apr	29	07:22	snprintf.h
-rw-rw-rw-	1	root	other	3633	May	11	14:17	stack.c
-rw-rw-rw-	1	root	other	2565	Apr	29	07:22	stack.h
-rw-r--r--	1	root	other	0	Jun	8	20:59	stamp-h
-rw-rw-rw-	1	root	other	0	Oct	17	1997	stamp-h.in
drwxr-xr-x	12	root	other	512	Jun	6	10:47	test
drwxr-xr-x	2	root	other	512	Jun	6	10:47	tests
-rw-rw-rw-	1	root	other	3764	May	21	17:56	tls.c
-rw-rw-rw-	1	root	other	6968	May	22	13:18	tls.h
-rw-rw-rw-	1	root	other	13522	May	15	04:56	token_cache.c
-rw-rw-rw-	1	root	other	3769	May	15	04:56	token_cache.h
-rw-rw-rw-	1	root	other	20635	May	29	16:16	variables.c
-rw-rw-rw-	1	root	other	4382	May	15	04:56	variables.h
drwxr-xr-x	2	root	other	512	Jun	6	10:47	win32
-rw-rw-rw-	1	root	other	1729	May	31	22:16	win95nt.h

## Anexo 2. Listado de la ejecución del script setup

```
#!/setup
```

```
*****
*
* Welcome to the PHP 3.0 setup script. Use this script if you do not
* want or know how to use the configure program. See the INSTALL file
* for further installation instructions.
*
* You will now be asked a series of questions for your installation.
* For each question, your options will be shown in parantheses, and
* for some a default value is listed shown in brackets.
*
*****
```

Whether to build PHP as an Apache module. If you are running Apache, building PHP as a module will give better performance and security. If you answer no PHP will be built as a CGI program. The CGI version also enables Apache users to run different PHP3-enabled pages under different user-ids.  
If you answer 'yes', the default directory is '/usr/local/etc/httpd'.

Build as an Apache module? ('yes', 'no' or dir) [no] : yes  
Enter Apache base install directory [/usr/local/etc/httpd] : /opt/src/apache\_1.3.0

Whether to build PHP as fhttpd module. If you are running fhttpd, building PHP as a module will give better performance, more control and remote execution capability. More info about fhttpd can be found at <http://phobos.illtel.denver.co.us/pub/fhttpd/>.  
If you answer 'yes', the default directory is '/usr/local/src/fhttpd'.

Build as an fhttpd module? ('yes', 'no' or dir) [no] : no

Whether to include GD support. If PHP can find the GD libraries on your system, it will be included automatically. You should enter something here if you do not want to include GD support or if you have installed GD in some unusual directory.  
If you answer 'yes', the default directory is '/usr'.

GD support? ('yes', 'no' or dir) [yes] : yes  
Enter GD install directory [/usr] : /usr/lib

Whether to build PHP with Oracle support. Has been confirmed to work with Oracle versions 7.0 to 7.3. If you have not set up your Oracle environment, enter what \$ORACLE\_HOME is usually set to here. More info about Oracle can be found at <http://www.oracle.com/>.  
If you answer 'yes', the default directory is 'Oracle'.

Oracle support? ('yes', 'no' or dir) [no] : yes  
Enter home directory [Oracle] : /oracle/app/oracle/product/7.3.2

Whether to build PHP with iODBC support. This feature was first developed for iODBC Driver Manager, a freely redistributable ODBC driver manager which runs under many flavors of UNIX.

More info about iODBC can be found on the FreeODBC Pages at <http://users.ids.net/~bjepson/freeODBC/>.

If you answer 'yes', the default directory is '/usr/local'.

iODBC support? ('yes', 'no' or dir) [no] : no

Whether to build with Adabas D support.

More info about Adabas D can be found at <http://www.adabas.com/>.

If you answer 'yes', the default directory is '/usr/local'.

Whether to build with Adabas D support.

More info about Adabas D can be found at <http://www.adabas.com/>.

If you answer 'yes', the default directory is '/usr/local'.

Adabas D support? ('yes', 'no' or dir) [no] : no

Whether to build PHP with Sybase support (DBLib only).

More info about Sybase can be found at <http://www.sybase.com/>.

If you answer 'yes', the default directory is '/home/sybase'.

Sybase support? ('yes', 'no' or dir) [no] : no

Whether to build PHP with Sybase-CT support.

If you answer 'yes', the default directory is '/home/sybase'.

Sybase-CT support? ('yes', 'no' or dir) [no] : no

Whether to build PHP with MySQL support.

More info about MySQL can be found at <http://www.tcx.se/>.

If you answer 'yes', the default directory is '/usr/local'.

MySQL support? ('yes', 'no' or dir) [no] : no

Whether to build PHP with mSQL support. PHP supports both mSQL 1.0 and mSQL 2.0. However, if you build PHP with mSQL 1.0 libraries, you will only be able to access mSQL 1.0 databases, ditto for mSQL 2.0.

More info about mSQL can be found at <http://www.hughes.com.au/>.

If you answer 'yes', the default directory is '/usr/local/Hughes'.

mSQL support? ('yes', 'no' or dir) [no] : no

Whether to build PHP with PostgreSQL support.

More info about PostgreSQL can be found at <http://www.postgreSQL.org/>.

If you answer 'yes', the default directory is '/usr/local/pgsql'.

PostgreSQL support? ('yes', 'no' or dir) [no] : no

Whether to build PHP with Solid support.

More information about Solid can be found at <http://www.solidtech.com/>.

If you answer 'yes', the default directory is '/usr/local/solid'.

Solid support? ('yes', 'no' or dir) [no] : no

Whether to build with LDAP (Lightweight Directory Access Protocol) support.

More information about LDAP can be found in RFC1777 and RFC1778. If you answer 'yes', the default directory is '/usr/local/ldap'.

LDAP support? ('yes', 'no' or dir) [no] : no

Whether to build PHP with Velocis support.

More information about Velocis can be found at <http://www.raima.com/>. If you answer 'yes', the default directory is '/usr/local/velocis'.

Velocis support? ('yes', 'no' or dir) [no] : no

Whether to build PHP with CODBC support. This feature was first developed for Sybase SQL Anywhere 5.5 on QNX, but may be used for any unknown ODBC driver on all flavors of UNIX.

If you answer 'yes', the default directory is '/usr/local'.

custom ODBC support? ('yes', 'no' or dir) [no] : no

Whether to use the bundled dbase library.

dBase support? (yes/no) [no] : no

Whether to use the bundled filePro library. Read-access only.

filePro support? (yes/no) [no] : no

Directory where the PHP3 configuration file (php3.ini) is located.

If you answer 'yes', the default directory is '/usr/local/lib'.

Default config directory? ('yes', 'no' or dir) [yes] : no

Whether to use the systems regular expression library rather than the bundled one. If you are building PHP3 as a server module, you must use the same library when building PHP3 as when linking the server. Enable this if the systems library provides special features you need. It is recommended that you use the bundled library if possible.

Use the system regex library? (yes/no) [no] : no

Whether to enable debug information. Answering "no" here will make PHP run faster, but it will be harder to trace bugs. You are encouraged to leave debugging on while PHP 3.0 is in alpha and beta state.

Compile with debug information? (yes/no) [yes] : no

Whether to enable PHP safe mode. This imposes several restrictions on what PHP can do, such as opening only files within



the document root. Read the Security chapter of the documentation for more information. CGI users should always enable secure mode. This only sets the default, it may be enabled or disabled in the configuration file later.

Enable safe mode by default? (yes/no) [no] : **no**

The directory where executables that may be run from safe mode are located.

Default safe mode exec dir? (dir) [/usr/local/bin] : **/usr/local/bin**

Enable the HTTP\_GET\_VARS, HTTP\_POST\_VARS and HTTP\_COOKIE\_VARS arrays by default (can be turned on or off in the configuration file).

Enable variable tracking by default? (yes/no) [no] : **no**

Whether to magic quotes by default. This can be changed in the configuration file.

Enable magic quotes by default? (yes/no) [no] : **no**

Whether to enable PHP remote debugging support. This feature is still under development.

Enable PHP remote debugger? (yes/no) [no] : **no**

Enables bc style arbitrary precision math functions.

Enable bc style precision math functions (yes/no) [no] : **no**

Compile with memory limit support.

Enable memory limit? (yes/no) [no] : **no**

Whether to enable the short form of the PHP HTML embed tags.

The short form is "<? code; ?>", while the long form is "<?php code; ?>". If you plan on using XML on your site, you should disable the short form. This is the default and can be overridden in the configuration file.

Allow short tag by default? (yes/no) [yes] : **yes**

Enable the URL-aware fopen wrapper that allows accessing files via http or ftp.

Enable URL fopen wrappers? (yes/no) [yes] : **yes**

```
*****
*
*           Running configure...
*
*****
```

```
./configure --with-apache=/opt/src/apache_1.3.0 --with-gd=/usr/lib --with-oracle
=/oracle/app/oracle/product/7.3.2 --with-config-file-path=no --enable-debug=no
loading cache ./config.cache
```

```
checking whether build environment is sane... yes
checking whether to enable maintainer-specific portions of Makefiles... no
checking for bison... (cached) bison -y
checking bison version... 1.25 (ok)
checking for gcc... (cached) gcc
checking whether the C compiler (gcc ) works... yes
checking whether the C compiler (gcc ) is a cross-compiler... no
checking whether we are using GNU C... (cached) yes
checking whether gcc accepts -g... (cached) yes
checking for ranlib... (cached) :
checking whether gcc and cc understand -c and -o together... (cached) yes
checking for perl... (cached) /usr/local/bin/perl
checking for sendmail... (cached) /usr/lib/sendmail
checking for gdbm_open in -lgdbm... (cached) no
checking for dbm_open in -ldb... (cached) no
checking for dbm_open in -lc... (cached) yes
checking preferred dbm library... ndbm chosen
checking for socket in -lc... (cached) no
checking for socket in -lsocket... (cached) yes
checking for gethostbyaddr in -lc... (cached) no
checking for gethostbyaddr in -lnsl... (cached) yes
checking for crypt in -lc... (cached) yes
checking for dlopen in -lc... (cached) no
checking for dlopen in -ldl... (cached) yes
checking for sin in -lc... (cached) no
checking for sin in -lm... (cached) yes
checking for res_search in -lsocket... (cached) no
checking for res_search in -lresolv... (cached) yes
checking how to run the C preprocessor... (cached) gcc -E
checking for ANSI C header files... (cached) yes
checking for dirent.h that defines DIR... (cached) yes
checking for opendir in -ldir... (cached) no
checking for fclose declaration... ok
checking for fcntl.h... (cached) yes
checking for unistd.h... (cached) yes
checking for crypt.h... (cached) yes
checking for sys/file.h... (cached) yes
checking for memory.h... (cached) yes
checking for pwd.h... (cached) yes
checking for grp.h... (cached) yes
checking for sys/socket.h... (cached) yes
checking for sys/wait.h... (cached) yes
checking for syslog.h... (cached) yes
checking for string.h... (cached) yes
checking for sys/varargs.h... (cached) yes
checking for stdarg.h... (cached) yes
checking for sys/time.h... (cached) yes
checking for signal.h... (cached) yes
checking for netinet/in.h... (cached) yes
```

checking for dlfcn.h... (cached) yes  
checking for limits.h... (cached) yes  
checking for sys/types.h... (cached) yes  
checking for unix.h... (cached) no  
checking whether struct tm is in sys/time.h or time.h... (cached) time.h  
checking for tm\_zone in struct tm... (cached) no  
checking for tzname... (cached) yes  
checking for st\_blksize in struct stat... (cached) yes  
checking for st\_blocks in struct stat... (cached) yes  
checking for st\_rdev in struct stat... (cached) yes  
checking for size\_t... (cached) yes  
checking for uid\_t in sys/types.h... (cached) yes  
checking for uint... (cached) yes  
checking for vprintf... (cached) yes  
checking for memcpy... (cached) yes  
checking for memmove... (cached) yes  
checking for strdup... (cached) yes  
checking for strerror... (cached) yes  
checking for strcasecmp... (cached) yes  
checking for strstr... (cached) yes  
checking for flock... (cached) no  
checking for lockf... (cached) yes  
checking for putenv... (cached) yes  
checking for tempnam... (cached) yes  
checking for usleep... (cached) yes  
checking for setlocale... (cached) yes  
checking for gettimeofday... (cached) yes  
checking for setvbuf... (cached) yes  
checking for srand48... (cached) yes  
checking for lrand48... (cached) yes  
checking for srandom... (cached) yes  
checking for random... (cached) yes  
checking for link... (cached) yes  
checking for symlink... (cached) yes  
checking for regcomp... (cached) yes  
checking for getlogin... (cached) yes  
checking for cuserid... (cached) yes  
checking for vsnprintf... (cached) no  
checking for snprintf... (cached) no  
checking for gcvt... (cached) yes  
checking for utime... (cached) yes  
checking for crypt... (cached) yes  
checking for setitimer... (cached) yes  
checking for rint... (cached) yes  
checking for unsetenv... (cached) no  
checking for strftime... (cached) yes  
checking for setsockopt... (cached) yes  
checking for tzset... (cached) yes  
checking whether utime accepts a null argument... (cached) yes  
checking for working alloca.h... (cached) yes  
checking for alloca... (cached) yes  
checking for broken sprintf... ok  
checking for getopt... (cached) yes  
checking whether to use bundled regex library... yes

```

checking for Apache module support via DSO through APXS... no
checking for Apache module support via DSO through APACI... no
checking for Apache module support... yes - Apache 1.3.x
checking for httpd module support... no
checking whether to include GD support... checking for gdImageString16 in -lgd... (cached) yes
checking for TT_Open_Face in -ltof... (cached) no
checking for Oracle support... yes
checking Oracle version... 7.3
checking for iODBC support... no
checking for Adabas support... no
checking for Sybase support... no
checking for Sybase-CT support... no
checking for MySQL support... no
checking for mSQL support... no
checking for PostgreSQL support... no
checking for Solid support... no
checking for LDAP support... no
checking for Velocis support... no
checking for a custom ODBC support... no
checking for zlib support... no
checking whether to include the bundled dbase library... no
checking whether to include the bundled filePro support... no
checking whether to enable unified ODBC support... no
checking whether to use a configuration file... no
checking whether to include debugging symbols... no
checking whether to enable safe mode by default... no
checking for safe mode exec dir... /usr/local/php/bin
checking whether to enable track_vars variables by default... no
checking whether to enable magic quotes by default... no
checking whether to enable remote debugger support... no
checking whether to enable bc style precision math functions... yes
checking whether to enable a memory limit... no
checking whether to enable short tags by default... yes
checking whether to enable the URL-aware fopen wrapper... yes
creating ./config.status
creating Makefile
creating build-defs.h
creating stamp-h
creating libphp3.module
creating doc/Makefile
creating doc/version.ent
creating doc/checkdoc
creating extra/gd/bdf2gdfont
creating regex/Makefile
creating dbase/Makefile
creating doc/funcparse
creating config.h

```

```

*****
*
* Configuration options were saved in do-conf, if desired you can
* edit this file and then run it. Run ./configure --help for a list
* of options.
*

```

```

* Read the INSTALL file for more information and for help in      *
* configuring apache. See the web site at: http://www.php.net/ for *
* documentation and more. If desired copy the php3.ini-dist file to *
* your config directory/php3.ini and edit it.                    *
*                                                                  *
*****

```

### Anexo 3. Listado de la ejecución del comando make de PHP

```

# make

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c language-parser.tab.c -o language-
parser.tab.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -w -c language-scanner.c

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c main.c -o main.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c hash.c -o hash.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c operators.c -o operators.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c variables.c -o variables.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c token_cache.c -o token_cache.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c stack.c -o stack.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c internal_functions.c -o internal_functions.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c snprintf.c -o snprintf.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c php3_sprintf.c -o php3_sprintf.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c alloc.c -o alloc.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c list.c -o list.o

```

```

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c highlight.c -o highlight.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c debugger.c -o debugger.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c configuration-parser.tab.c -o configuration-
parser.tab.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -w -c configuration-scanner.c

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c request_info.c -o request_info.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c safe_mode.c -o safe_mode.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c fopen-wrappers.c -o fopen-wrappers.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c constants.c -o constants.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/adabasd.c -o functions/adabasd.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/apache.c -o functions/apache.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/fttpd.c -o functions/fttpd.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/basic_functions.c -o
functions/basic_functions.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/crypt.c -o functions/crypt.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/datetime.c -o functions/datetime.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/db.c -o functions/db.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/dbase.c -o functions/dbase.o

gcc -g -O2 -O2 -I. -I. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix -I/usr/lib/include -c functions/dir.c -o functions/dir.o

```







```

-l/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/url.c -o functions/url.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/base64.c -o functions/base64.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/info.c -o functions/info.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -w -c functions/number.c -o functions/number.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/bcmath.c -o functions/bcmath.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/unified_odbc.c -o
functions/unified_odbc.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/ldap.c -o functions/ldap.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/browscap.c -o
functions/browscap.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/velocis.c -o functions/velocis.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/gdttf.c -o functions/gdttf.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/gdcache.c -o functions/gdcache.o

gcc -g -O2 -O2 -f. -f. -I/oracle/app/oracle/product/7.3.2/rdbms/demo -I/opt/src/apache_1.3.0/src/include
-I/opt/src/apache_1.3.0/src/os/unix      -l/usr/lib/include -c functions/zlib.c -o functions/zlib.o

ar rc libmodphp3.a language-parser.tab.o language-scanner.o main.o hash.o operators.o variables.o
token_cache.o stack.o internal_functions.o snprintf.o php3_sprintf.o alloc.o list.o highlight.o debugger.o
configuration-parser.tab.o configuration-scanner.o request_info.o safe_mode.o fopen-wrappers.o constants.o
functions/adabasdb.o functions/apache.o functions/fhttpd.o functions/basic_functions.o functions/crypt.o
functions/datetime.o functions/db.o functions/dbase.o functions/dir.o functions/dl.o functions/dns.o
functions/exec.o functions/file.o functions/filepro.o functions/filestat.o functions/formatted_print.o
functions/fsock.o functions/gd.o functions/head.o functions/html.o functions/image.o functions/link.o
functions/mail.o functions/math.o functions/md5.o functions/microtime.o functions/mime.o functions/mysql.o
functions/mysqli.o functions/oracle.o functions/pack.o functions/pageinfo.o functions/pgsql.o functions/post.o
functions/rand.o functions/reg.o functions/solid.o functions/soundex.o functions/string.o functions/syslog.o
functions/type.o functions/uniqid.o functions/sybase.o functions/sybase-ct.o functions/url.o
functions/base64.o functions/info.o functions/number.o functions/bcmath.o functions/unified_odbc.o
functions/ldap.o functions/browscap.o functions/velocis.o functions/gdttf.o functions/gdcache.o
functions/zlib.o
: libmodphp3.a

```

## Anexo 4. Listado de la ejecución del comando make de Apache

```
# make
====> src
make[1]: Entering directory `/opt/src/apache_1.3.0'
make[2]: Entering directory `/opt/src/apache_1.3.0/src'
====> src/regex
gcc -I. -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX
-DSERVER_SUBVERSION=\"PHP/3.0\" `./apaci` -DPOIX_MISTAKE -c regcomp.c -o regcomp.o
gcc -I. -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX
-DSERVER_SUBVERSION=\"PHP/3.0\" `./apaci` -DPOIX_MISTAKE -c regexec.c -o regexec.o
gcc -I. -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX
-DSERVER_SUBVERSION=\"PHP/3.0\" `./apaci` -DPOIX_MISTAKE -c regerror.c -o regerror.o
gcc -I. -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX
-DSERVER_SUBVERSION=\"PHP/3.0\" `./apaci` -DPOIX_MISTAKE -c regfree.c -o regfree.o
rm -f librege.a
ar cr librege.a regcomp.o regexec.o regerror.o regfree.o
true librege.a
<==== src/regex
====> src/os/unix
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` os.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` os-inline.c
rm -f libos.a
ar cr libos.a os.o os-inline.o
true libos.a
<==== src/os/unix
====> src/ap
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` ap_execve.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` ap_cpystn.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` ap_signal.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` ap_slack.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` ap_snprintf.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` ap_strings.c
rm -f libap.a
ar cr libap.a ap_execve.o ap_cpystn.o ap_signal.o ap_slack.o ap_snprintf.o ap_strings.o
true libap.a
<==== src/ap
====> src/main
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\"
`./apaci` gen_test_char.c
gcc -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION=\"PHP/3.0\" `./apaci` -o gen_test_char
gen_test_char.o -lsocket -lnsl -Lmodules/php3 -L../modules/php3 -lmodphp3 -lresolv -lm -ldl -lnsl -lsocket
-L/oracle/app/oracle/product/7.3.2/lib -lcintsh -lsocket -lnsl -lm -ldl
./gen_test_char>test_char.h
```

```

gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`gen_uri_delims.c
gcc -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0" `../apaci` -o gen_uri_delims
gen_uri_delims.o -lsocket -lnsl -Lmodules/php3 -L../modules/php3 -lmodphp3 -lresolv -lm -ldl -lnsl -lsocket
-L/oracle/app/oracle/product/7.3.2/lib -lclntsh -lsocket -lnsl -lm -ldl
./gen_uri_delims >uri_delims.h
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`alloc.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`buff.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`http_config.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`http_core.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`http_log.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`http_main.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`http_protocol.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`http_request.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`http_vhost.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`util.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`util_date.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`util_script.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`util_uri.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`util_md5.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`md5c.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`rfc1413.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`fnmatch.c
rm -f libmain.a
ar cr libmain.a alloc.o buff.o http_config.o http_core.o http_log.o http_main.o http_protocol.o http_request.o
http_vhost.o util.o util_date.o util_script.o util_uri.o util_md5.o md5c.o rfc1413.o fnmatch.o
true libmain.a
<==== src/main
====> src/modules
====> src/modules/standard
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`mod_env.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`mod_log_config.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
  ../apaci`mod_mime.c
gcc -c -I./os/unix -I./include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"

```

```

../apaci` mod_negotiation.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_include.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_autoindex.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_dir.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_cgi.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_asis.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_imap.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_actions.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_userdir.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_alias.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_access.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_auth.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_setenvif.c
rm -f libstandard.a
ar cr libstandard.a mod_env.o mod_log_config.o mod_mime.o mod_negotiation.o mod_include.o mod_autoindex.o
mod_dir.o mod_cgi.o mod_asis.o mod_imap.o mod_actions.o mod_userdir.o mod_alias.o mod_access.o mod_auth.o
mod_setenvif.o
true libstandard.a
<=== src/modules/standard
==> src/modules/php3
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` mod_php3.c
cp libmodphp3.a libphp3.a
ar r libphp3.a mod_php3.o
true libphp3.a
<=== src/modules/php3
<=== src/modules
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` modules.c
gcc -c -I../os/unix -I../include -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0"
../apaci` buildmark.c
gcc -DSOLARIS2=251 -DUSE_HSREGEX -DSERVER_SUBVERSION="\PHP/3.0" ../apaci` \
-o httpd buildmark.o modules.o modules/standard/libstandard.a modules/php3/libphp3.a main/libmain.a
../os/unix/libos.a ap/libap.a regex/libregex.a -lsocket -lnsl -Lmodules/php3 -L../modules/php3 -lmodphp3 -lresolv -
lm
-ldl -lnsl -lsocket -L/oracle/app/oracle/product/7.3.2/lib -lcintsh -lsocket -lnsl -lm -ldl -lgd
make[2]: Leaving directory `/opt/src/apache_1.3.0/src'
make[1]: Leaving directory `/opt/src/apache_1.3.0'

```

## Anexo 5. Listado del archivo `httpd.conf`

```
# This is the main server configuration file. See URL http://www.apache.org/
# for instructions.

# Do NOT simply read the instructions in here without understanding
# what they do, if you are unsure consult the online docs. You have been
# warned.

# Originally by Rob McCool

# ServerType is either inetd, or standalone.

ServerType standalone

# If you are running from inetd, go to "ServerAdmin".

# Port: The port the standalone listens to. For ports < 1023, you will
# need httpd to be run as root initially.

Port 80

# HostnameLookups: Log the names of clients or just their IP numbers
# e.g. www.apache.org (on) or 204.62.129.132 (off)
# You should probably turn this off unless you are going to actually
# use the information in your logs, or with a CGI. Leaving this on
# can slow down access to your site.
HostnameLookups on

# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.

# User/Group: The name (or #number) of the user/group to run httpd as.
# On SCO (ODT 3) use User nouser and Group nogroup
# On HPUX you may not be able to use shared memory as nobody, and the
# suggested workaround is to create a user www and use that user.

User nobody
Group #nobody

# The following directive disables keepalives and HTTP header flushes for
# Netscape 2.x and browsers which spoof it. There are known problems with
# these

BrowserMatch Mozilla/2 nokeepalive

# ServerAdmin: Your address, where problems with the server should be
# e-mailed.

ServerAdmin admonweb@ine.gob.mx

# ServerRoot: The directory the server's config, error, and log files
```

```
# are kept in
ServerRoot /usr/local/etc/httpd

# BindAddress: You can support virtual hosts with this option. This option
# is used to tell the server which IP address to listen to. It can either
# contain "*", an IP address, or a fully qualified Internet domain name.
# See also the VirtualHost directive.

#BindAddress *

# ErrorLog: The location of the error log file. If this does not start
# with /, ServerRoot is prepended to it.

ErrorLog logs/error_log

# TransferLog: The location of the transfer log file. If this does not
# start with /, ServerRoot is prepended to it.

TransferLog logs/access_log

# PidFile: The file the server should log its pid to
PidFile logs/httpd.pid

# ScoreBoardFile: File used to store internal server process information.
# Not all architectures require this. But if yours does (you'll know because
# this file is created when you run Apache) then you *must* ensure that
# no two invocations of Apache share the same scoreboard file.
ScoreBoardFile logs/apache_status

# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e. use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.

ServerName www.ine.gob.mx

# CacheNegotiatedDocs: By default, Apache sends Pragma: no-cache with each
# document that was negotiated on the basis of content. This asks proxy
# servers not to cache the document. Uncommenting the following line disables
# this behavior, and proxies will be allowed to cache the documents.

#CacheNegotiatedDocs

# Timeout: The number of seconds before receives and sends time out

Timeout 300

# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
```

```
KeepAlive On

# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.

MaxKeepAliveRequests 100

# KeepAliveTimeout: Number of seconds to wait for the next request

KeepAliveTimeout 15

# Server-pool size regulation. Rather than making you guess how many
# server processes you need, Apache dynamically adapts to the load it
# sees --- that is, it tries to maintain enough server processes to
# handle the current load, plus a few spare servers to handle transient
# load spikes (e.g., multiple simultaneous requests from a single
# Netscape browser).

# It does this by periodically checking how many servers are waiting
# for a request. If there are fewer than MinSpareServers, it creates
# a new spare. If there are more than MaxSpareServers, some of the
# spares die off. These values are probably OK for most sites ---

MinSpareServers 5
MaxSpareServers 10

# Number of servers to start --- should be a reasonable ballpark figure.

StartServers 5

# Limit on total number of servers running, i.e., limit on the number
# of clients who can simultaneously connect --- if this limit is ever
# reached, clients will be LOCKED OUT, so it should NOT BE SET TOO LOW.
# It is intended mainly as a brake to keep a runaway server from taking
# Unix with it as it spirals down...

MaxClients 150

# MaxRequestsPerChild: the number of requests each child process is
# allowed to process before the child dies.
# The child will exit so as to avoid problems after prolonged use when
# Apache (and maybe the libraries it uses) leak. On most systems, this
# isn't really needed, but a few (such as Solaris) do have notable leaks
# in the libraries.

MaxRequestsPerChild 30

# Proxy Server directives. Uncomment the following line to
# enable the proxy server:

#ProxyRequests On

# To enable the cache as well, edit and uncomment the following lines:
```

```

#CacheRoot /usr/local/etc/httpd/proxy
#CacheSize 5
#CacheGcInterval 4
#CacheMaxExpire 24
#CacheLastModifiedFactor 0.1
#CacheDefaultExpire 1
#NoCache a_domain.com another_domain.edu joes.garage_sale.com

# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the VirtualHost command

#Listen 3000
#Listen 12.34.56.78:80

# VirtualHost: Allows the daemon to respond to requests for more than one
# server address, if your server machine is configured to accept IP packets
# for multiple addresses. This can be accomplished with the ifconfig
# alias flag, or through kernel patches like VIF.

# Any httpd.conf or srm.conf directive may go into a VirtualHost command.
# See also the BindAddress entry.

#<VirtualHost host.some_domain.com>
#ServerAdmin webmaster@host.some_domain.com
#DocumentRoot /www/docs/host.some_domain.com
#ServerName host.some_domain.com
#ErrorLog logs/host.some_domain.com-error_log
#TransferLog logs/host.some_domain.com-access_log
#</VirtualHost>

```

## *Anexo 6. Listado del archivo srm.conf*

```

# With this document, you define the name space that users see of your http
# server. This file also defines server settings which affect how requests are
# serviced, and how results should be formatted.

# See the tutorials at http://www.apache.org/ for
# more information.

# Originally by Rob McCool; Adapted for Apache

# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.

DocumentRoot /opt/data

```



```
# UserDir: The name of the directory which is appended onto a user's home
# directory if a -user request is recieved.

UserDir public_html

# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index. Separate multiple entries with spaces.

DirectoryIndex index.html

# FancyIndexing is whether you want fancy directory indexing or standard
FancyIndexing on

# AddIcon tells the server which icon to show for different files or filename
# extensions

AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrm .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.

DefaultIcon /icons/unknown.gif

# AddDescription allows you to place a short description after a file in
# server-generated indexes.
```

```
# Format: AddDescription "description" filename

# ReadmeName is the name of the README file the server will look for by
# default. Format: ReadmeName name
#
# The server will first look for name.html, include it if found, and it will
# then look for name and include it as plaintext if found.
#
# HeaderName is the name of a file which should be prepended to
# directory indexes.

ReadmeName README
HeaderName HEADER

# IndexIgnore is a set of filenames which directory indexing should ignore
# Format: IndexIgnore name1 name2...

IndexIgnore */.??* *~* */HEADER* */README* */RCS

# AccessFileName: The name of the file to look for in each directory
# for access control information.

AccessFileName .htaccess

# DefaultType is the default MIME type for documents which the server
# cannot find the type of from filename extensions.

DefaultType text/plain

# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+) uncompress
# information on the fly. Note: Not all browsers support this.

AddEncoding x-compress Z
AddEncoding x-gzip gz

# AddLanguage allows you to specify the language of a document. You can
# then use content negotiation to give a browser a file in a language
# it can understand. Note that the suffix does not have to be the same
# as the language keyword --- those with documents in Polish (whose
# net-standard language code is pl) may wish to use "AddLanguage pl .po"
# to avoid the ambiguity with the common suffix for perl scripts.

AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage da .da
AddLanguage el .el
AddLanguage it .it

# LanguagePriority allows you to give precedence to some languages
# in case of a tie during content negotiation.
# Just list the languages in decreasing order of preference.

LanguagePriority en fr de
```

```
# Redirect allows you to tell clients about documents which used to exist in
# your server's namespace, but do not anymore. This allows you to tell the
# clients where to look for the relocated document.
# Format: Redirect fakename url

# Aliases: Add here as many aliases as you need (with no limit). The format is
# Alias fakename realname

# Note that if you include a trailing / on fakename then the server will
# require it to be present in the URL. So "/icons" isn't aliased in this
# example.

#Alias /icons/ /usr/local/etc/httpd/icons/

# ScriptAlias: This controls which directories contain server scripts.
# Format: ScriptAlias fakename realname

ScriptAlias /cgi-bin/ /opt/data/cgi-bin/

# If you want to use server side includes, or CGI outside
# ScriptAliased directories, uncomment the following lines.

# AddType allows you to tweak mime.types without actually editing it, or to
# make certain files to be certain types.
# Format: AddType type/subtype ext1

AddType application/x-httpd-php .html
AddType application/x-httpd-php .htm
AddType application/x-httpd-php .phtml

phpUploadTmpDir /opt/data/tmp
phpShowInfo off
phpIncludePath /opt/data:/opt/data/uca/p/
phpMaxDataSpace 16535

# AddHandler allows you to map certain file extensions to "handlers",
# actions unrelated to filetype. These can be either built into the server
# or added with the Action command (see below)
# Format: AddHandler action-name ext1

# To use CGI scripts:
AddHandler cgi-script .cgi

# To use server-parsed HTML files
#AddType text/html .shtml
#AddHandler server-parsed .shtml

# Uncomment the following line to enable Apache's send-asis HTTP file
# feature
#AddHandler send-as-is asis
```

```

# If you wish to use server-parsed imagemap files, use
#AddHandler imap-file map

# To enable type maps, you might want to use
#AddHandler type-map var

# Action lets you define media types that will execute a script whenever
# a matching file is called. This eliminates the need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location

# MetaDir: specifies the name of the directory in which Apache can find
# meta information files. These files contain additional HTTP headers
# to include when sending the document

#MetaDir .web

# MetaSuffix: specifies the file name suffix for the file containing the
# meta information.

#MetaSuffix .meta

# Customizable error response (Apache style)
# these come in three flavors
#
# 1) plain text
#ErrorDocument 500 "The server made a boo boo.
# n.b. the (") marks it as text, it does not get output
#
# 2) local redirects
#ErrorDocument 404 /missing.html
# to redirect to local url /missing.html
#ErrorDocument 404 /cgi-bin/missing_handler.pl
# n.b. can redirect to a script or a document using server-side-includes.
#
# 3) external redirects
#ErrorDocument 402 http://some.other_server.com/subscription_info.html
#

```

## *Anexo 7. Listado del archivo access.conf*

```

# access.conf: Global access configuration
# Online docs at http://www.apache.org/
# This file defines server settings which affect which types of services
# are allowed, and in what circumstances.
# Each directory to which Apache has access, can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
# Originally by Rob McCool

```

```
# This should be changed to whatever you set DocumentRoot to.

<Directory /opt/data>

# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".

# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you (or at least, not yet).

Options Indexes FollowSymLinks

# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"

AllowOverride AuthConfig

# Controls who can get stuff from this server.
order allow,deny
allow from all
</Directory>

# /usr/local/etc/httpd/cgi-bin should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
<Directory /opt/data/cgi-bin>
AllowOverride None
Options None
</Directory>

# Allow server status reports, with the URL of http://servername/server-status
# Change the ".your_domain.com" to match your domain to enable.
#<Location /server-status>
#SetHandler server-status
#order deny,allow
#deny from all
#allow from .your_domain.com
#</Location>

# There have been reports of people trying to abuse an old bug from pre-1.1
# days. This bug involved a CGI script distributed as a part of Apache.
# By uncommenting these lines you can redirect these attacks to a logging
# script on phf.apache.org. Or, you can record them yourself, using the script
# support/phf_abuse_log.cgi.

#<Location /cgi-bin/phf*>
#deny from all
#ErrorDocument 403 http://phf.apache.org/phf\_abuse\_log.cgi
#</Location>
# You may place any other directories or locations you wish to have
# access information for after this one.
```

## Anexo 8. Listado de la página principal de la Intranet del INE

```

<HTML>
<HEAD>
  <TITLE>Intranet - Instituto Nacional de Ecología</TITLE>
</HEAD>

<BODY TEXT="#000000" Bgcolor="#339900">

<IMG SRC="/logos/plecamaya.gif" HEIGHT=35 WIDTH="100%">

<CENTER>
<TABLE BORDER=3 CELLSPACING=0 CELLPADDING=8 WIDTH="100%" >

  <TR VALIGN=TOP>
    <TD BGCOLOR="#FFFFFF">
      <IMG SRC="/logos/title.gif" ALT="INE" HEIGHT=90 WIDTH=480>&nbsp;
    </TD>
  </TR>
</TABLE>

  <center>
  <br>
  <FONT FACE="Comic Sans MS" size=+1>
    Bienvenidos a la Intranet del INE
  </FONT>
  <HR WIDTH="270">
  </center>
  <BR>

  <center>
  <table border=0 cellpadding=5 cellspacing=10>
  <tr>
  <td>
  <A HREF="/intranet/directorio/directorio.phtml">
  <IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
  BORDER=0></A>Directorio
  </td>
  <td>
  <A HREF="/intranet/informativos/index.phtml">
  <IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
  BORDER=0></A>Resúmenes informativos
  </td>
  </tr>
  <tr>
  <td>
  <A HREF="/intranet/aplicaciones/index.phtml">
  <IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
  BORDER=0></A>Aplicaciones
  </td>
  <td>
  <A HREF="/intranet/capacitacion/index.phtml">
  <IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
  BORDER=0></A>Capacitación
  </td>
  </tr>
  </table>
  </center>

```

```

        <td>
        <A HREF="/intranet/metodologia/index.phtml">
        <IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
BORDER=0></A>Informática
        </td>
        <td>
        <A HREF="/intranet/foros/index.phtml">
        <IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
BORDER=0></A>Foros de discusión
        </td>
        </tr>
        <tr>
        <td>
        <A HREF="/intranet/salas/index.phtml">
        <IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
BORDER=0></A>Salas virtuales
        </td>
        <td>
        <A HREF="/intranet/clarin/index.phtml">
        <IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
BORDER=0></A>Clarín informático
        </td>
        </tr>

        </table>
        <BR><BR>
        </center>

</TD>

<TD ROWSPAN="0" BGCOLOR="#FFFFFF">
<IMG SRC="/logos/edif4t.gif" ALT="Edificio del INE" HEIGHT=214 WIDTH=188>&nbsp;
<BR>
La Intranet del INE es un medio interactivo para el intercambio de información, accesible a todo el
personal del instituto. Esta información puede ser colocada y mantenida por los propios
usuarios.

<br><br>

Si requiere más información acerca del uso de esta herramienta envíenos un
<A HREF="mailto:admonweb@ine.gob.mx">mensaje</a>.
</TD>
</TR>
</TABLE>
<IMG SRC="/logos/plecamaya.gif" HEIGHT=35 WIDTH=100% VSPACE=5>
</center>

<? include "pie.phtml";?>

```

**Anexo 9. Listado del archivo directorio.phtml**

```

<?
  $TituloPagina = "Directorio del INE";
  include "cabeza.phtml";
?>

<?if(!$pattern1);?>
<BR>
Para obtener informaci&oacute;n sobre el personal que labora en el
Instituto Nacional de Ecolog&iacute;a teclee los datos que conozca y el
bot&oacute;n buscar.

<BR><BR>
  <form action="directorio.phtml" method=POST>

<center>
<table border=0>
<tr>
<td align="right">Nombre: </td>
<td><input type="text" name="pattern1" size=50></td>
</tr>
<tr>
<td align="right">Cargo:</td>
<td> <Select Name="cargo">
<Option>Director General
<option>Asesor
<Option>Director de &Aacute;rea
<option>Subdirector
<option>Jefe de Departamento
<Option Selected>Otro
</select>
</td>
</tr>
<tr>
<td align="right">&Aacute;rea donde labora:</td>
<td>
<Select Name="area">
<Option>Coordinaci&oacute;n de Asesores
<option>Unidad Coordinadora de &Aacute;reas Naturales Protegidas
<Option>Direcci&oacute;n General de Vida Silvestre
<option>Direcci&oacute;n General de Ordenamiento e Impacto Ambiental
<option>Direcci&oacute;n General de Regulaci&oacute;n Ambiental
<option>Direcci&oacute;n General de Materiales, Residuos y Actividades Riesgosas
<option>Direcci&oacute;n General de Gesti&oacute;n e Informaci&oacute;n Ambiental
<option>Unidad de Asuntos Jur&iacute;dicos
<option>Unidad de Cr&eacute;dito Externo
<option>Unidad de Administraci&oacute;n
<option>Unidad de Sistemas e Inform&aacute;tica
<option>Unidad de Cooperaci&oacute;n y Convenios Internacionales
<option>Unidad de Sustancias Qu&iacute;micas y Evaluaci&oacute;n Ambiental

```



```

<option>Unidad de Comunicaci&ocute;n Social
<option>Unidad de Publicaciones
<Option Selected>Otro
</select>
</td>
</tr>
<tr><td></td><TD></td></tr>
<tr><td></td><TD></td></tr>
<tr><td></td><TD></td></tr>
<tr>
<TD></td>
<td align="right"><input type="submit" value=" Buscar "></td>
</tr>
</table>
</form>

<?else; ?>

    <h2 align=center>Resultados de la B&uacute;squeda</h2>
<BR><BR>
    Expresi&ocute;n buscada: <?echo $pattern1; ?>
    <BR><BR><BR><BR>
<?
$pattern2=strtoupper("$pattern1");

switch($area) {
    case "Coordinaci&ocute;n de Asesores";
        $area1="COORDINACION DE ASESORES";
        break;
    case "Unidad Coordinadora de &Aacute;reas Naturales Protegidas";
        $area1="UNIDAD COORD. DE AREAS NAT. PROTEG.";
        break;
    case "Direcci&ocute;n General de Vida Silvestre";
        $area1="DIRECCION GENERAL DE VIDA SILVESTRE";
        break;
    case "Direcci&ocute;n General de Ordenamiento e Impacto Ambiental";
        $area1="DIR. GRAL. DE ORD. E IMPACTO AMB.";
        break;
    case "Direcci&ocute;n General de Regulaci&ocute;n Ambiental";
        $area1="DIR. GRAL. DE REGULACION AMBIENTAL";
        break;
    case "Direcci&ocute;n General de Materiales, Residuos y Actividades Riesgosas";
        $area1="DIR. GRAL. DE RES., MAT. Y ACT. R.";
        break;
    case "Direcci&ocute;n General de gesti&ocute;n e informaci&ocute;n Ambiental";
        $area1="DIR. GRAL. DE GEST. E INF. AMB.";
        break;
    case "Unidad de Asuntos Jur&uacute;dicos";
        $area1="UNIDAD DE ASUNTOS JURIDICOS";
        break;
    case "Unidad de Cr&eacute;dito Externo";
        $area1="UNIDAD DE CREDITO EXTERNO";
        break;
    case "Unidad de Administraci&ocute;n";

```

```

    Sarea1="UNIDAD DE ADMINISTRACION";
    break;
    case "Unidad de Sistemas e Informática";
    Sarea1="UNIDAD DE SISTEMAS E INFORMATICA";
    break;
    case "Unidad de Cooperación y Convenios Internacionales";
    Sarea1="UNIDAD DE COOP. Y CONV. INT.";
    break;
    case "Unidad de Sustancias Químicas y Evaluación Ambiental";
    Sarea1="UNIDAD DE SUST. QUIM. Y EVAL. AMB.";
    break;
    case "Unidad de comunicación Social";
    Sarea1="UNIDAD DE COMUNICACIÓN SOCIAL";
    break;
    case "Unidad de Publicaciones";
    Sarea1=strtoupper("Sarea");
    break;
default;
    Sarea1="%";
    break;
}

```

```

switch($cargo) {
    case "Director General";
    $cargol=strtoupper("$cargo");
    break;
    case "Asesor";
    $cargol=strtoupper("$cargo");
    break;
        case "Director de Área";
    $cargol="Director de Area";
    break;
        case "Subdirector";
    $cargol=strtoupper("$cargo");
    break;
        case "Jefe de Departamento";
    $cargol=$cargo;
    break;
    default;
    $cargol="%";
    break;
}

```

```

PutEnv("ORACLE_HOME=/oracle/app/oracle/product/7.3.2");
PutEnv("ORACLE_SID=INE");

```

```

/* Establecer la conexion entre PHP y Oracle. */
$conn = Ora_Logon("directorio","directorio");

```

```

if ( $conn < 0) {
    include "mantenimiento.html";
    echo "ATTE. <br>Administraci&oacute;n de la p&aacute;gina del INE.\n";
    exit;
}

```

```

/* Abrir un cursor en Oracle. */
Scursor = Ora_Open($conn);

if ($Scursor < 0) {
    echo "No puedo abrir un cursor.\n";
    Ora_Logoff($conn);
    exit;
}

/* Apagar el autocommit. */
Ora_CommitOff($conn);

if ($Sareal=="%" && $Scargo!="%"){

/*Query de SQL*/
$query = "select
persona_general.nombre_comun,correo_electronico.direccion_correo,telefono.numero,relacion_persona_
fisica_moral.puesto,relacion_persona_fisica_moral.adscripcion from
persona_general,correo_electronico,telefono,relacion_persona_fisica_moral where
persona_general.nombre_comun=correo_electronico.nombre_comun(+) and
persona_general.nombre_comun=telefono.nombre_comun(+) and
persona_general.nombre_comun=relacion_persona_fisica_moral.nombre_comun(+) and
persona_general.nombre_comun like '%$pattern2%';
}else{

/* Query de SQL. */
$query="SELECT
persona_general.nombre_comun,correo_electronico.direccion_correo,telefono.numero,relacion_persona_
fisica_moral.puesto,relacion_persona_fisica_moral.adscripcion from
persona_general,correo_electronico,telefono,relacion_persona_fisica_moral where
persona_general.nombre_comun=correo_electronico.nombre_comun(+) and
persona_general.nombre_comun=telefono.nombre_comun(+) and
persona_general.nombre_comun=relacion_persona_fisica_moral.nombre_comun(+) and
persona_general.nombre_comun like '%$pattern2%' and relacion_persona_fisica_moral.puesto like
'$scargo1' and relacion_persona_fisica_moral.adscripcion like '$sareal';
}

if (Ora_Parse($Scursor, $query) < 0) {
    echo "Parse failed!\n";
    Ora_Logoff($conn);
    exit;
}

/* Ejecuta la declaración SQL asociada con $Scursor y prepara el almacenamiento para la información. */
$ncols = Ora_Exec($Scursor);
echo "<center>\n";
echo "<P>\n";
echo "<TABLE BORDER=1 CELLSPACING=0>\n";
echo "<TR>\n";
echo "<th>Nombre</th>\n";
echo "<th>Correo electr&ocute;nico</th>\n";
echo "<th>Tel&eacute;fono</th>\n";
echo "<th>Puesto</th>\n";

```

```

echo "<th>Adscripci&oacute;n</th>\n";
echo "</tr>\n";

/* Recupera los renglones de la base de datos uno a uno. */
while (Ora_Fetch($cursor) == 1) {
    $i = 0;
    echo "<TR>\n";
    while ($i < $ncols) {

        /* Obtiene los datos de la columna del actual renglón. */
        $col = Ora_GetColumn($cursor, $i);

        if ($i==0);
            echo "<TD>$col</TD>\n";
        endif;

        if ($i==1);
            echo "<TD><center><a href=\\"mailto:$col\">$col</a></center></TD>\n";
        endif;

        if ($i==2 || $i==3 || $i==4);
            echo "<TD><center>$col</center></TD>\n";
        endif;

        $i++;
    }
    echo "</TR>\n";
}

echo "</TABLE>\n";
echo "</center>\n";

/* Cierra la conexión a Oracle. */
Ora_Close($cursor);

/* Se desconecta. */
Ora_Logoff($conn);
?>
<BR><BR><BR>
<hr size=5>

<? Include "pie.phtml": ?>

```

## Anexo 10. Listado del archivo de mensajes personalizado form.var

```

<?
$msg[0]="No puedo acceder el archivo de configuracion";
$msg[1]="No puedo acceder la base de datos";
$msg[2]="El documento no existe";
$msg[3]="Debe especificar un documento";
$msg[4]="Por favor, da tu nombre";
$msg[5]="Titulo : Campo obligatorio";
$msg[6]="Por favor, inserta el texto ";
$msg[7]="No puedo crear el archivo $note";
$msg[8]="No puedo acceder $server_file ";
$msg[9]="No puedo insertar el documento";
$msg[10]="No estas autorizado para suprimir";
$msg[11]="<em><center><h2>Lista de documentos</h2></em></center><hr>";
$msg[12]="Password incorrecto";
$msg[13]="No puedo suprimir";
$msg[14]="El documento ha sido borrado";
$msg[15]="No puedo acceder el directorio : $dir";
$msg[16]="Nuevo Password es obligatorio";
$msg[17]="Antiguo Password es obligatorio";
$msg[18]="Documento raiz es obligatorio";
$msg[19]="El directorio del foro es obligatorio";
$msg[20]="El documento del directorio es obligatorio";
$msg[21]="Archivo de base de datos es obligatorio";
$msg[22]="<h2 align=center>Resultados de la busqueda en <i>$pattern</i></h2>";
$msg[23]="Problema de creacion, checa permisos y espacio en el sistema de archivos";
$msg[24]="<center><em>El foro (<i>$bn_name</i>) ha sido modificado</em></center>";
$msg[25]="<h2><em><center>directorio : $dir</center></em></h2><hr> ";
$msg[26]="Buscar ";
$msg[27]="Buscar en foro";
$msg[28]=" Empezar busqueda ";
$msg[29]="<p align=center><h2>No hay resultados!</h2>";
$msg[30]="<p><center><h2>$num documentos encontrados</h2></center>";
$msg[31]="Documento";
$msg[32]="Tamano";
$msg[33]="Relevancia";
$msg[34]="Enviar";
$msg[35]="Limpiar";
$msg[36]="Cancelar";
$msg[37]="<center><b>Archivo $pcfile_name cargado ($pcfile_size bytes transferidos)</b></center>";
$msg[38]="<center><b>El archivo $enclosure ha sido copiado </b></center>";
$msg[39]="<center><b>Gracias, el mensaje ha sido agregado al servidor. </b></center>";
$msg[40]="<em><center><h2>Suprimir un documento</h2></em></center><hr>";
$msg[41]="Password :";
$msg[42]=" Suprimir ";
$msg[43]=" Previo ";
$msg[44]=" Salir ";
$msg[45]="<center><em>La base (<i>$bn_name</i>) ha sido creada </em></center>";
$msg[46]="Por favor, escribe tu password :";
$msg[47]="Tu browser no soporta frames.<br>. Baja <a href=http://www.microsoft.com>Internet Explorer version 3 (o mayor)</a> or <a href=http://home.netscape.com>Netscape version 2 (or greater)</a><br> para ver este documento";

```

```

$msg[48]="Acceso prohibido";
$msg[49]="<h3>Deseas suprimir el siguiente documento? </h3><br>";
$msg[50]=" Responder ";
$msg[51]=" Regresar a la lista ";
$msg[52]="Por favor, contacte a el <a href=mailto:$bn_mail_address>administrador</a>";
$msg[53]="Cargar el archivo en tu estacion : ";
$msg[54]="Pasarl al archivo del servidor $SERVER_NAME (ruta completa)";
$msg[55]="Examinar...";
$msg[56]="Debes llenar este campo. Gracias";
$msg[57]="<i>Documentos relacionados</i> : ";
$msg[58]=" Abrir ";
$msg[59]="<center><b>El documento $note ha sido modificado</b></center>";
$msg[60]="Editar";
$msg[61]=" Lista de notas";
$msg[62]=" Poner mensaje ";
$msg[63]=" Suprimir una nota";
$msg[64]=" Buscar ";
$msg[65]=" Lista de notas ";
?>

```

## Anexo 11. Listado del script chat.phtml

```

<html><head><title>Salas interactivas!</title>
</head>

<frameset rows="120,*">
<frame name="input" src="input.phtml" scrolling="off">
<frame src="refresh.phtml" name="refresh">
</frameset>

<?
$enters=0;
$leaves=0;

/** User-defined channels **/
/** Read from channels.txt **/
  $channelinput[]=file("channels.txt");
  $numchannels[]=count($channelinput);
  $i=0;
  while ($i<$numchannels);
    $chan[$i] = strtok($channelinput[$i], "=");
  $inputfiles[$i] = strtok ("=");
    $i++;
  endwhile;

/** Set up defaults **/
  $drefresh = 30;
  $dchannel = $chan[0];
  if (!$enters);
    $enters=0;

```

```

endif;
if (!$leaves);
    $leaves=0;
endif;

/** Check to see if defaults need to be used **/
if (!$refresh);
    $refresh = $drefresh;
endif;
if (!$channel);
    $channel= $dchannel;
endif;
/** Replace spaces in channel name to pass with URL **/
    $chancode=reg_replace(" ","+",$channel);

/** Channels **/
    $i=0;
    while ($i<$numchannels);
        if ($channel==$chan[$i]);
            $inputfile = $inputfiles[$i];
        endif;
        if ($oldchannel==$chan[$i]);
            $oldinputfile = $inputfiles[$i];
        endif;
        $i++;
    endwhile;

/** Check to see if they're changing the channel **/
if (!$oldchannel);
else;
if ($oldchannel!=$channel);
    $enters=1;
    $leaves=1;
endif;
endif;
/** Fix the Nicks **/
if ($nick);
    $nick = reg_replace ("[^a-zA-Z0-9. ]","",$nick);
    $nickcode = reg_replace(" ","+",$nick);
endif;
>
<META HTTP-EQUIV=REFRESH CONTENT="<? echo $refresh;>; URL=chat.phtml<?
echo "?refresh=";
echo $refresh;
echo "&channel=";
echo $channel;
if ($nick);
echo "&nick=";
echo $nickcode;
endif;>">

<? /** If there's a comment, post it **/
if (!$chat);
else;

```

```

$nick = reg_replace("[<]", "", $nick);
$chat = reg_replace("[<]", "", $chat);
if (strlen($chat) > 1000);
$chat = "<i>Est&acute; interesante...</i>";
endif;
$chatemp = "";
$stok = strtok($chat, " ");
$chatlen = strlen($chat);
while($chatlen > 0);
$stoken = reg_replace("(^[Hh][Tt][Pp]:.*\^[Mm][Aa][Ll]
|[Ll][Tt][Oo]:.*\^[Ff][Tt][Pp]:.*", "<a href=$stok target=_blank>$stok</a>", $stok);
$chatemp = $chatemp . " " . $stoken;
$chatlen = $chatlen - 1 - strlen($stok);
$stok = strtok(" ");
endif;
$chat = $chatemp;

$string = "\n<b><font color=#000000>" . $nick . "</font></b><i> ";
$string = $string + date("D h:i") + " :</i> ";

if ($bold);
    $string = $string + "<b>";
endif;
if ($italics);
    $string = $string + "<i>";
endif;
if ($big);
    $string = $string + "<font size=+1>";
endif;

$string = $string + $chat;
if ($big);
    $string = $string + "</font>";
endif;
if ($italics);
    $string = $string + "</i>";
endif;
if ($bold);
    $string = $string + "</b>";
endif;
if (!$semoticon);
    $semoticon = "none";
endif;

if ($semoticon=="happy");
    $string = $string + "&#160; <b>:-)</b>";
endif;

if ($semoticon=="wink");
    $string = $string + "&#160; <b>;-)</b>";
endif;

if ($semoticon=="sad");

```



```

        Sstring = Sstring + "&#160; <b>:-(</b>";
    endif;

    Sfp = fopen("Sinputfile","a");
    fputs("$fp","$string");
    fclose ($fp);
endif;
>
<head>
<title><? echo $nick;> Platicando</title>
</head>
        <body background="/logos/fondo.jpg">
<p><center>
<?

/** Check to see if this is first login **/
/** If so, log the user name, email, and remote host **/
if ($login==1);
    if (!$email);
        $nick = "";
    else;
        Sfp = fopen("users.log","a");
        Srestuff = "\n" + date ("D h:i") + ": " + $nick + "|" + $email;
        Srestuff = Srestuff + " from " + $REMOTE_HOST;
        fputs("$fp","$restuff");
        fclose ($fp);
        $enters = 1;
        $leaves = 0;
    endif;
endif;

if ($enters==1);
    Sfp = fopen("Sinputfile","a");
    Shelllothere = "\n<b><font color=#000000>" + $nick + "</font></b><i> " ;
    Shelllothere = Shelllothere + date ("D h:i") + ": * Entro *</i>";
    fputs("$fp","$shelllothere");
    fclose ($fp);

    endif;

if ($leaves==1);
    Sfp = fopen("$oldinputfile","a");
    Shelllothere = "\n<b><font color=#000000>" + $nick + "</font></b><i> " ;
    Shelllothere = Shelllothere + date ("D h:i") + ": * Salio del ";
    Shelllothere = Shelllothere + $channel + " *</i>";
    fputs("$fp","$shelllothere");
    fclose ($fp);

    endif;

if (!$nick);
    echo "<form action=chat.phtml method=post>";
    echo "<input type=hidden name=login value=1>";

```

```

echo "Escribe un seud&oacute;nimo: ";
echo "<input name=nick><br>";
    echo "Escribe tu direcci&oacute;n de correo: ";
echo "<input name=email><br>";
echo "<input type=submit value=Entrar!>";
echo "</form>";
else;
/** Ok, they're in. Show chat interface. **/
/** First form: Change Channel and refresh rate **/
echo "<form action=chat.phtml method=get>";
echo "<input type=hidden name=oldchannel";
echo " value=\"$Schannel\">";
echo "<input type=hidden name=nick value= >"
<? echo $nick;><?
echo ">";
echo "<a href=index.html target=_parent>Salir</a>";
echo "&#160;&#160; &#160;&#160;";
echo "<b><font color=#000000>";
echo $nick;
echo "</font></b>";
echo "&#160;&#160; &#160;&#160;";

/** This shows the channel selections
and selects the correct channel **/

echo "Canal:";
echo "<select name=channel>";
$i=0;
while ($i<$numchannels);
echo "<option";
if ($Schannel==$chan[$i]);
    echo " selected";
endif;
echo ">";
echo $chan[$i];
$i++;
endwhile;
echo "</select>";
echo " <input type=submit value=Cambiar> ";

/** Refresh Rates **/
echo "Refrescar cada (seg): <input type=radio name=refresh value=15> 15";
echo "<input type=radio name=refresh value=30 checked> 30 ";
echo "<input type=radio name=refresh value=60> 60 ";
echo "</form> ";

/** Second form: Sala de discusion interactiva! **/
echo "<form action=chat.phtml method=get>";
echo "<input type=text name=chat size=65%>";
echo "<input type=hidden name=channel value=\"$Schannel\">";
echo "<input type=hidden name=refresh value=$refresh>";
echo " <input type=submit value=Entrar!><br>";

echo "<b>Options:</b> <input type=checkbox name=bold>Negrilla ";

```

```

echo " <input type=checkbox name=italics>Itálicas ";
echo " <input type=checkbox name=big>Grandes &#160;&#160;&#160;&#160;";

echo " <input type=radio name=emoticon value=happy>Feliz ";
echo " <input type=radio name=emoticon value=wink>Gui&ntilde;ar ";
echo " <input type=radio name=emoticon value=sad>Triste ";

echo "<input type=hidden name=nick value=";>
    "<? echo $nick;>"
    "<? echo ">";
echo "</form>";

endif;

echo "</center></td></tr></table>";

/** Read in the chatfile **/
    Schatfile]=file("$inputfile");

/** Show the last 15 lines in reverse order **/
    Schatsize = count ($chatfile);
    echo $chatfile[$Schatsize];
    Sstop = $Schatsize - 15;
    while($Schatsize > Sstop);
        Schatsize = $Schatsize -1;
        echo $chatfile[$Schatsize];
        echo "<br>";
    endwhile;
/** Trim the file if it gets too big **/

/** You could also save it off as a log... **/

    $fz = fileSize("$inputfile");
    if ($fz > 20000);
    $fp = fopen("$inputfile","w");
    Sstop2 = $stop + 15;
    while($stop != $stop2);
        $restuff = "\n" + $chatfile[$stop];
        fputs("$fp","$restuff");
        $stop++;
    endwhile;
    fclose ($fp);

/** Make a log file **/
/** Commented Out!
    Schatsize = count ($chatfile);
    $i = 0;
    $fp = fopen("chat.log","a");
    while ($i < $Schatsize-15);
        $restuff = "\n" + $chatfile[$i];
fputs("$fp","$restuff");
        $i++;
    endwhile;

```

```

fclose ($fp);

Commented Out! **/

endif;
>

</noframes></frameset></html>

```

### *Anexo12. Listado del script input.phtml*

```

<?
/** Read from channels.txt **/
$channelinput[]=file("channels.txt");
$numchannels[]=count($channelinput);
$i=0;
while ($i<$numchannels);
    $chan[$i] = strtok($channelinput[$i],"=");
    $inputfiles[$i] = strtok ("=");
    $i++;
endwhile;

/** Set up defaults **/
$drefresh = 15;
$dchannel = $chan [0];
if (!$senter);
    $senter=0;
endif;
if (!$sleaves);
    $sleaves=0;
endif;

/** Check to see if defaults need to be used **/
if (!$srefresh);
    $srefresh = $drefresh;
endif;
if (!$schannel);
    $schannel= $dchannel;
endif;

/** Replace spaces in channel name to pass with URL **/
$chancode=reg_replace(" ","+",$schannel);

>

<?
/** Channels **/
$i=0;
while ($i<$numchannels);

```

```

        if ($channel==$chan[$i]);
        $inputfile = $inputfiles[$i];
        endif;
        if ($oldchannel==$chan[$i]);
        $oldinputfile = $inputfiles[$i];
endif;
    $i++;
endwhile;
/** Check to see if they're changing the channel **/
if (!$oldchannel);
else;
if ($oldchannel!=$channel);
    $enters=1;
    $leaves=1;
endif;
endif;

/** Fix the Nick **/
if ($nick);
    $nick = reg_replace ("^[a-zA-Z0-9. ]","",$nick);
    $nickcode = reg_replace(" ","+",$nick);
endif;

/** If there's a comment, post it **/
if (!$chat);
else;

    $nick = reg_replace("{<>}", "", $nick);
    $chat = reg_replace("{<>}", "", $chat);
    if (strlen($chat)>1000);
    $chat="<i>Est&aacute; interesante...</i>";
    endif;
    $chatemp = "";
    $chatlen = strlen ($chat);
    $stok = strtok($chat, " ");
    while($chatlen>0);
        $stoken = reg_replace("^([Hh][Tt][Tt][Pp]:.*\\^[Mm][Aa][Ll]
)]([Ll][Tt][Oo]:.*\\^[Ff][Tt][Pp]:.*","<a href=$stok target=_blank>$stok</a>",$stok);
        $chatemp = $chatemp + " " + $stoken;
        $chatlen = $chatlen - 1 - strlen ($stok);
        $stok = strtok(" ");
    endwhile;
    $chat = $chatemp;

    $string = "\n<b><font color=#000000>" + $nick + "</font><b><i> ";
    $string = $string + date ("D h:i") + " :</i> ";

    if ($bold);
    $string = $string + "<b>";
    endif;
    if ($italics);
    $string = $string + "<i>";
    endif;

```

```

if ($big);
    Sstring = Sstring + "<font size=+!>";
endif;

Sstring = Sstring + $chat;
if ($big);
    Sstring = Sstring + "</font>";
endif;
if ($italics);
    Sstring = Sstring + "</i>";
endif;
if ($bold);
    Sstring = Sstring + "</b>";
endif;
if (!$emoticon);
    Semoticon = "none";
endif;

if (Semoticon=="happy");
    Sstring = Sstring + "&#160; <b>:-)</b>";
endif;

if (Semoticon=="wink");
    Sstring = Sstring + "&#160; <b>;-)</b>";
endif;

if (Semoticon=="sad");
    Sstring = Sstring + "&#160; <b>-(</b>";
endif;

$fp = fopen("$inputfile", "a");
fputs("$fp", "Sstring");
fclose ($fp);
endif;

/** The body loads up the refresh window with values passed **/
>
<center>
<?

/** Check to see if this is first login **/
if ($login==1);
    if (!$email);
        $nick = "";
    else;
        $fp = fopen("users.log", "a");
        $stuff = "\n" + date ("D h:i") + ": " + $nick + " at " + $email;
        $stuff = $stuff + " from " + $REMOTE_HOST;
        fputs("$fp", "$stuff");
        fclose ($fp);
        $enters = 1;
        $leaves = 0;
    endif;
endif;

```

```

endif;
endif;

if ($enters==1);
    $fp = fopen("$inputfile","a");
    $shellthere = "\n<b><font color=#000000>" + $snick + "</font></b><i> ";
    $shellthere = $shellthere + date ("D h:i") + ": * Entro *</i>";
    fputs("$fp","$shellthere");
    fclose ($fp);

    endif;

if ($leaves==1);
    $fp = fopen("$dinputfile","a");
    $shellthere = "\n<b><font color=#000000>" + $snick + "</font></b><i> ";
    $shellthere = $shellthere + date ("D h:i") + ": * Salio del ";
    $shellthere = $shellthere + $channel + " *</i>";
    fputs("$fp","$shellthere");
    fclose ($fp);

    endif;

if (!$snick);
    echo "<body background=/logos/fondo.jpg text=#000000 link=#CCCC00 vlink=#CCCCCC>";
    echo "<form action=input.phtml method=get>";
    echo "<input type=hidden name=login value=1>";
    echo "Escribe un seud&oacute;nimo: ";
    echo "<input name=nick><br>";
    echo "Escribe tu direcci&oacute;n de correo: ";
    echo "<input name=email><br>";
    echo "<input type=submit value=Entrar!>";
    echo "</form>";
else;
    /** Ok, they're in. Show chat interface. **/
    >
    <body background="/logos/fondo.jpg" vlink="#0000ff" onLoad="setfocus()"
    <?
    /** First form: Change Channel and refresh rate **/
    echo "<table cellpadding=12 border=0 cellspacing=2><tr><td valign=top>";
    echo "<center>";
    echo "<a href=index.html target=_parent>";
    /** echo "<img src=exitchat.gif border=0 alt=Salir>";
    echo "</a><br><img src=pchat2.gif>"; **/
    echo "<BR><BR><h2>Salir</h2></a><BR><BR>";
    echo "</center></td><td valign=top>";

    echo "<form action=input.phtml method=get>";
    echo "<input type=hidden name=oldchannel>";
    echo " value=\"\$channel\">";
    echo "<input type=hidden name=nick value= ";
>"<? echo $snick;>"<?
    echo ">";
    echo "<nobr>";
    echo "<b><font color=#000000>";

```

```

echo $nick;
echo "</font></b>";
echo "&#160;&#160;";

/** This shows the channel selections
and selects the correct channel **/

echo "<select name=channel>";
Si=0;
while (Si<$numchannels);
echo "<option>";
if ($channel==$chan[$i]);
echo " selected";
endif;
echo ">";
echo $chan[$i];
Si++;
endwhile;
echo "</select>";
echo " <input type=submit value=Cambiar> ";

/** Refresh Rates **/
echo "<font size=-1 color=#000000>Refrescar cada (seg):</font>";
echo "<input type=radio name=refresh value=15 checked> 15 ";
echo "<input type=radio name=refresh value=30> 30 ";
echo "<input type=radio name=refresh value=60> 60 ";
echo "</nobr></form>";

/** Second form: Chat! **/
echo "<form action=input.phtml name=input method=get>";
echo "<input type=text name=chat size=50>";
echo "<input type=hidden name=channel value=\"\$channel!\">";
echo "<input type=hidden name=refresh value=$refresh>";
echo " <input type=submit value=Entrar!><br>";

echo "<b>Opciones:</b> <input type=checkbox name=bold>Negrilla ";
echo " <input type=checkbox name=italics>Italias ";
echo " <input type=checkbox name=big>Grande &#160;&#160;&#160;&#160;";

echo " <input type=radio name=emoticon value=happy>Feliz ";
echo " <input type=radio name=emoticon value=wink>Gui&ntilde;ar ";
echo " <input type=radio name=emoticon value=sad>Triste ";

echo "<input type=hidden name=nick value= ";
>"<? echo $nick;>"<?
echo ">";
echo "</form>";
endif;
echo "</td></tr></table></center>";
>
<script language=javascript>
!!--
function setfocus() {
parent.refresh.location.href='refresh.phtml?channel=<? echo $channelcode;>&

```



```
refresh=<? echo $refresh;>';
    document.input.chat.focus();
    return true;}
// -->
</script>
```

### Anexo 13. Listado del archivo refresh.phtml

```
<?
/** User-defined channels */
/** Read from channels.txt */
    $channelinput[]=file("channels.txt");
    $numchannels[]=count($channelinput);
    $i=0;
    while ($i<$numchannels);
        $chan[$i] = strtok($channelinput[$i],"=");
        $inputfiles[$i] = strtok ("=");
        $i++;
    endwhile;

/** Set up defaults */
    $drefresh = 30;
    $dchannel = $chan[0];

/** Check to see if defaults need to be used */
    if (!$refresh);
        $refresh = $drefresh;
    endif;
if (!$channel);
    $channel= $dchannel;
endif;

/** Channels */
    $i=0;
    while ($i<$numchannels);
        if ($channel==$chan[$i]);
            $inputfile = $inputfiles [$i];
        endif;
        $i++;
    endwhile;

/** Replace spaces in channel name to pass with URL */
    $chancode=reg_replace(" ","+",$channel");
>
<META HTTP-EQUIV=REFRESH CONTENT="<? echo $refresh;>; URL=refresh.phtml<?
    echo "?refresh=";
    echo $refresh;
    echo "&channel=";
```

```
    echo $chancode;>">
<body background="/logos/fondo.jpg">
<h3><tt><font color="#000000"><? echo $chanel;></font></tt></h3>

<?

/** Read in the chatfile **/
    $chatfile[]=file("$inputfile");

/** Show the last 15 lines in reverse order **/
    $chatsize = count ($chatfile);
    echo $chatfile[$chatsize];
    $stop = $chatsize - 15;
    while($chatsize > $stop);
        $chatsize = $chatsize -1;
        echo $chatfile[$chatsize];
        echo "<br>";
    endwhile;

/** Trim the file if it gets too big **/

/** You could also save it off as a log... **/

    $fz = fileSize("$inputfile");
    if ($fz > 20000);

        $fp = fopen("$inputfile","w");
        $stop2 = $stop + 15;
        while($stop != $stop2);
            $restuff = "\n" + $chatfile[$stop];
            fputs("$fp","$restuff");
            $stop++;
        endwhile;
        fclose ($fp);
        /** Make a log file **/
        /** Commented Out!

        $chatsize = count ($chatfile);
        $i = 0;
        $fp = fopen("chat.log","a");
        while ($i < $chatsize-15);
            $restuff = "\n" + $chatfile[$i];
            fputs("$fp","$restuff");
            $i++;
        endwhile;
        fclose ($fp);
        Commented Out! **/
    endif;>
</body>
</html>
```

## Anexo 14. Listado del archivo *index.phtml*. Sección Resúmenes Informativos

```

<?
  $titulo = "Intranet-Res&uacute;menes informativos";
  include "cabeza.phtml";
>

<? if($dia<=0 || $mes<=0 || $anio<=0);>
<BR>

<CENTER>
<H2>Res&uacute;menes informativos</H2>
</CENTER>

<HR SIZE=8>
<BR>
<BR>

En atención a los lineamientos definidos por el Gobierno Federal en el sentido de que la relación con la
prensa debe ser profesional, transparente, constructiva y medida a partir del reconocimiento de las
mutuas necesidades y responsabilidades, atiende, en estrecha coordinación con las Direcciones Generales
y/o Unidades, las solicitudes de información formuladas por los medios acreditados, centros de
investigación, agrupaciones sociales y público en general, fomentando flujos informativos que hagan del
conocimiento público los esfuerzos que desde el Instituto se realizan en pro del diseño de la política
ambiental y en la aplicación de sus instrumentos.
<BR><BR>

Así mismo, tiene la función de diseñar las políticas y herramientas de difusión para la integración de una
imagen institucional que dé plena cuenta de las actividades que en cada una de las áreas del INE se
desarrollan.
<BR><BR>
Para consultar los res&uacute;menes informativos solo elija la fecha y oprima el bot&ocaron "Buscar".
<BR><BR><BR><BR>
<?
$dia1=date("d");
$mes1=date("m");
$anio1=date("y");
$anio2=date("Y");
>
<form action="index.phtml" method=POST>

<center>

<table border=0>

<tr>
<td align="right">Dia:</td>
<td><Select Name="dia">
<option>1
<option>2
<option>3
<option>4

```

```
<option>5
<option>6
<option>7
<option>8
<option>9
<option>10
<option>11
<option>12
<option>13
<option>14
<option>15
<option>16
<option>17
<option>18
<option>19
<option>20
<option>21
<option>22
<option>23
<option>24
<option>25
<option>26
<option>27
<option>29
<option>30
<option>31
<option Selected><? echo "Sdia!";>
</select>
</td>
<td align="right">Mes:</td>
<td>
<Select Name="mes">
<option>01
<option>02
<option>03
<option>04
<option>05
<option>06
<option>07
<option>08
<option>09
<option>10
<option>11
<option>12
<option Selected><? echo "$mes!";>
</select>
</td>

<td align="right">A&ntilde;o:</td>
<td>
<Select Name="anio">
<option>1998
<option>1997
<option selected><? echo "$anio2";>
```

```
</select>
</td>
</tr>

<tr><td></td><TD></td></tr>
<tr><td></td><TD></td></tr>
<tr><td></td><TD></td></tr>

<tr>
<TD></td>
<TD></td>
<TD></td>
<TD></td>
<TD></td>
<td align="right"><input type="submit" value=" Buscar "></td>
</tr>
</table>
</form>

<?else?>

<?
switch($dia) {
    case "1";
        $dia="01";
        break;
    case "2";
        $dia="02";
        break;
    case "3";
        $dia="03";
        break;
    case "4";
        $dia="04";
        break;
    case "5";
        $dia="05";
        break;
    case "6";
        $dia="06";
        break;
    case "7";
        $dia="07";
        break;
    case "8";
        $dia="08";
        break;
    case "9";
        $dia="09";
        break;
    case "10";
        $dia="10";
        break;
    case "11";
```

```
$dia="11";  
break;  
case "12";  
$dia="12";  
break;  
case "13";  
$dia="13";  
break;  
case "14";  
$dia="14";  
break;  
case "15";  
$dia="15";  
break;  
case "16";  
$dia="16";  
break;  
case "17";  
$dia="17";  
break;  
case "18";  
$dia="18";  
break;  
case "19";  
$dia="19";  
break;  
case "20";  
$dia="20";  
break;  
case "21";  
$dia="21";  
break;  
case "22";  
$dia="22";  
break;  
case "23";  
$dia="23";  
break;  
case "24";  
$dia="24";  
break;  
case "25";  
$dia="25";  
break;  
case "26";  
$dia="26";  
break;  
case "27";  
$dia="27";  
break;  
case "28";  
$dia="28";  
break;  
case "29";
```

```
        Sdia="29";
        break;
        case "30";
        Sdia="30";
        break;
        case "31";
        Sdia="31";
        break;
        default;
        Sdia="0";
        break;
    }

    switch($anio) {
        case "1997";
        $anio="97";
        break;
        case "1998";
        $anio="98";
        break;
        default;
        $anio="0";
        break;
    }

    $todo="$anio"+"$mes"+"$dia+".html";

    $fd=@fopen ($todo, "r");
    if ($fd < 0);
        echo "<BR><BR><BR><BR>";
        echo "<h2>Lo sentimos, el res&uacute;men informativo no est&aacute; disponible.
    </h2>";
    else;
        include "$todo";
    endif;
?>
<?endif>

<BR><BR><BR>
<hr size=5>
</center>
<BR>
<? include "pie.phtml";?>
```

## Anexo 15. Listado del archivo index.phtml. Sección Aplicaciones

```

<?
  Titulo = "Intranet-Aplicaciones";
  include "cabeza.phtml";
?>

<CENTER>
<H2>Aplicaciones de uso com&uacute;n </H2>
</CENTER>

<HR SIZE=8>
<BR>
<BR><BR>
Para copiar a disco local alg&uacute;n programa solo haga "click" en el logotipo
.
<blockquote>
<table border=2>
<tr>
<td>
<A HREF="/intranet/aplicaciones/icq98a.exe">
<IMG SRC="icqdownload.gif" BORDER=0 ></A>
</td>
<td>
<td>
<B>ICQ</B> la revolucionaria herramienta para saber quien se encuentra en l&iacu
te;nea
en cualquier momento y poder entablar una comunicaci&oacute;n, intercambio de so
ftware, URL, etc. Est&aacute; versi&oacute;n es para Windows 95.
</td>
</tr>
<tr>
<td>
<A HREF="/intranet/aplicaciones/icq311.exe">
<IMG SRC="icqdownload.gif" BORDER=0 ></A>
</td>
<td>
<B>ICQ</B> para Windows 3.11
</td>
</tr>
<tr>
<td>
<A HREF="/intranet/aplicaciones/v95i314e.zip">
<IMG SRC="mcafee.gif" BORDER=0 ></A>
</td>
<td>
Antivirus de McAfee para Windows 95
</td>
</tr>
<tr>
<td>
<A HREF="/intranet/aplicaciones/dat-3108.zip">
<IMG SRC="mcafee.gif" BORDER=0 ></A>

```



```

</td>
<td>
Actualizaci&oacute;n de la base de datos del antivirus de McAfee
</td>
</tr>
<tr>
<td>
<A HREF="/intranet/aplicaciones/winzip95.exe">
<center><IMG SRC="zip.gif" BORDER=0 HEIGHT=50 WIDTH=60></A></center>
</td>
<td>
Winzip. Herramienta para compactar archivos.
</td>
</tr>

<tr>
<td>
<A HREF="/intranet/aplicaciones/cc32e406.exe">
<center><IMG SRC="netscape.gif" BORDER=0 ></A></center>
</td>
<td>
Netscape Communicator v4.06
</td>
</tr>

<tr>
<td>
<A HREF="/intranet/aplicaciones/ws_ftp.zip">
<center><IMG SRC="wsftp.gif" BORDER=0 HEIGHT=50 WIDTH=100></A></center>
</td>
<td>
WsFTP95
</td>
</tr>

</table>
</blockquote>
<BR>
<BR>

<P><CENTER><A HREF="/intranet/index.phtml"><IMG SRC="/logos/regresa.gif"
ALT="INE" BORDER=0 ALIGN=bottom></A></CENTER></P>
<HR size=5>
<BR>

<? include "pie.phtml"; ?>

```

**Anexo 16. Listado del archivo index.phtml. Sección Informática**

```

<?
  Stitulo = "Intranet-Infom&aacute;tica";
  include "cabeza.phtml";
?>

<CENTER>
<H2>Infom&aacute;tica</H2>
</CENTER>

<HR SIZE=8>
<BR>
<BR><BR>
<blockquote>
<A HREF="/intranet/metodologia/introducci.phtml">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Introducción</B>
<BR><BR>
<A HREF="/intranet/metodologia/discipli.phtml">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Disciplina de Desarrollo de Sistemas</B>
<BR><BR><BR>
<b>Metodolog&iacute;a de Desarrollo de Sistemas</b>
</blockquote>

<A HREF="/intranet/metodologia/vision.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Documento de visi&oacute;n</B>
<BR><BR>

<A HREF="/intranet/metodologia/especfun.phtml">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Especificaci&oacute;n Funcional</B>

</blockquote>
<BR>
<A HREF="/intranet/metodologia/convenci.phtml">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Convenciones para la incorporaci&oacute;n de informaci&oacute;n a la p&aacute;gina del Instituto Nacional de Ecolog&iacute;a en la WWW (101197)
</B>
<BR>

<BR><BR>
<b>Procedimientos</b>
</blockquote>

<A HREF="/intranet/metodologia/guia.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Gu&iacute;a para la elaboraci&oacute;n</B>
<BR><BR>

```

```

<A HREF="/intranet/metodologia/maniip.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Integraci&oacute;n de informaci&oacute;n a la p&aacute;gina del INE</B>
<BR><BR>

<A HREF="/intranet/metodologia/manrecip.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Recuperaci&oacute;n de informaci&oacute;n de la p&aacute;gina del INE</B>
<BR><BR>

<A HREF="/intranet/metodologia/manrip.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Respaldo de informaci&oacute;n de la p&aacute;gina del INE</B>
<BR><BR>
</blockquote>
<BR>
<BR>

<P><CENTER><A HREF="/intranet/index.phtml"><IMG SRC="/logos/regresa.gif"
ALT="INE" BORDER=0 ALIGN=bottom></A></CENTER></P>
<HR size=5>
<BR>

<? include "pie.phtml"; ?>

```

### *Anexo 17. Listado del archivo index.phtml. Sección Capacitación*

```

<?
  $titulo = "Intranet-Capacitaci&oacute;n";
  include "cabeza.phtml";
?>

<CENTER>
<H2>Capacitaci&oacute;n</H2>
</CENTER>

<HR SIZE=8>
<BR>
<BR><BR>
<blockquote>
<A HREF="/intranet/capacitacion/programa.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Programa de capacitaci&oacute;n</B>
<BR><BR>
<A HREF="/intranet/capacitacion/manual.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>
<B>Manual de la sala de operaci&oacute;n</B>
<BR><br>
<A HREF="/intranet/capacitacion/cursos.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30></A>

```

```

<B>Cursos que se imparten</B>
<BR><BR>
<A HREF="/intranet/capacitacion/calendario.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30</A>
<B>Calendario de cursos</B>
<br><BR>
<A HREF="/intranet/capacitacion/ficha.html">
<IMG SRC="/logos/balat.gif" BORDER=0 HEIGHT=21 WIDTH=30</A>
<B>Ficha de inscripci&oacute;n</B>
<BR><BR>

</blockquote>
<BR><BR>

<P><CENTER><A HREF="/intranet/index.phtml"><IMG SRC="/logos/regresa.gif"
ALT="INE" BORDER=0 ALIGN=bottom</A></CENTER></P>
<HR size=5>
<BR>
<? Include "pie.phtml"; ?>

```

### *Anexo 18. Listado del archivo index.phtml. Sección Clarín Informático*

```

<?
  Stitulo = "Intranet-Clar&iacute;n inform&aacute;tico";
  include "cabeza.phtml";
>

<center></center>
<p>
<hr size=5>
<br><BR>
<center><table cellpadding=20 cellspacing=0 border=0>
<tr>
<td valign="top"><a href="directorio.html"><IMG SRC="/logos/balat.gif" WIDTH=30
HEIGHT=21 ALIGN="LEFT" BORDER=0></a>
<font size=2 face="Comic Sans MS"><b>Acerca del Clarín inform&aacute;tico.</b></font>
</td>
<td rowspan=2><center></center>
</td>
<td valign="top" align="left"><a href="detalles.html"><IMG SRC="/logos/balat.gif
" WIDTH=30 HEIGHT=21 ALIGN="LEFT" BORDER=0></a>
<font size=2 face="Comic Sans MS"><b>¿Deseas colaborar con este boletin?
Ve los detalles...</b></font>
</td>
</tr>
<tr>
<td valign="top"><a href="encartes.html"><IMG SRC="/logos/balat.gif" WIDTH=30 H

```

```
EIGHT=21 ALIGN="LEFT" BORDER=0></a>
<font size=2 face="Comic Sans MS"><b>ENCARTES de difusión INFORMÁTICA.</b></font>
</td>
<td valign="top">
<a href="num_ant.html"><IMG SRC="/logos/balat.gif" WIDTH=30 HEIGHT=21 ALIGN="LEFT"
BORDER=0></a>
<font size=2 face="Comic Sans MS"><b>Clarines InformáticoS.</b></font></td>
</tr>
</table></center>
<BR>
<br><BR>
<center>
<table border=0 width="100%">
<tr>
<td align=left><a href="http://ddi.semarnap.gob.mx"></a>
</td>
<td><CENTER><A HREF="/index.html"><IMG SRC="/logos/regresa.gif" ALT="¡NE" BORDER=0
ALIGN=bottom></A></CENTER>
</td>
<td align=right><a href="mailto:difusion@ine.gob.mx"></a>
</td>
</tr>
</table>
</center>
<p>
<br>
<hr size=5>
<br>
<? include "pie.phtml"; ?>
```

## **GLOSARIO**

### **Ancho de banda**

Cantidad de bits que pueden viajar por el medio físico (cable coaxial, par trenzado, fibra óptica, etc.). Entre mayor sea el ancho de banda obtenemos más rápido la información. Se mide en millones de bits por segundo (Mbps). Las velocidades típicas hoy en día son de 10 Mbps a 100 Mbps.

### **Aplicación**

*Software* que realiza una función útil. Los programas que se utilizan para realizar alguna función (como correo electrónico, FTP, etc.) son las aplicaciones cliente.

### **API (Application Program Interface)**

Conjunto de reglas de programación que determinan como una aplicación debe acceder a un servicio.

### **ARCNET**

Red de computadoras con recursos asignados (*Attached Resource Computer Network*). Red de área local (LAN) desarrollada por Datapoint Corporation que utilizaba una tecnología de acceso llamada *Token Passing* y que tiene un ancho de banda de 2.5 Mbps. El diseño original de las redes ARCNET se apoyaba en la topología de estrella.

### **Archie**

Un sistema para la localización de archivos que están disponibles públicamente por FTP anónimo. Es necesario conocer el nombre del archivo o una subcadena del mismo para utilizar archie.

### **Archivo binario**

Archivo que contiene códigos y caracteres que sólo pueden ser utilizados por un tipo específico de *software*. Los más comunes son los archivos ejecutables, gráficos y documentos con formato.

### **Archivos compactados**

Los archivos compactados son la compresión de los datos al eliminar datos redundantes, de esta manera permiten: un mayor almacenamiento de archivos, aumentar la velocidad de transferencia de los mismos, etc. Entre los formatos de archivos compactados se encuentran:

- Para PC  
arc, arj, lha, zip. Utilería: Winzip: <http://www.winzip.com>  
zip. Utilería PKZIP, PKUNZIP
- Para Macintosh  
hqx, bin, SIT Utilerías: Stuffit
- Para UNIX

gzip Utileria Gzip

Z Utileria compress/uncompress

### **Archivos de dominio público**

Son los archivos que se pueden obtener de Internet y que han sido puestos a disposición de los usuarios por compañías, dependencias y personas. Pueden ser freeware o shareware.

### **Archivo de texto**

Archivo que utiliza solamente caracteres del estándar ASCII y por lo tanto que puede ser enviado por correo electrónico sin ningún tipo de modificación.

### **ARPANET. (Advanced Research Projects Agency Network)**

Red experimental con fines militares establecida en los setenta, en la cual se probaron las teorías y el *software* en los que está basado Internet. ARPANET era una red experimental que apoyaba la investigación militar, en particular la investigación sobre como construir redes que pudieran soportar fallas parciales (como las producidas por los bombardeos) y aún así funcionar. La red fue diseñada para requerir un mínimo de información de las computadoras que forman parte de ella. La filosofía era que cada computadora en la red se pudiese comunicar, como un elemento particular con cualquier computadora.

### **ASCII (American Standard Code for Information Interchange)**

Es el estándar para el código utilizado por computadoras para representar todas las letras (mayúsculas, minúsculas, letras latinas, números, signos de puntuación, etc.). El código estándar ASCII es de 128 letras representadas por un dígito binario de 7 posiciones (7 bits), de 0000000 a 1111111.

### **Backbone**

Línea de transmisión de información de alta velocidad o una serie de conexiones que juntas forman una vía con gran ancho de banda. Un backbone conecta dos puntos o redes distanciadas geográficamente, a altas velocidades.

### **Bases de datos distribuidas**

Bases de datos que se pueden encontrar en diversas partes del planeta y que se presentan ante el usuario como una base de datos única. Un ejemplo de ello es el DNS (*Domain Name Service*) en que se basa Internet, donde las direcciones de las computadoras se encuentran en diversas computadoras (cada una encargada de un dominio), y que se presentan ante el usuario como una base de datos única con todos los dominios del planeta.

### **baud**

Sólo a baja velocidad, los baudios son iguales a los bits por segundo (bps); por ejemplo, 300 baudios es igual a 300 bps. Sin embargo, se puede hacer que un baudio represente más de un bit por segundo. Por ejemplo un modem V.22 bits genera 1,200 bps a 600 baudios.

**bit (Binary digit)**

Unidad mínima de almacenamiento de información. Su valor puede ser 0 ó 1 ó verdadero o falso.

**BITNET. (Because It's Time NETwork ó Because It's There NETwork)**

Red de sitios educativos (investigación y universitarios) separada de Internet, pero el correo electrónico es libremente intercambiado entre BITNET e Internet. Los *Listservs* son la forma más popular de los grupos de noticias originados en BITNET. Las computadoras de BITNET son usualmente mainframes corriendo el sistema operativo VMS (variante de UNIX).

**bps**

Bits por segundo (*Bits-Per-Second*). Es la velocidad a la que se transmiten los bits en un medio de comunicación.

**Bps**

Bauds por segundo. Número de cambios que sufre la señal por segundo y es indicativo de la cantidad de bits por segundo que se están transmitiendo. Uno puede aumentar la velocidad de enlace si utiliza compresión de datos. Para aprovechar la máxima velocidad de un módem, tanto el proveedor como el usuario deben tener módems que operen a la máxima velocidad y utilizar ambos la compresión de datos.

**Byte**

Conjunto de 8 bits. Suele representar un valor asignado a un carácter.

**CERN**

Laboratorio Europeo de Física de Partículas. Fue el desarrollador inicial del World Wide Web. Actualmente los estándares del Web son desarrollados por la *World Wide Web Organization* (3W0). El Website del CERN se encuentra en <http://www.cern.ch>

**Chat**

Término utilizado para describir la comunicación de usuarios en tiempo real.

**CGI (Common Gateway Interface)**

Una interfaz escrita en un lenguaje de programación (perl, c, c++, visual basic, PHP, etc) y posteriormente ejecutada o interpretada por una computadora servidor para contestar a peticiones del usuario desde una computadora con una aplicación cliente; casi siempre desde el World Wide Web. Esta interfaz permite obtener los resultados pedidos, como los que resultan al consultar una base de datos.

**Cliente**

- a) Una aplicación que permite a un usuario obtener un servicio de un servidor localizado en la red.
- b) Un sistema o proceso que solicita a otro sistema o proceso que le preste un servicio.



### **Correo electrónico (e-mail)**

Permite el intercambio de mensajes entre personas conectadas a una red de manera similar al correo tradicional. Entre las aplicaciones cliente de correo electrónico tenemos a *Eudora*, *Mail*, *Pine*, *Pegasus*, etc. La definición acerca del correo electrónico fue especificada en el RFC # 822.

### **CSMA/CD (Carrier Sense Multiple Access/Collision Detection)**

Sensor de portadora de accesos múltiples/detección de colisiones. Un método de acceso en las comunicaciones de banda base que emplea una técnica de detección de colisiones. Cuando un dispositivo trata de ganar acceso a la red, verifica si la misma está libre. Si no lo está, espera una cantidad aleatoria de tiempo antes de intentarlo nuevamente. Si la red está libre y dos dispositivos tratan de ganar acceso exactamente al mismo tiempo, ambos se retractan para evitar una colisión y luego cada uno de ellos espera una cantidad aleatoria de tiempo antes de reintentarlo.

### **Cyberspacio**

Término originado por William Gibson en su novela *Neuromancer*. La palabra Cyberspacio es ampliamente usada para descubrir los recursos de información disponibles a través de Internet.

### **Dirección electrónica**

Dirección de un usuario en Internet. Por medio de ella es posible enviar correo electrónico a un usuario. Esta es única para cada usuario y se compone por el login de un usuario, arroba y el nombre del servidor de correo electrónico. Por ejemplo: paco@ine.gob.mx

### **Dirección IP**

La dirección del protocolo de Internet (IP) es la dirección numérica de una computadora en Internet. Cada dirección electrónica se asigna a una computadora conectada a Internet y por lo tanto es única. La dirección IP está compuesta de cuatro octetos como 200.33.112.3

### **DNS (Domain Name Server)**

Servidor de nombres de dominio. Es un sistema que se establece en un servidor (que se encarga de un dominio) que traduce nombres de computadoras (como siankan.ine.gob.mx) a domicilios numéricos de Internet (direcciones IP) (como 200.33.112.15).

### **Dominio**

Conjunto de computadoras que comparten una característica común, como el estar en el mismo país, en la misma organización o en el mismo departamento. Cada dominio es administrado por un servidor de dominios. Los dominios se establecen de acuerdo al uso que se le da a la computadora y al lugar donde se encuentre. Ejemplo:

- .com Comercial
- .edu educación (USA)
- .gob gobierno (USA)
- .mx México
- .es España, etc,

Los dominios a su vez se van dividiendo en otros dominios:

.gob.mx Gobierno de México  
.com.mx Comercio en México

### **Ethernet**

Tipo de red de área local desarrollada en forma conjunta por Xerox, Intel y Digital Equipment. Se apoya en la topología de bus y tiene un ancho de banda de 10 Mbps.

### **Fibra óptica**

Combinación de vidrio y materiales plásticos. A diferencia del cable coaxial y del par trenzado no se apoya en los impulsos eléctricos, sino por medio de impulsos luminosos. Es el medio físico por medio del cual se pueden conectar varias computadoras.

### **Finger**

Programa que permite determinar si un usuario específico está en un momento determinado en línea, o bien, muestra información del usuario, nombre, dirección, teléfono y si está activo o no.

### **Fire Wall**

Una combinación de *hardware* y *software* que separa una red de área local (LAN) en dos o más partes con propósitos de seguridad.

### **Freeware**

Aplicaciones que pueden obtenerse directamente de Internet y que no es necesario pagar por su utilización.

### **FTP**

Protocolo de transferencia de archivos (File Transfer Protocol).

### **GIF (Graphics Interchange Format)**

Formato binario de archivos que contienen imágenes.

### **GIF Animado**

Formato binario que permite almacenar varios archivos con formato GIF de manera que un visualizador pueda desplegar cada una de las imágenes en orden.

### **GNU**

Fundación para el *software* libre (*FSF - Free Software Foundation*). Busca eliminar las restricciones de uso, copia, modificación y distribución del *software*. Actualmente se encuentra apoyando el desarrollo de sistemas operativos (Linux), compiladores (compilador GNU C Compiler (gcc), Perl), etc. Trata de promover, desarrollar y usar del *software* libre en todas las áreas de la computación. Específicamente, la Fundación pone a disposición de todo el mundo un completo e integrado sistema de *software* llamado GNU. La mayor parte de este sistema está ya siendo utilizado y distribuido. El costo del *software* únicamente está determinado por el costo del material utilizado para distribuirlo.

### **Gopher**

Un servicio basado en menús que sirve para explorar recursos de Internet.

### **Grupos de noticias**

Área de mensajes automatizada, operada normalmente a través de USENET, en la que los suscriptores dejan mensajes a todo el grupo sobre temas específicos.

### **Guest (huésped)**

Palabra clave utilizada comúnmente para obtener archivos públicos de una computadora llamada host (anfitrión), que es el servidor donde se encuentran los archivos.

### **Hipermedia**

Combinación de texto y multimedia. Actualmente es un recurso ampliamente explotado en el World Wide Web.

### **Hipertexto**

Documentos que contienen vínculos con otros documentos, al seleccionar un vínculo automáticamente se despliega el segundo documento.

### **Homepage (Página inicial)**

Es la página Web de entrada a un lugar del World Wide Web. Es considerada la página principal.

### **Host (Anfitrión)**

Computadora a la que tenemos acceso de diversas formas (telnet, FTP, World Wide Web, etc.). Es el servidor que nos provee de la información que requerimos para realizar algún procedimiento desde una aplicación cliente.

### **HTML (Hiper Text Markup Lenguaje)**

Lenguaje de marcado de hipertexto. Es el lenguaje con el cual se escriben los documentos en el World Wide Web.

### **HTTP (Hiper Text Transfer Protocol)**

Protocolo de transferencia de hipertextos. Es el protocolo usado por el World Wide Web para transmitir páginas HTML.

### **INTERNET**

Es la red de redes. Es una red de cómputo a nivel mundial que agrupa a distintos tipos de redes usando un mismo protocolo de comunicación. Los usuarios en Internet pueden compartir datos, recursos y servicios. Internet se apoya en el conjunto de protocolos TCP/IP. De forma más específica, Internet es la WAN más grande que hay en el planeta, e incluye decenas de MAN's y miles de LAN's. Las computadoras que lo integran van desde modestos equipos personales, minicomputadoras, estaciones de trabajo, mainframes hasta supercomputadoras. Internet no tiene una autoridad central, es descentralizada. Cada red mantiene su independencia y se une cooperativamente al resto respetando una serie de normas de interconexión. El organismo que se

encarga de regular, establecer estándares, administrar y hacer operacional a Internet es la ISOC (*Internet Society*).

### **InterNIC**

Es el nombre que se le da al conjunto de proveedores de servicios de registro. El InterNIC define los nombres de dominio a nivel mundial. El sitio de la Internic (<http://www.internic.net>) es mantenido además por la *National Science Fundation* (NSF <http://www.nsf.gov>) y la compañía de telecomunicaciones ATT (<http://www.att.com>).

### **Intranet**

Una red privada dentro de una compañía u organización que utiliza el mismo *software* que se encuentra en Internet, pero que es solo para uso interno. Por ejemplo, muchas compañías tienen servidores World Wide Web disponibles solo para sus empleados.

### **IP (Internet Protocol)**

Protocolo de internet. Permite a un paquete de datos viajar a través de múltiples redes hasta alcanzar su destino. Se encarga de la capa de red del modelo OSI.

### **ISDN (Integrated Services Digital Network)**

Red Digital de Servicios Integrados (RDSI). En el servicio de ISDN las líneas telefónicas transportan señales digitales en lugar de señales analógicas, lo que aumenta considerablemente la velocidad de transferencia de datos a la computadora. Si se cuenta con el equipo y el *software* necesarios, y si la central telefónica local ofrece ISDN y el proveedor de servicios lo soporta, el ISDN es posible utilizarlo. La velocidad de transferencia que puede alcanzar ISDN es de 128,000 bps, aunque en la práctica las velocidades comunes son de 56,000 ó 64,000.

### **ISO (Internacional Standards Organization)**

Organización Internacional de Estandarización. Es una organización que ha definido un conjunto de protocolos diferentes, llamados protocolos ISO/OSI. Esta organización de carácter voluntario fue fundada en 1946 y es responsable de la creación de estándares internacionales en muchas áreas, incluyendo la informática, la ecología y las comunicaciones. Está formada por las organizaciones de normalización de sus 89 países miembros.

### **Java**

Un lenguaje de programación que permite ejecutar programas escritos en un lenguaje muy parecido al C++, llamados applets, a través del World Wide Web. La diferencia con un CGI es que la ejecución se realiza totalmente en la computadora cliente, en lugar del servidor.

### **JPG o JPEG.(Joint Photographic Experts Group)**

Un formato para guardar imágenes que las hace ocupar poco espacio en la memoria de la computadora y en disco. Por esta razón son más rápidas de transmitir a través del web. A diferencia del formato GIF, este formato no es aceptado por todos los visualizadores del World Wide Web.

**Kilobyte**

Mil bytes. Actualmente es usado como 1024 (dos elevado a la 10) bytes.

**LAN (Local Area Network)**

Red de área local. Red cuyas dimensiones no exceden 10 km. Puede tratarse de computadoras conectadas en una oficina, en un edificio o en varios.

**Listas de correo, listas de discusión o foros de discusión**

Servicio automatizado de mensajes, a menudo moderado por un propietario en el que los suscriptores reciben mensajes dejados por otros suscriptores de un tema dado.

**Login**

Clave de acceso que se le asigna a un usuario para que pueda utilizar los recursos de una computadora. El login define al usuario y lo identifica dentro de internet junto con la dirección electrónica de la computadora que utiliza.

**MAN (Metropolitan Area Network)**

Red de área metropolitana. Red que no va más allá de los 100 km. Equipos de cómputo y sus periféricos conectados en una ciudad o en varias forman una MAN.

**Mbps (Mega Bits Per Second)**

Megabits por segundo.

**MBps (Mega Bytes Per Second)**

Megabytes por segundo.

**MIME (Multipurpose Internet Mail Extensions)**

Extensiones de Correo de Internet de Múltiples Propósitos Técnica para codificar archivos y anexarlos a un mensaje de correo electrónico. Permite principalmente enviar archivos binarios como parte de un mensaje.

**Modelo Cliente/Servidor.**

El modelo cliente/servidor se apoya en terminales (clientes) conectadas a una computadora que los provee de un recurso (servidor). De esta manera los clientes son los elementos que necesitan servicios del recurso y el servidor es la entidad que poseen el recurso. El tráfico en la red de esta forma se ve aligerado y las comunicaciones entre las computadoras se realizan más rápido.

**MPEG (Moving Pictures Expert Group)**

MPEG es un estándar de compresión de vídeo. Se adhiere a los visualizadores por medio de un plugin.

**NCSA (National Center for Supercomputing Applications)**

Centro Nacional de Aplicaciones de Supercómputo. Desarrolladores del visualizador Mosaic para el World Wide Web. Localizado en. <http://www.ncsa.uiuc.edu/>

### **Netscape Navigator**

Visualizador para el World Wide Web. Disponible en <http://www.netscape.com>, en las siguientes plataformas X-Windows (UNIX), Macintosh y Windows.

### **NSFNET (National Science Foundation)**

Red de la Fundación Nacional de la Ciencia. Fue de las primeras redes académicas en hacerse cargo de Internet y es actualmente eje central de la misma. Información disponible en <http://www.nsf.gov>

### **OSI (Open System Interconnection)**

Interconexión de Sistemas Abiertos. Un modelo de referencia que fue definido por la ISO como un estándar para las comunicaciones mundiales. Define una estructura para implementación de protocolos en siete estratos o capas: Física, Datos, Red, Transporte, Sesión, Presentación y Aplicación.

### **Página Web**

Es el resultado en hipertexto e hipermedia que proporciona un visualizador de World Wide Web después de obtener la información solicitada.

### **Par trenzado**

Parecido al cable utilizado para teléfonos, pero con una cantidad mayor de cables dentro. Es el medio físico por medio del cual se pueden conectar varias computadoras.

### **Password o contraseña**

Palabra clave que se le asigna a un usuario (además de su login) como contraseña para la utilización de los recursos de una computadora. El password no es visible en la pantalla al momento de teclearlo.

### **Perl**

Lenguaje de programación, principalmente utilizado para realizar consultas a bases de datos como Oracle, SQL-Server, SyBase, etc., o a herramientas locales como WAIS. Perl es un lenguaje para manipular textos, archivos y procesos, proporciona una forma fácil y legible para realizar trabajos que normalmente se realizarían en C o en un shell.

### **Plugins**

Programas que se agregan a un visualizador para realizar funciones determinadas. Estas pueden ser visualización de archivos multimedia, soporte a archivos gráficos no estándares con el visualizador, etc.

### **Protocolo**

Es la definición de como deben comunicarse dos computadoras, sus reglas de comportamiento, etc.

### **Puerto**

a) Es un número que identifica a una aplicación particular de Internet

b) Uno de los canales de entrada/salida de una computadora.

### **RED**

Agrupación tanto de equipos como de programas que comparten recursos entre sí, observando "reglas de comportamiento" a partir del uso de un lenguaje y medios de transmisión comunes, sin importar -en lo esencial- la naturaleza de cada elemento dentro de la red.

### **RFC (Request for Comments)**

Solicitud para comentarios. Es un conjunto de documentos en los cuales los estándares de Internet, los estándares propuestos y, generalmente las ideas en proceso de aceptación son documentadas y publicadas.

### **Ruteador**

Elemento que determina la trayectoria más eficiente de datos entre dos segmentos de red. Opera en la capa superior del modelo OSI. No está limitado por protocolos de acceso o medio.

### **Shareware**

Programas que pueden ser obtenidos, por Internet, de computadoras con archivos públicos. La regla de utilización es que se paguen después de un periodo de evaluación (por lo regular 30 días).

### **SMTP (Simple Mail Transfer Protocol)**

Protocolo que se usa para transferir correo electrónico entre servidores de correo. Como sólo transfiere mensajes entre servidores, el usuario debe utilizar otro protocolo para acceder los mensajes como POP o IMAP

### **TCP (Transmission Control Protocol)**

Protocolo de control de transmisión. Es el protocolo que se encarga de la transferencia de los paquetes a través de Internet. Se encarga de que los paquetes lleguen al destino sin ningún error o pide su reenvío.

### **Token Passing (Paso de señales)**

Protocolo que se utiliza en redes Arcnet y Token Ring, y que se basa en un esquema libre de colisiones, dado que la señal (token) se pasa de un nodo o estación al siguiente nodo. Con esto se garantiza que todas las estaciones tendrán la misma oportunidad de transmitir y que un sólo paquete viajará a la vez en la red.

### **Token Ring**

Red local desarrollada por IBM que utiliza el protocolo de acceso Token Passing y que utiliza un ancho de banda de 4 y 16 Mbps. Utiliza la topología de anillo.

### **Topología de anillo**

Topología en donde las estaciones de trabajo se conectan físicamente en un anillo, terminando el cable en la misma estación de donde se originó.

### **Topología de bus**

Topología en donde todas las estaciones se conectan a un cable central llamado "bus". Este tipo de topología es fácil de instalar y requiere menos cable que la topología de estrella.

### **Topología de estrella**

Topología donde cada estación se conecta con su propio cable a un dispositivo de conexión central, bien sea un servidor de archivos, un concentrador o un repetidor.

### **Topología de red**

Se refiere a como se establece y se cablea físicamente una red. La elección de la topología afectará la facilidad de la instalación, el costo del cable y la confiabilidad de la red. Tres de las topologías principales de red son la topología de bus, de estrella, y de anillo.

### **UDP (User Datagram Protocol)**

Protocolo de Datagramas de Usuario. Protocolo que no pide confirmación de la validez de los paquetes enviados por la computadora emisora. Este protocolo es actualmente usado para la transmisión de sonido y vídeo a través de Internet. El UDP está diseñado para satisfacer necesidades concretas de ancho de banda, como no reenvía los datos perdidos, es ideal para el tráfico de voz digitalizada, pues un paquete perdido no afecta la calidad del sonido. Entre las aplicaciones que utilizan este protocolo encontramos a Real Audio.

### **UNIX**

Sistema operativo especializado en capacidades de multiusuario y multitarea. Fue la base inicial de Internet. Entre sus características más importantes se encuentran:

- Redireccionamiento de Entrada/Salida
- Alta portabilidad al estar escrito en lenguaje C, lo que lo hace independiente del *hardware*
- Interfase simple e interactiva con el usuario

### **URL (Uniform Resource Locator)**

Localizador Uniforme de Recursos. Sistema de direccionamiento estándar para archivos y funciones de Internet, especialmente en el World Wide Web. El URL está conformado por el servicio (p. e. <http://>) más el nombre de la computadora (p. e. [www.ine.gob.mx](http://www.ine.gob.mx)) más el directorio y el archivo referido (</noticias/noticias.html>).

### **USENET (USER NETWORK)**

Sistema de redes que contiene artículos llamados NEWS que pueden ser consultados sin necesidad de estar inscrito.

### **Usuario**

Un usuario es la persona que tiene una cuenta en una determinada computadora por medio de la cual puede acceder a los recursos y servicios que ofrece una red. Un usuario que reside en una determinada computadora tiene una dirección electrónica única.



**Vídeo conferencia**

Sistema que permite la transmisión en tiempo real de vídeo, sonido y texto a través de una red, ya sea de área local (LAN) o global (WAN). El *hardware* necesario es tarjeta de sonido y vídeo, vídeo cámara, micrófono y bocinas. La velocidad de transmisión lograda actualmente es de 10 cuadros por segundo.

**Vínculo (link)**

Es un indicador de texto o una imagen que sirve como enlace a otro documento.

**Visualizador (Browser)**

Programa que despliega la información almacenada en páginas HTML que se encuentran disponibles en servidores del World Wide Web. Como ejemplo de visualizadores tenemos Cello, Internet Explorer, Mosaic, Netscape, etc.

**VRML (Virtual Reality Modeling Language antes Virtual Reality Markup Language)**

Lenguaje de programación utilizado para hacer presentaciones de realidad virtual en el World Wide Web. Puede ser un visualizador propio o integrado a los visualizadores WWW a través de un *plugin*.

**W30 (World Wide Web Organization)**

Es la organización dedicada al desarrollo e implantación de nuevos estándares para el World Wide Web. Localizada en <http://www.3w.org/>

**WAN (World Area Network)**

Red de área mundial. Puede extenderse a todo un país o a muchos a través del mundo.

**Website**

Conjunto de páginas Web que comparten un mismo tema e intención y que generalmente se encuentra en un sólo servidor, aunque esto no es forzoso.

**World Wide Web**

Sistema mundial basado en hipertextos cuya función es buscar y tener acceso a documentos a través de la red.