UNIVERSIDAD TÉCNICA DE COTOPAXI

CARRERA DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

ESPECIALIDAD DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

TEMA: IMPLEMENTACIÓN DE UN SERVIDOR WEB SSL (Secure Socket Layer) CON ENCRIPTACIÓN A 128 BITS BAJO PLATAFORMA LINUX EN PETROECUADOR.

PROYECTO DE TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

EGDA. CAISAGUANO CHANGOTÁSIG ANITA MARGOTH

DIRECTOR: ING. JESÚS GONZÁLEZ

DICIEMBRE 2003

LATACUNGA - ECUADOR

AUTORÍA

Del contenido de la presente tesis, se responsabiliza en su totalidad la autora.

....

ANITA MARGOTH CAISAGUANO CHANGOTÁSIG

C.I. 050240860-2

CERTIFICADO

En cumplimiento a lo estipulado en el artículo 9 literal f) del Reglamento del curso Preprofesional de la Universidad Técnica de Cotopaxi, en calidad de director de tesis del tema "IMPLEMENTACIÓN DE UN SERVIDOR WEB SSL (Secure Socket Layer) CON ENCRIPTACIÓN A 128 BITS BAJO PLATAFORMA LINUX EN PETROECUADOR", propuesto por la Egda. Anita Margoth Caisaguano Changotásig debo confirmar que el presente trabajo de investigación fue desarrollado de acuerdo a los planeamientos formulados por la denuncia y construcción teórica del objeto de estudio.

La claridad y veracidad de su contenido a más del desempeño y dedicación puesto por su autora en cada etapa de su realización merece especial atención y su consideración como trabajo de calidad.

En virtud de lo antes expuesto considero que la autora de la presente tesis se encuentra habilitada para presentarse al acto de defensa de tesis.

……………………………………………………

Ing. Jesús González

DIRECTOR DE TESIS

CERTIFICADO

A QUIEN INTERESE:

Por el presente cúmpleme certificar que la señorita **ANITA MARGOTH CAISAGUANO CHANGOTÁSIG**, realizó su tesis de graduación en la Unidad de Sistemas de PETROECUADOR, desde el mes de octubre del 2002 hasta el mes de octubre del 2003 en el desarrollo del proyecto "IMPLEMENTACIÓN DE UN SERVIDOR WEB SSL (Secure Socket Layer) CON ENCRIPTACIÓN A 128 BITS BAJO PLATAFORMA LINUX EN PETROECUADOR" y una aplicación web realizada con PHP, INTERBASE Y LINUX.

Es todo cuanto puedo afirmar en honor a la verdad por lo tanto la señorita puede hacer uso de esta certificación en la forma que considere conveniente.

Quito, noviembre 18 del 2003

...

Ing. José Vanoni M.

JEFE UNIDAD DE SISTEMAS

PETROECUADOR

AGRADECIMIENTO

A mis padres, quienes estuvieron todos los días a mi lado apoyándome en los desvelos, las frustraciones, las preocupaciones, las alegrías y hoy en la culminación de una de las metas que me propuse en la vida.

A la Universidad Técnica de Cotopaxi, sus autoridades y catedráticos, por orientarme con sus valiosos conocimientos y sabios consejos para mi formación académica y personal.

A mi Director de tesis Ing. Jesús González, por haberme guiado en el desarrollo de mi proyecto y sus importantes sugerencias.

Al personal de la Unidad de Sistemas de PETROECUADOR, en especial a los Ingenieros José Vanoni y Gustavo Palacios por haber sugerido el tema y el apoyo brindado durante la realización del proyecto.

A William y a mis amigos/as que de una u otra manera me apoyaron en todo lo que estuvo a su alcance.

DEDICATORIA

Al culminar una etapa más de mi formación académica dedico este esfuerzo y sacrificio a la Virgencita de Baños por bendecirme, cuidarme en los momentos más difíciles de mi vida y por darme la fuerza para seguir adelante en esta etapa importante y difícil de mi carrera universitaria.

A mis padres Miguel y María por su infinito cariño, apoyo, comprensión y haberme guiado por el camino del bien, lo cual me permitió alcanzar esta anhelada ingeniería.

INDICE GENERAL

Pág.

PORTADA I

PÁGINA DE AUTORÍA II

CERTIFICACIÓN DEL DIRECTOR DE TESIS.. III

CERTIFICACIÓN DEL JEFE DE UNIDAD DE PETROECUADOR................ IV

AGRADECIMIENTO V

DEDICATORIA VI

ÍNDICE GENERAL VII

ÍNDICE DE GRÁFICOS XIV

ÍNDICE DE TABLAS XVIII

RESUMEN XX

ABSTRACT XXII

INTRODUCCIÓN XXIV

CAPITULO I

FUNDAMENTO TEÓRICO

1.1. Introducción 1

1.2. Criptografía simétrica o privada 3

1.2.1 Clasificación 4

1.2.2 DES (Data Encryption Standar) y TDES (Triple Data

Encryption Estándar) 4

1.2.3 Funciones Hash 6

1.3 Criptografía Asimétrica o Pública.. 7

1.3.1 Clasificación 8

1.3.2 Firma Digital 8

1.3.3 RSA (Rivest Shamir Adleman) 9

1.3.3.1 Esquema de cifrado.. 9

1.3.3.2 Esquema de firma digital.. 11

1.4 Servidores seguros 12

1.4.1. URL (Uniform Resorce Locator) del servidor seguro...................... 13

1.4.2 Autoridad Certificadora.. 15

1.4.2.1. Creando un Nivel-Raíz de Autoridades Certificadoras....... 15

1.5 Certificados digitales 16

1.5.1. Formato de los certificados digitales.. 17

1.5.2 Elaboración de un certificado digital.. 18

1.5.3 Comprobación de la validez del certificado digital........................ 19

1.5.4 Tipos de certificados.. 20

1.6 Infraestructura de claves públicas.. 21

1.7 Protocolos de seguridad 23

1.7.1 Protocolo SSL(Secure Socket Layer).. 23

1.7.2 Establecimiento de sesión SSL... 25

1.7.2.1 Método de intercambio de clave.. 27

1.7.2.2 Cifrado para la transferencia de datos................................. 28

1.7.2.3 Funciones de resumen... 29

1.7.2.4 Protocolo de secuencia de acuerdos 30

1.7.2.5 Transferencia de datos... 32

1.7.2.6 Asegurando las comunicaciones http.................................. 33

CAPITULO II

ARQUITECTURA DEL SERVIDOR WEB SSL

2.1 Introducción 34

2.2 Modelo cliente / servidor.. 35

2.2.1 Estructura del cliente.. 36

2.2.2 Estructura del servidor.. 37

2.2.3 Componentes esenciales de la infraestructura cliente / servidor.... 38

2.2.3.1 Plataforma operativa 39

2.2.3.2 Entorno de desarrollo de aplicaciones.................................. 39

2.2.3.3 Gestión de sistemas... 40

2.3 Arquitectura de software de dos capas.. 40

2.4 Arquitectura de software de tres capas... 41

2.5 Cliente servidor en TCP / IP.. 43

2.6 Características funcionales.. 45

2.6.1 Primer nivel 46

2.6.2 Segundo nivel 46

2.6.3 Tercer nivel 46

2.6.4 Cuarto nivel 47

2.6.5 Quinto nivel 47

2.7 Características físicas 47

2.8 Características lógicas 48

2.9 Ventajas e inconvenientes... 49

2.9.1 Ventajas 49

2.9.2 Inconvenientes 51

2.10 Introducción a las aplicaciones Web.. 52

2.10.1 Entorno de implementación 54

2.11 Paradigma construcción de prototipos.. 55

2.11.1 Características 57

2.11.2 Ventajas 57

2.11.3 Desventajas 58

2.12 Comercio electrónico 58

CAPITULO III

IMPLEMENTACIÓN DEL SERVIDOR WEB SSL

3.1 Sistema Operativo RED HAT LINUX.. 61

3.1.1 Características generales.. 62

3.2 Servidor Web apache.. 63

3.2.1 Características generales.. 64

3.2.2 Requisitos 65

3.3 Instalación del servidor Web apache con seguridades.............................. 66

3.3.1 Intalación del Interbase 6.0.. 67

3.3.2 Instalación del servidor SSL.. 73

3.3.3 Ejecución del servidor SSL.. 84

3.3.4 Pasos previos para comprar un certificado a la Autoridad Certificadora Verising.. 91

3.3.4.1 Comprobación del certificado...................................... 98

CAPITULO IV

DESARROLLO DE LA APLICACIÓN VALIDACIÓN DE USUARIO Y CONTRASEÑA

4.1 Diseño de la aplicación.. 100

4.1.1 Introducción 100

4.1.2 Objetivos 100

4.1.3 Justificación 101

4.1.4 Análisis del sistema.. 101

4.1.5 Metodología de desarrollo del sistema.. 102

4.1.6 Diagrama de contexto.. 103

4.1.7 Modelos 103

4.1.8 Definición de procesos.. 104

4.1.8.1 Diagrama de flujo de datos.. 104

4.1.8.2 Glosario de procesos.. 105

4.1.8.3 Glosario de flujo de datos... 107

4.1.8.4 Glosario de almacenamiento de datos 108

4.1.8.5 Glosario de entidades.. 109

4.1.8.6 Glosario de documentos de entrada y de salida........................ 109

4.1.9 Definición del prototipo.. 110

4.1.10 Definición de páginas 110

4.1.11 Diagrama jerárquico de páginas 115

4.1.12 Características físicas de almacenamiento.. 116

4.1.13 Diagrama de comunicación de red de datos.. 118

4.1.14 Requerimientos de Hardware y Software.......................... 119

4.1.15 Estimaciones sobre archivos.. 120

4.1.16 Etapa de construcción... 121

4.1.17 Etapa de implantación.. 123

4.2 Ejecución de la aplicación validación de usuario y contraseña.................. 124

4.2.1 Introducción... 124

4.2.2 Objetivos de la aplicación.. 125

4.2.3 Ingreso a la aplicación... 125

4.2.4 Conclusiones de la aplicación.. 134

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Verificación de objetivos... 135

5.2 Conclusiones 136

5.3 Recomendaciones 137

BIBLIOGRAFÍA 139

GLOSARIO DE TÉRMINOS

ANEXOS

INDICE DE GRÁFICOS

Gráfico # 1. Proceso de cifrado con TDES... 6

Gráfico # 2. Certificado digital.. 7

Gráfico # 3. Jerarquía de Autoridades Certificadoras... 21

Gráfico # 4. Secuencia simplificada de acuerdos SSL... 26

Gráfico # 5. Protocolo de secuencia de acuerdos.. 30

Gráfico # 6. Protocolo SSL apilado.. 31

Gráfico # 7. Protocolo de registro SSL... 32

Gráfico # 8. Aplicaciones de dos capas.. 40

Gráfico # 9. Aplicación de tres capas... 42

Gráfico # 10. Características funcionales del modelo cliente / servidor................. 45

Gráfico # 11. Construcción de prototipos.. 56

Gráfico # 12. Consola de trabajo... 67

Gráfico # 13. Opciones para instalar el Interbase 6.0.. 68

Gráfico # 14. Contrato de la licencia de Interbase 6.0... 69

Gráfico # 15. Ingreso de licencias del Interbase 6.0... 70

Gráfico # 16. Archivo profile... 71

Gráfico # 17. Archivo hosts.equiv.. 72

Gráfico # 18. Inicio del servidor de Interbase 6.0.. 72

Gráfico # 19. Consola de trabajo... 73

Gráfico # 20. Instalación del paquete Openssl.. 74

Gráfico # 21. Instalación del paquete MM Shared Memory............................. 75

Gráfico # 22. Instalación del paquete Mod_ssl.. 76

Gráfico # 23. Instalación del paquete apache.. 77

Gráfico # 24. Recompilación del paquete apache.. 78

Gráfico # 25. Datos para la Autoridad Certificadora... 79

Gráfico # 26. Datos para el servidor... 79

Gráfico # 27. Ingreso de claves para la Autoridad Certificadora

y para el servidor.. 80

Gráfico # 28. Inicio del servidor SSL.. 81

Gráfico # 29. Actualización del apxs.. 82

Gráfico # 30. Instalación de los paquetes php4 e Interbase 6.0.......................... 83

Gráfico # 31. Archivo http.conf.. 84

Gráfico # 32. Nestcape Navigator... 85

Gráfico # 33. Generación del certificado (parte I).. 86

Gráfico # 34. Generación del certificado (parte II).. 86

Gráfico # 35. Generación del certificado (parte III).. 87

Gráfico # 36. Generación del certificado (parte IV)... 87

Gráfico # 37. Generación del certificado (parte V)... 88

Gráfico # 38. Generación del certificado (parte VI).. 88

Gráfico # 39. Aviso de transmisión de información segura.................................. 89

Gráfico # 40. Pagina index.html.. 89

Gráfico # 41. Certificado.. 90

Gráfico # 42. Archivo CSR.. 93
Gráfico # 43. Demanda de un certificado a Verising.. 94
Gráfico # 44. Aplicación para certificado de Verising.. 96
Gráfico # 45. Página e inicio del certificado de Verising...................................... 99
Gráfico # 46. Diagrama de contexto nivel 0... 103
Gráfico # 47. Diagrama conceptual de datos.. 103
Gráfico # 48. Diagrama físico de datos.. 104
Gráfico # 49. Nivel 1.. 104
Gráfico # 50. Nivel 2.. 105
Gráfico # 51. Prototipo.. 110
Gráfico # 52. Página index.html... 110
Gráfico # 53. Página validar.html... 111
Gráfico # 54. Página graba_validar.html.. 111
Gráfico # 55. Página menu.html... 112
Gráfico # 56. Página registro.html... 112
Gráfico # 57. Página graba_registro.html.. 113
Gráfico # 58. Página bitácora.html.. 113
Gráfico # 59. Página graba_bitácora.html... 114
Gráfico # 60. Página consulta.html.. 114
Gráfico # 61. Jerarquía de páginas.. 115
Gráfico # 62. Red de datos.. 118

Gráfico # 63. Página index.html.. 126

Gráfico # 64. Página validar.html... 127

Gráfico # 65. Mensaje de error de validación.. 127

Gráfico # 66. Página graba_validar.html.. 128

Gráfico # 67. Página menu.html.. 128

Gráfico # 68. Página consulta.html... 130

Gráfico # 69. Página registro.html.. 130

Gráfico # 70. Mensaje de error de registro... 131

Gráfico # 71. Página graba_registro.html... 131

Gráfico # 72. Página bitácora.html... 132

Gráfico # 73. Mensaje de error de registro de bitácora... 133

Gráfico # 74. Página graba_bitácora.html.. 133

INDICE DE TABLAS

Tabla # 1. Versiones del protocolo SSL.. 24

Tabla # 2. Software usado en la instalación del servidor Web SSL........................ 36

Tabla # 3. Proceso [1] sistema de validación de usuario y clave........................... 105

Tabla # 4. Proceso [1.1] subproceso validar usuario y clave................................ 106

Tabla # 5. Proceso [1.1.1] subproceso registrar datos.. 106

Tabla # 6. Proceso [1.1.2] subproceso registrar bitácora...................................... 106

Tabla # 7. Proceso [1.1.3] subproceso consultar datos... 107

Tabla # 8. Flujo [1] consulta_datos.. 107

Tabla # 9. Flujo [1.1] consulta_datos... 107

Tabla # 10. Flujo [1.1.1] datos_datos a guardar... 108

Tabla # 11. Flujo [1.1.2] datos bitácora a guardar_datos bitácora......................... 108

Tabla # 12. Flujo [1.1.3] consulta_datos a verificar... 108

Tabla # 13. Lista de almacenes de información.. 108

Tabla # 14. Entidades.. 109

Tabla # 15. Documentos de entrada y salida... 109

Tabla # 16. Tablas principales... 116

Tabla # 17. Tabla lista.. 116

Tabla # 18. Tabla descripción.. 117

Tabla # 19. Tabla bitácora.. 117

Tabla # 20. Requerimientos de Hardware.. 119

Tabla # 21. Requerimientos de Software... 119

Tabla # 22. Estimación de archivos.. 120

Tabla # 23. Mensajes de error del sistema... 122

RESUMEN

El proyecto IMPLEMENTACIÓN DE UN SERVIDOR WEB SSL (Secure Socket Layer) CON ENCRIPTACIÓN A 128 BITS BAJO PLATAFORMA LINUX EN PETROECUADOR se ha desarrollado en las instalaciones de PETROECUADOR y brinda la seguridad a la información que se encuentra en los servidores, evitando que personas inescrupulosas tengan acceso y monitoreen la transmisión de datos ya que pueden alterar, borrar datos o producir daños de cualquier tipo. Para lo cual se hace uso del protocolo SSL el mismo que ha sido desarrollado por Nestcape para dar seguridad a la información que se transmite por la red.

Con el objetivo de comprobar el funcionamiento del servidor web SSL se ha diseñado una aplicación web de validación de usuario y clave, la cual permitirá al servidor y al usuario autenticar y negociar entre ambas partes un algoritmo de encriptación y llaves criptográficas, antes de que se transmita o reciba cualquier información. Es decir, una vez en línea y habiéndose decidido hacer la operación, el navegador se conecta a un servidor seguro SSL, el cual ha sido autenticado o validado por una Autoridad Certificadora, la cual actúa como un "tercero de confianza". El servidor seguro usa su llave privada y genera una sesión segura de conexión con el usuario; el navegador decodifica la llave enviada por el servidor y si la descifra correctamente se abre un canal o conexión segura, y toda la información que se cruce entre las partes estará encriptada o protegida.

Para la aplicación web se ha diseñado una base de datos en Interbase 6.0 donde se almacenan usuarios y sus datos. El usuario que use esta aplicación tendrá la posibilidad de validar su nombre y clave, ingresar nuevos usuarios, consultar usuarios existentes y realizar el correspondiente registro de bitácora.

Finalmente con el servidor web SSL implementado se indican los pasos para adquirir un certificado digital de una autoridad certificadora reconocida como Verising, cuando PETROECUADOR cuente con el presupuesto destinado para la compra de este tipo de seguridad procederá con la implantación correspondiente.

SUMMARY

The project IMPLEMENTATION OF A SERVER WEB SSL (Secure Socket Layer) WITH ENCRIPTACIÓN TO 128 BITS LOW PLATFORM LINUX IN PETROECUADOR has been developed in the facilities of PETROECUADOR and it offers the security to the information that is in the server, avoiding unscrupulous people to have access and monitoreen since the transmission of data they can alter, to erase data or to produce damages of any type. For that which use of the protocol SSL the same one is made that has been developed by Netscape to give security to the information that is transmitted by the net.

With the objective of checking the operation of the server web SSL has been designed an application web of user's validation and key, which will allow to the server and the user to authenticate and to negotiate among both parts an encriptación algorithm and cryptographic keys, before it is transmitted or receive any information. That is to say, once on-line and there being you resolved to make the operation, the navigator is connected a sure server SSL, which has been authenticated or validated by an Authority Certificadora, which acts as a "third of trust". The sure server uses his private key and it generates a sure session of connection with the user; the navigator decodes the key sent by the servant and if it deciphers it correctly a channel or sure connection, and all the information that he/she crosses among the parts opens up it will be encrypted or protected.

For the application web a database has been designed in Interbase 6.0 where users and their data are stored. The user that uses this application will have the possibility to validate his name and key, to enter new users, to consult existent users and to carry out the corresponding binnacle registration.

Finally with the server web implemented SSL the steps are indicate to acquire a digital certificate of an authority grateful certificadora as Verising, when PETROECUADOR has the budget dedicated for the purchase of this type of security it will proceed with the corresponding installation.

INTRODUCCIÓN

En la actualidad las empresas ecuatorianas han dado sus primeros pasos para migrar sus negocios a la red aunque no a la velocidad deseada y exigida por los tiempos actuales. Muchas instituciones públicas y privadas la utilizan para dar a conocer sus actividades, publicar datos de un tema específico o información confidencial para sus clientes la cual se ha visto desprotegida o con poca seguridad. Sin embargo, el número de empresas con presencia en Internet crece a diario, pero muy pocas de ellas tienen implementado un servidor Web en el cual sus transacciones sean seguras, esta situación se presenta ya sea por falta de recursos o información.

PETROECUADOR en los actuales momentos no posee un sistema de encriptación de usuarios y claves para el acceso a los datos que presenta la web por lo que fácilmente personas que conocen de la tecnología pueden acceder a su información. Ante este panorama, es importante que la seguridad de las páginas web sea lo primordial, ya que no es bueno que alguien (los llamados hackers) monitoree y puedan obtener las claves y usuarios a bases de datos, información confidencial, o que puedan atacar el sistema obteniendo el acceso al mismo e interrumpir los servicios, inutilizar los sistemas o alterar, suprimir o robar información.

Así mismo para resolver el problema planteado se ha definido el objetivo general de la propuesta que consiste en implementar un servidor web SSL (Secure Socket Layer) con encriptación a 128 bits bajo plataforma Linux, permitiendo realizar transacciones seguras en PETROECUADOR, dicho objetivo general será alcanzado a través del planteamiento de objetivos específicos los mismos que se enmarcan en investigar el protocolo SSL y modelos de encriptación, determinar el modelo de aplicación para orientarlo a que sus transacciones sean seguras, establecer y configurar las herramientas informáticas para la implementación del servidor web SSL y finalmente realizar una aplicación web para comprobar el funcionamiento del servidor implementado.

Con este proyecto se dotará a PETROECUADOR de un mecanismo de seguridad mediante la encriptación del nombre de usuario y su contraseña para tener acceso a las páginas web y de esta manera proporcionar confidencialidad, integridad y disponibilidad de la información a los usuarios.

CAPITULO I

FUNDAMENTO TEÓRICO

1.1. INTRODUCCION

La palabra criptografía proviene del griego kryptos, que significa esconder y grápheín, escribir, es decir, escritura escondida. La criptografía ha sido usada a través de los años para mandar mensajes confidenciales cuyo propósito es que sólo las personas autorizadas puedan entender el mensaje.

Alguien que quiere enviar información confidencial aplica técnicas criptográficas para poder “esconder” el mensaje (llamado cifrar o encriptar), envía el mensaje por una línea de comunicación que se supone insegura y después solo el receptor autorizado pueda leer el mensaje “escondido” (llamado descifrar).

La criptografía actual se inicia en la segunda mitad de la década de los años 70. No es hasta la invención del sistema conocido como DES (Data Encryption Standard) en 1976 que se da a conocer mas ampliamente, principalmente en el mundo industrial y comercial. Posteriormente con el sistema RSA (Rivest, Shamir, Adleman) en 1978, se abre el comienzo de la criptografía en un rango de aplicaciones como: en transmisiones militares, financieras, comunicación de satélite, redes de computadoras, líneas telefónicas, transmisiones de televisión, etc. Los principales problemas de seguridad que resuelve la criptografía son: la privacidad, la integridad, la autenticación y el no rechazo.

La privacidad, se refiere a que la información sólo pueda ser leída por personas autorizadas. Ejemplos: en la comunicación por teléfono, que alguien intercepte la comunicación y escuche la conversación quiere decir que no existe privacidad. En la comunicación por Internet es muy difícil estar seguros que la comunicación es privada, ya que no se tiene control de la línea de comunicación. Por lo tanto al cifrar la información cualquier intercepción no autorizada no podrá entender la información. Esto es posible si se usan técnicas criptográficas, en particular la privacidad se logra si se cifra el mensaje con un método simétrico.

La integridad, se refiere a que la información no pueda ser alterada en el transcurso de ser enviada. Ejemplos: cuando compramos un boleto de avión y están cambiados los datos del vuelo, puede afectar los planes del viajero. En Internet las compras se puede hacer desde dos ciudades muy distantes, la información tiene necesariamente que viajar por una línea de transmisión de la cual no se tiene control, si no existe integridad podrían cambiarse por ejemplo el número de una tarjeta de crédito, los datos del pedido, en fin información que causaría problemas a cualquier comercio y cliente.

La autenticidad, se refiere a que se pueda confirmar que el mensaje recibido haya sido enviado por la persona correcta. Ejemplo: cuando se quiere cobrar un cheque a nombre de alguien, quien lo cobra debe someterse a un proceso de verificación de identidad para la comprobación correspondiente. En Internet es muy fácil engañar a una persona con quien se tiene comunicación respecto a la identidad, por lo cual para resolver este problema se usan técnicas que verifican la autenticidad tanto de personas como de mensajes, quizá la más conocida aplicación de la criptografía asimétrica es la firma digital que reemplaza a la firma autógrafa que se usa comúnmente.

El no rechazo, se refiere a que no se pueda negar la autoría de un mensaje enviado.

La criptografía se divide en dos grandes ramas, la criptografía de clave privada, simétrica o DES y la criptografía de clave pública, asimétrica o RSA.

1.2. CRIPTOGRAFÍA SIMÉTRICA O PRIVADA

La criptografía simétrica se refiere al conjunto de métodos que permiten tener comunicación segura entre las partes siempre y cuando anteriormente se hayan intercambiado la clave correspondiente que llamaremos clave simétrica. La simetría se refiere a que las partes tienen la misma llave tanto para cifrar como para descifrar.

La criptografía simétrica llamada también criptografía de llave privada ha sido la más usada en toda la historia, ésta ha podido ser implementada en diferentes dispositivos, manuales, mecánicos, eléctricos, hasta los algoritmos actuales que son programables en cualquier computadora. La idea general es aplicar diferentes funciones al mensaje que se quiere cifrar de tal modo que solo conociendo una clave pueda aplicarse de forma inversa para poder así descifrar.

1.2.1 Clasificación

Existe una clasificación de este tipo de criptografía en tres familias y son:

a) La criptografía simétrica de bloques (block cipher).
b) La criptografía simétrica de lluvia (stream cipher)
c) La criptografía simétrica de resumen (hash functions).

1.2.2 DES (Data Encryption Standard) y TDES (Triple Data Encryption Standard)

DES es un sistema criptográfico que toma como entrada un bloque de 64 bits del mensaje y este se somete a 16 interacciones, una clave de 56 bits, en la práctica el bloque de la clave tiene 64 bits, ya que a cada conjunto de 7 bits se le agrega un bit que puede ser usada como de paridad.

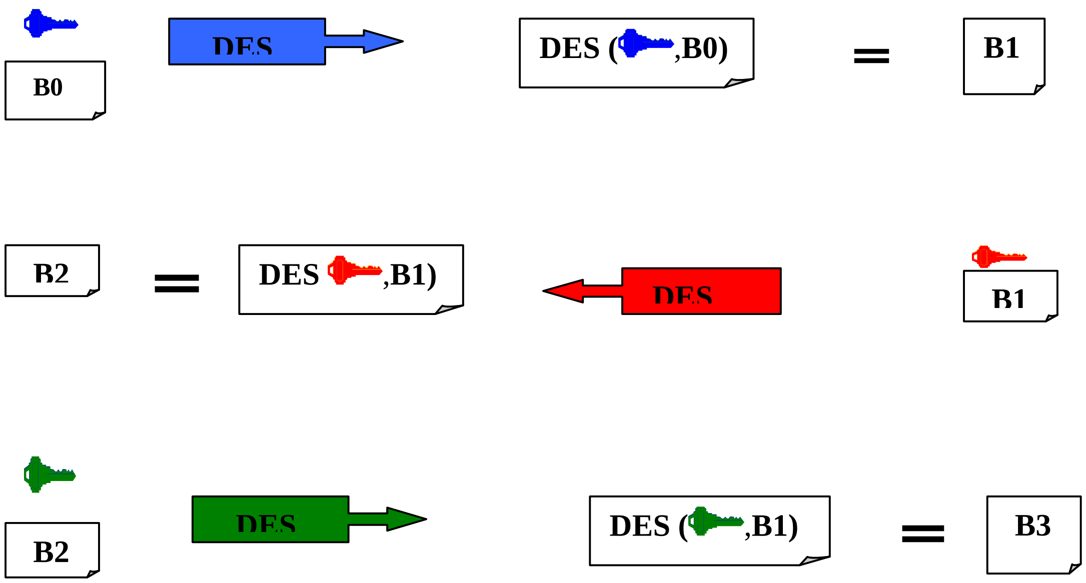
Dependiendo de la naturaleza de la aplicación DES tiene 4 modos de operación para poder implementarse: ECB (Electronic Codebook Mode) para mensajes cortos, de menos de 64 bits, CBC (Cipher Block Chaining Mode) para mensajes largos, CFB (Cipher Block Feedback) para cifrar bit por bit ó byte por byte y el OFB (Output Feedback Mode) el mismo uso pero evitando propagación de error.

En la actualidad no se ha podido romper el sistema DES desde la perspectiva de poder deducir la clave simétrica a partir de la información interceptada, sin embargo con un método a fuerza bruta, es decir probando alrededor de 2^{56} posibles claves, se pudo romper DES en Enero de 1999.

La opción que se ha tomado para poder suplantar a DES ha sido usar lo que se conoce como cifrado múltiple, es decir aplicar varias veces el mismo algoritmo para fortalecer la longitud de la clave, esto ha tomado la forma de un nuevo sistema de cifrado que se conoce actualmente como triple-DES o TDES.

El funcionamiento de TDES consiste en aplicar 3 veces DES de la siguiente manera: la primera vez se usa una clave K1 junto con el bloque B0, de forma ordinaria E (de Encription), obteniendo el bloque B1. La segunda vez se toma a B1 con la clave K2, diferente a K1 de forma inversa, llamada D (de Descencription) y la tercera vez a B2 con una clave K3 diferente a K1 y K2, de forma ordinaria E (de Encription), es decir, aplica de la interacción 1 a la 16 a B0 con la clave K1, después aplica de la 16 a la 1, a B1 con la clave K2, finalmente aplica una vez mas de la 1 a la 16 a B2 usando la clave K3, obteniendo finalmente a B3. En cada una de estas tres veces aplica el modo de operación más adecuado. Este sistema TDES usa entonces una clave de 168 bits.[1]

Gráfico # 1. PROCESO DE CIFRADO CON TDES



FUENTE: http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html

1.2.3 FUNCIONES HASH

Una herramienta fundamental en la criptografía, son las funciones hash, usadas principalmente para resolver el problema de la integridad de los mensajes, así como la autenticidad de mensajes y de su origen.

1 http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html ; Ultimo acceso: Martes 5 de Agosto del 2003

Las funciones hash más conocidas y que se crean a partir de un block cipher son DES, MD5, SHA-1, y RIPEMD 160. La función hash toma un mensaje para partirlo en pedazos de longitud constante y combinar de alguna forma pedazo por pedazo hasta obtener un solo mensaje de longitud fija.

Una función hash también es usada para la firma digital, ya que los documentos a firmar son en general demasiado grandes, la función hash les asocia una cadena de longitud 160 bits que los hace más manejables para el propósito de firma digital.

1.3 CRIPTOGRAFÍA ASIMÉTRICA O PÚBLICA

La criptografía asimétrica es por definición aquella que utiliza dos claves diferentes para cada usuario, una para cifrar que se le llama clave pública y otra para descifrar que es la clave privada. El nacimiento de la criptografía asimétrica se dio al estar buscando un modo más práctico de intercambiar las llaves simétricas. Es un método de Rivest Shamir y Adleman RSA publicado en 1978, cuando toma forma la criptografía asimétrica, su funcionamiento esta basado en la imposibilidad computacional de factorizar números enteros grandes.

Actualmente la Criptografía asimétrica es muy usada, sus dos principales aplicaciones son el intercambio de claves privadas y la firma digital. Los fundamentos de la criptografía asimétrica pertenecen a la teoría de números.

1.3.1 Clasificación

En la actualidad la criptografía asimétrica o de clave pública se divide en tres familias según el problema matemático del cual basan su seguridad y son:

1. La primera familia se basa su seguridad en el Problema de Factorización Entera PFE, los sistemas que pertenecen a esta familia son: el sistema RSA y el de Rabin Williams RW.
2. La segunda familia se basa su seguridad en el Problema del Logaritmo Discreto PLD, a esta familia pertenece el sistema de Diffie Hellman DH de intercambio de claves y el sistema DSA de firma digital.
3. La tercera familia se basa su seguridad en el Problema del Logaritmo Discreto Elíptico PLDE, en este caso hay varios esquemas tanto de intercambio de claves como de firma digital que son el DHE (Diffie Hellman Elíptico), DSAE, (Nyberg-Rueppel) NRE, etc.

1.3.2 Firma digital

Una firma digital es un código especial, único para cada mensaje, que son generadas utilizando un algoritmo de clave pública. Las firmas digitales son utilizadas para verificar la integridad y autenticidad de un mensaje; además garantizan la no repudiabilidad de un mensaje y por lo tanto tiene el mismo valor legal que una firma convencional.

Para crear una firma digital para un mensaje electrónico se necesita de dos cosas:

- Un certificado de firma que lo identifique para este propósito. Cada vez que firme un mensaje, su certificado de firma se enviará con él.
- Una clave privada, que se crea y se almacena en el ordenador al obtener un certificado por primera vez. La clave privada se protege mediante una contraseña maestra, y normalmente no se revela a nadie.[2]

1.3.3. RSA (Rivest, Shamir, Adleman)

RSA usa la factorización de un número entero n grande (1024 bits), este número entero se sabe es producto de dos números primos p, q de la misma longitud, entonces la clave pública es el número n y la privada es p, q. Existen dos formas de uso del sistema RSA que son: esquema de firma y esquema de cifrado.

1.3.3.1 Esquema de cifrado

Este esquema se usa principalmente en cifrar claves de sistemas simétricos (claves de 128 bits aprox.) y se realiza de la siguiente manera:

2 http://nave.escomposlinux.org/productos/mozilla/1.0.1/progreso/Platform_neutral/help/glossary.html ; Ultimo acceso: Jueves 5 de Junio del 2003

1) Se toma el mensaje m (por ejemplo una clave simétrica de 128 bits), como en la práctica actual es recomendable usar arreglos de longitud de 1024 bits, los complementa esos 128 bits con una serie de técnicas para obtener un arreglo de 1024 bits, después se aplica un proceso de codificación para que la computadora entienda al mensaje como un número entero m.
2) Se le aplica la fórmula de cifrado de RSA al entero m.
3) Se envía el número entero c.
4) Al recibir este número se aplica la fórmula de descifrado al entero c para obtener el entero m.
5) Se decodifica m para obtener el mensaje m.

Ejemplo:

Generación de parámetros

1) $p = 3$, $q = 5$ (se eligen dos números primos como clave privada)
2) $n = 15$ (se calcula el producto, es la clave pública)
3) $\varphi(n)=(3-1)(5-1)=8$
4) Sea $e=3$, entonces $d=3$, ya que $e*d = 3*3 = 9 \bmod 8 = 1$ (como este caso solo es para mostrar el funcionamiento no importa que d sea igual a e, sin embargo en la práctica e es pequeño y d es muy grande)
5) Si el mensaje es $m=2$

Proceso de cifrado

6) El mensaje cifrado es $c= m^e \bmod n$, es decir, $c=2^3 \bmod 15$, o sea $c=8$

Proceso de descifrado

7) Para descifrar el mensaje $m=8^3 \bmod 15$, es decir, $m=512 \bmod 15$, así $m=2$ (ya que $512/15=2 \bmod 15 = m$). Por lo tanto es correcto el funcionamiento.

1.3.3.2 Esquema de firma digital

Existen dos tipos de esquemas sobre firma digital, el que se denomina esquema de firma digital con apéndice y el esquema de firma digital con mensaje recuperable. También cualquier esquema de firma cuenta con dos partes la primera parte se denomina proceso de firma (similar al cifrado) y la segunda parte proceso de verificación de la firma (similar al descifrado). El esquema más usado y conocido es el esquema de firma con apéndice.[3]

Ejemplo:

Se toma los mismos parámetros del ejemplo en el esquema de cifrado, $p=3$, $q=5$, $m=2$, $\varphi=8$, $e=3$, $d=3$

3 http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html ; Ultimo acceso: Martes 5 de Agosto del 2003

Proceso de Firma

1) La firma del documento m es: $s= m^d \bmod n = 2^3 \bmod 15 =8$

2) El mensaje firmado es entonces (m,s) = (2,8)

Proceso de verificación

3) Aplicando la función de verificación $s^e \bmod n = 8^3 \bmod 15 = 2$

4) Como 2 (el obtenido de la anterior fórmula) = 2 (el mensaje enviado)

5) Entonces la firma es válida

1.4 SERVIDORES SEGUROS

Se entiende por Servidor Seguro un servidor de páginas web que establece una conexión cifrada con el cliente que ha solicitado la conexión, de manera que nadie, salvo el servidor y el cliente, puedan tener acceso a la información transmitida de forma útil. El uso de servidores seguros es un elemento imprescindible en todos aquellos servicios que utilicen información confidencial, como operaciones bancarias on-line, compras por Internet, acceso a servidores de datos sensibles, etc.

Para conseguir la confidencialidad e integridad de datos perseguida los servidores seguros se basan en el uso de sistemas criptográficos mixtos, que combinan la Criptografía de clave pública con la de clave simétrica. Pero esta protección que debiera darnos la criptografía es en la práctica difícil de encontrar, debido a las severas leyes de exportación de software de cifrado que impone el gobierno de EEUU, sobre todo en lo que respecta a la longitud de las claves que usan. Para garantizar al usuario su autenticidad, los servidores seguros hacen uso de los certificados digitales.

Cuando se accede a un servidor seguro normalmente aparece una ventana indicando que va a iniciar una conexión segura, y el candado situado en la parte inferior de la ventana del navegador aparecerá cerrado cuando ingrese a la página segura. Además, si se observa en la barra de direcciones se ve que ahora se usa el protocolo HTTPS, que corresponde al protocolo HTTP con privacidad.

1.4.1. URL (Uniform Resorce Locator) del servidor seguro

Como se ve, se tiene una longitud de clave RC4 de 40 bits, limitación impuesta por el gobierno de EEUU al sistema de cifrado para su exportación. Esta longitud de clave no es completamente segura para ser usada en transacciones delicadas; ya ha sido violentada anteriormente, y aunque esto no significa que pueda serlo en el tiempo de duración de la conexión segura, si es un indicio de la debilidad del sistema con esas longitudes de clave. También se puede acceder desde una página segura al certificado digital del servidor en forma rápida. Para ello basta seleccionar el botón "Certificates" de la ventana anterior o hacer doble clic sobre el candado cerrado.

Otro aspecto importante a considerar son los fallos a la hora de implementar los protocolos criptográficos, sobre todo en lo que respecta a la configuración propia del servidor web seguro y a los fallos de implementación que de los protocolos hacen los navegadores cliente. Uno de estos fallos es la relativa falta de seguridad de los números pseudo aleatorios generados para el proceso de creación de claves durante la fase Handshake. Para minimizar los riesgos posibles, a la hora de implementar o aceptar un servicio de servidor seguro se exige que se cumplan una serie de condiciones, entre las que se destacan:

- Que el certificado del servidor seguro proporcione la máxima garantía de verificación y que haya sido expedido por una Autoridad Certificadora de toda confianza tales como: Verisign/RSA, EuroSign y Thawte.

- Que el navegador usado en la comunicación tenga implementada la última versión de SSL, es decir, el protocolo SSL 3.0. Las versiones anteriores son válidas, pero no recomendadas.

- El uso de un sistema de cifrado simétrico robusto (RC4, RC5 o similar) con longitudes de clave largas (entre 64 y 128 bits). [4]

4 http://www.seguridad%20SSL.html ; Ultimo acceso: Viernes 11 de Julio del 2003

1.4.2 Autoridad Certificadora

Es la tercera parte fiable que acredita la correspondencia entre una determinada clave y su propietario real. Actúa como una especie de notario electrónico que extiende un certificado de claves el cual está firmado con su propia clave, para así garantizar la autenticidad de dicha información. La ley de firma electrónica las define como "prestadores de servicios de certificación" y, según la legislación de cada país, "son aquellas personas físicas o jurídicas que expiden certificados, pudiendo prestar, además otros servicios en relación con la firma electrónica".

Para poder implementar la Autoridad Certificadora se utilizan los paquetes OpenSSL, Apache y el módulo mod_ssl, puesto que es software libre y para el montaje se utiliza el sistema operativo Linux. Las principales Autoridades Certificadoras actuales son: Verisign (filial de RSA Data Security Inc.) y Thawte.

1.4.2.1. Creando un Nivel-Raíz de Autoridades Certificadoras

Cada certificado necesita un expendedor que afirme la validez de la identidad del sujeto certificado, hasta llegar a la autoridad certificadora más alta (CA). Pero esto presenta un problema: ¿quién certifica a la autoridad certificadora más alta? En este único caso, el certificado es "auto-firmado", es decir, el expendedor del certificado es el mismo que el sujeto. Como resultado, uno tiene que ser extremadamente cuidadoso al confiar en certificados "auto-firmados". La publicación de una clave pública por la autoridad raíz reduce el riesgo de confianza en esa clave. Algunas compañías, como Thawte y Verising se han establecido ellas mismas como autoridades certificadoras y proporcionan los siguientes servicios: peticiones de verificación, procesar peticiones de certificados, expedición y tratamiento de certificados

También es posible crearse una Autoridad Certificadora propia, aunque tiene riesgos en el entorno de Internet, puede ser útil dentro de una Intranet donde la organización puede fácilmente verificar la identidad de los individuos y los servidores, para lo cual requiere una estructura administrativa, técnica y de dirección sólida. [5]

1.5 CERTIFICADOS DIGITALES

Un Certificado Digital es un documento electrónico que contiene datos identificativos de una persona o entidad (empresa, servidor web, etc.) y la llave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada autoridad certificadora. Los Certificados Digitales están basados en la criptografía de clave pública y en el sistema de firmas digitales.

5 http://www.modssl.org/docs/2.8/ ; Ultimo acceso: Jueves 7 de agosto del 2003

La misión principal de un certificado digital es garantizar con toda confianza el vínculo existente entre una persona, entidad o servidor web con una pareja de claves correspondientes a un sistema criptográfico de clave pública.

1.5.1. Formato de los certificados digitales

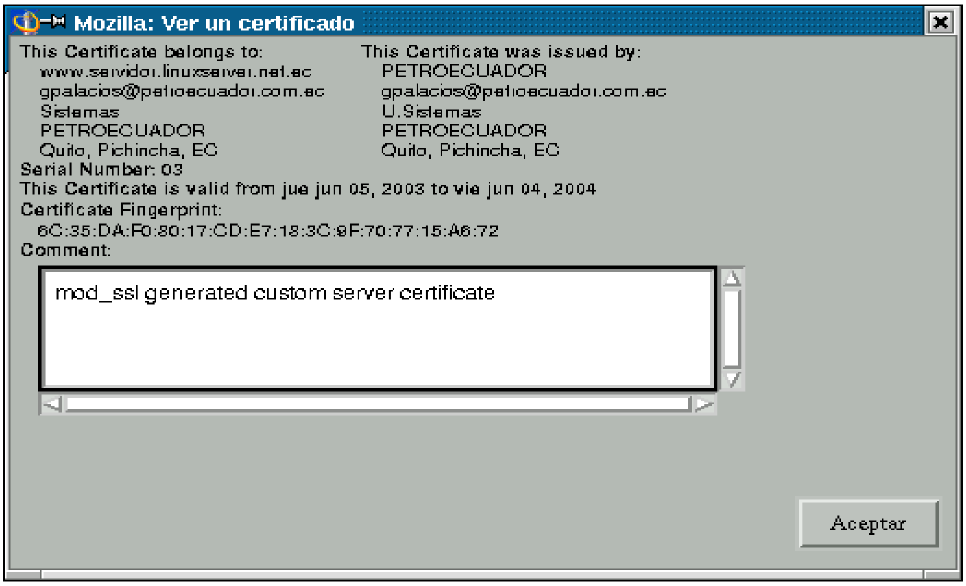
El formato de los certificados digitales es estándar, siendo X.509 v3 el recomendado por la Unión Internacional de Comunicaciones (ITU) y el que está en vigor en la actualidad. Los datos que figuran generalmente en un certificado son:

1) Versión: versión del estándar X.509, generalmente la 3, que es la más actual.
2) Número de serie: número identificador del certificado, único para cada certificado expedido por una AC determinada.
3) Algoritmo de firma: algoritmo criptográfico usado para la firma digital.
4) Autoridad Certificadora: datos sobre la autoridad que expide el certificado.
5) Fechas de inicio y de fin de validez del certificado. Definen el periodo de validez el mismo, que generalmente es de un año.
6) Propietario: persona o entidad vinculada al certificado.
7) Llave pública: representación de la llave pública vinculada a la persona o entidad (en hexadecimal), junto con el algoritmo criptográfico para el que es aplicable.
8) Algoritmo usado para la misma para obtener la firma digital de la Autoridad Certificadora.

9) Firma de la Autoridad Certificadora, que asegura la autenticidad del mismo.

10) Información adicional, como tipo de certificado, etc.[6]

Gráfico # 2. CERTIFICADO DIGITAL



FUENTE: La autora

1.5.2 Elaboración de un certificado digital

El software del firmante aplica un algoritmo hash sobre el texto a firmar, obteniendo un extracto de longitud fija, y absolutamente específico para ese mensaje. Un mínimo cambio en el mensaje produciría un extracto completamente diferente, y por tanto no correspondería con el que originalmente firmó el autor.

6 http://www.seguridad%20SSL.html ;_Ultimo acceso: Viernes 11 de Julio del 2003

Los algoritmos hash más utilizados son el MD5 ó SHA-1. El extracto conseguido, cuya longitud oscila entre 128 y 160 bits (según el algoritmo utilizado), se somete a continuación a cifrado mediante la clave secreta del autor.

El algoritmo más utilizado en este procedimiento de encriptación asimétrica es el RSA. De esta forma se obtiene un extracto final cifrado con la clave privada del autor, el cual se añadirá al final del texto o mensaje para que se pueda verificar la autoría e integridad del documento por aquella persona interesada que disponga de la clave pública del autor.

1.5.3 Comprobación de la validez del certificado digital

Para verificar la validez del documento o archivo, es necesario la clave pública del autor y el procedimiento es el siguiente: el software del receptor previa introducción en el mismo de la clave pública de remitente (obtenida a través de una Autoridad de Certificación), descifra el extracto cifrado del autor y a continuación calcula el extracto hash que le corresponde al texto del mensaje y si el resultado coincide con el extracto anteriormente descifrado, se considera válida; en caso contrario significa que el documento ha sufrido una modificación posterior y por lo tanto no es válido.[7]

7 [http://sigma.poligran.edu.co/politecnico/apoyo/sistemas/dist/docs_011/ssl.doc] ; Ultimo acceso: Jueves 16 de Enero del 2003

1.5.4 Tipos de certificados

Dependiendo del uso que se vaya a dar al certificado y de qué persona o entidad lo solicita, las Autoridades Certificadoras han dividido los certificados en varios tipos. Desde el punto de vista de la finalidad, los certificados electrónicos se dividen en:

1. **Certificados SSL para cliente**: usados para identificar y autenticar a clientes ante servidores en comunicaciones mediante el protocolo Secure Socket Layer, y se expiden a una persona física, un particular o un empleado de una empresa.

2. **Certificados SSL para servidor**: usados para identificar a un servidor ante un cliente en comunicaciones mediante el protocolo SSL, y se expiden a nombre de la empresa propietaria del servidor seguro o del servicio que éste va a ofrecer vinculando también el dominio por el que se debe acceder al servidor.

3. **Certificados S/MIME**: usados para servicios de correo electrónico firmado y cifrado, que se expide a una persona física. El mensaje lo firma digitalmente el remitente, lo que proporciona autenticación, integridad y no rechazo.

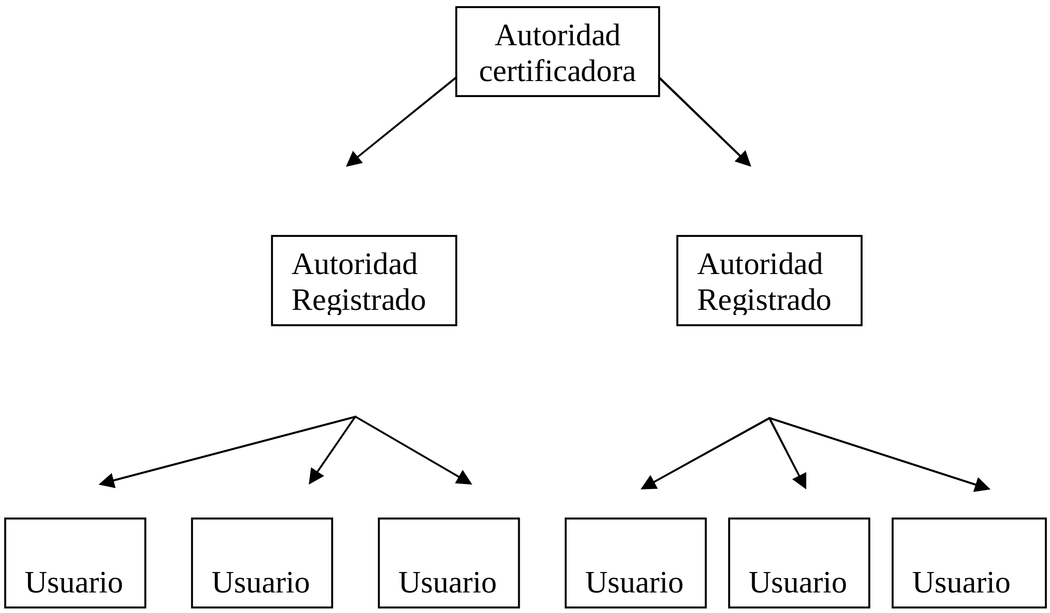
4. **Certificados de firma de objetos**: usados para identificar al autor de ficheros o porciones de código en cualquier lenguaje de programación que se deba ejecutar en red (Java, JavaScript, CGI, etc).

5. **Certificados para AC**: que identifican a las propias Autoridades Certificadoras, y es usado por el software cliente para determinar si pueden confiar en un certificado cualquiera, accediendo al certificado de la AC y comprobando que ésta es de confianza.

1.6 INFRAESTRUCTURA DE CLAVES PÚBLICAS

Teniendo ya un certificado digital que es generado con la ayuda de un algoritmo de clave pública ahora el problema es como administración todos estos, la estructura más básica es la siguiente:

Gráfico # 3. JERARQUÍA DE AUTORIDADES CERTIFICADORAS



FUENTE: [http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html

El papel de la Autoridad certificadora (AC) es de firmar los certificados digitales de los usuarios, generar los certificados, mantener el status correcto de los certificados, esto cumple el siguiente ciclo:

1) La generación del certificado se hace primero por una solicitud de un usuario, el usuario genera sus claves pública y privada y envía junto con los requerimientos de la solicitud su clave pública para que esta sea certificada por la AC.
2) Una vez que la AR (es la AC regional) verifica la autenticidad del usuario, la AC vía la AR firma el certificado digital y es mandado al usuario.
3) El status del usuario puede estar en: activo, inactivo o revocado.
4) Cuando termina el período de activación del certificado el usuario puede solicitar su renovación.

Las operaciones que puede realizar una AC son: generar, revocar, suspender, renovar y mantener un respaldo de los certificados. En cambio las operaciones que puede realizar una AR son: recibir las solicitudes de certificación, el proceso de la autenticación de usuarios, generar las claves, el respaldo de las claves, proceso de recobrar las claves y reportar las revocaciones. Y las actividades de los usuarios son: solicitar el certificado, solicitar la revocación del certificado, solicitar la renovación del certificado. [8]

8 http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html ; Ultimo acceso: Martes 5 de Agosto del 2003

1.7 PROTOCOLOS DE SEGURIDAD

Un protocolo de seguridad es la parte visible de una aplicación, es el conjunto de programas y actividades programadas que cumplen con un objetivo especifico y que usan esquemas de seguridad criptográfica.

Los protocolos de seguridad procuran resolver los problemas de la seguridad como la integridad, la confidencialidad, la autenticación y el no rechazo, mediante sus diferentes características, las cuales se derivan de las múltiples posibilidades con que se puede romper un sistema, es decir, robar información, cambiar información, leer información no autorizada, y todo lo que se considere no autorizado por los usuarios de una comunicación por red. Entre los protocolos de seguridad que existen anotamos los siguientes: SSL(Secure Socket Layer), TLS(Transport Layer Security), SET (Secure Electronic Transactions), etc.

1.7.1 Protocolo SSL (Secure Socket Layer)

El protocolo SSL es un protocolo de capa que esta situado entre un protocolo de capa de red fiable orientado a conexión (como TCP/IP) y el protocolo de la capa de aplicación (como el HTTP). SSL proporciona una comunicación segura entre el cliente y el servidor permitiendo una autentificación mutua, el uso de firmas digitales para comprobar la integridad del mensaje, y la encriptación para la privacidad.

El protocolo esta diseñado para soportar un rango de algoritmos específicos seleccionables para encriptar, resumir y firmar. La elección se negocia entre el servidor y el cliente, al principio del establecimiento del protocolo de sesión. SSL es integrado en el Browser de Netscape y hace su aparición cuando el candado de la barra de herramientas se cierra y también si la dirección de Internet cambia de http a https.

Tabla # 1. VERSIONES DEL PROTOCOLO SSL

VERSIÓN	**FUENTE**	**DESCRIPCIÓN**	**SOPORTE PARA EL NAVEGADOR**
SSL v2.0	Vendor Standard (de Netscape Corp.) [SSL2]	Primer protocolo SSL para el cual existen implementaciones	- NS Navigator 1.x/2.x - MS IE 3.x - Lynx/2.8+OpenSSL
SSL v3.0	Expired Internet Draft (de Netscape Corp.) [SSL3]	Revisiones para prevenir ataques específicos a la seguridad, añade encriptadores no-RSA, y soporte para series de certificados	- NS Navigator 2.x/3.x/4.x/5.x/6.x/7.x - MS IE 3.x/4.x/5.x/6.x Lynx/2.8+OpenSSL
TLS v1.0	Proposed Internet Standard (de IETF) [TLS1]	Revisión del SSL 3.0 que actualiza desde la capa MAC hasta la capa HMAC, añade bloques de relleno para los encriptadores de bloques, estandarización de mensajes y más mensajes de alerta.	- Lynx/2.8+OpenSSL

FUENTE: http://www.modssl.org/doc/2.8/

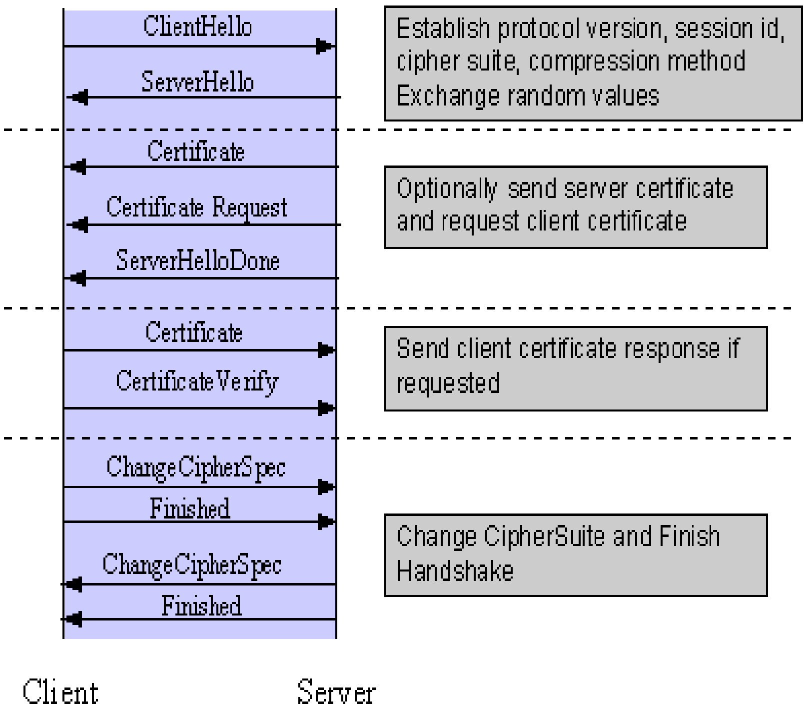
Uno de los beneficios en el SSL 3.0 es que añade soporte para cargar series de certificados. Esta característica permite al servidor pasar un certificado de servidor junto con el certificado del expendedor al navegador.

La carga de series, también permite al navegador validar el certificado de servidor, incluso si los certificados de Autoridad Certificadora de los expendedores intermediarios no han sido instalados, una vez se hayan incluido dentro de la serie certificada. SSL 3.0 es la base para el protocolo estándar Transport Layer Security (TLS), actualmente en desarrollo por el Internet Engineering Task Force (IETF).

1.7.2 Establecimiento de sesión SSL

La sesión SSL se establece siguiendo una secuencia de acuerdos entre el cliente y el servidor, como muestra el gráfico Nº 4. Esta secuencia puede variar, dependiendo de si el servidor esta configurado para proporcionar un certificado de servidor o para pedir un certificado de cliente. Si bien existen casos donde se requieren acuerdos adicionales para el tratamiento de la información cifrada.

Gráfico # 4. SECUENCIA SIMPLIFICADA DE ACUERDOS SSL



FUENTE: http://www.modssl.org/doc/2.8/

Una vez que se ha establecido una sesión SSL ésta se puede reutilizar, así se evita tener que volver a repetir los muchos pasos necesarios para comenzar una sesión. Debido a esto, el servidor asigna a cada sesión SSL un identificador de sesión único que es almacenado en la caché del servidor, y que el cliente puede utilizar en las conexiones que disponga para reducir la secuencia de acuerdos. Los elementos de la secuencia de acuerdos, que utilizan el cliente y el servidor, se muestran seguidamente:

1. Negociar el juego de cifrado que se utilizará durante la transferencia de datos
2. Establecer y compartir la clave de sesión entre el cliente y el servidor
3. Opcionalmente autentificar el servidor por parte del cliente
4. Opcionalmente autentificar el cliente por parte del servidor

El primer paso, negociar el juego de cifrado, permite al cliente y al servidor elegir unas condiciones que puedan ser soportadas por ambos. Las especificaciones del protocolo SSL 3.0 definen 31 juegos de cifrado diferentes. Un juego de cifrado viene definido por los siguientes componentes:

- Método de intercambio de Clave
- Cifrado para la Transferencia de Datos
- Resumen del Mensaje para crear el Código de Autentificación del Mensaje.

1.7.2.1 Método de intercambio de Clave

El método de intercambio de clave, define cómo se acuerda la clave criptográfica simétrica secreta compartida utilizada por la aplicación de transferencia de datos, entre el cliente y el servidor. SSL 2.0 utiliza sólo el intercambio de clave RSA, mientras que SSL 3.0 soporta la elección de varios algoritmos de intercambio de claves, incluido el intercambio de clave RSA (cuando los certificados son utilizados), y el intercambio de clave Diffie-Hellman para intercambiar claves sin certificados y sin comunicación de prioridad entre cliente y servidor.

Una variable en la elección de los métodos de intercambio de claves es la firma digital tanto si la utilizamos o no, y en caso que la utilicemos, qué clase de firmas utilizar. Firmar con la clave privada, proporciona seguridad contra un ataque humano durante la mitad del proceso de intercambio de información que se utiliza para la generación de la clave compartida.

1.7.2.2 Cifrado para la transferencia de datos

SSL utiliza el algoritmo criptográfico convencional (criptografía simétrica) para encriptar mensajes en una sesión. Hay nueve posibles elecciones, incluyendo la elección de ejecución sin encriptación:

- Sin encriptación
- Cifrado de Ráfagas
    - RC4 con claves de 40-bits
    - RC4 con claves de 128-bits
- CBC Cifrado de Bloques
    - RC2 con clave de 40 bits
    - DES con clave de 40 bits

- DES con clave de 54 bits
- Triple-DES con clave de 168 bits
- Idea (clave de 128 bits)
- Fortezza (clave de 96 bits)

“CBC” se refiere a Series de Bloques Cifrada, que significa que una porción del texto cifrado previamente encriptado se utiliza en la encriptación del bloque actual. “DES” se refiere al Estándar de Encriptación de Datos que tiene un número de variantes. “Idea” es uno de los mejores algoritmos disponibles y el más fuerte criptográficamente hablando, y “RC2” es un algoritmo cuyo propietario es RSA DSI.

1.7.2.3 Funciones de Resumen

La elección de las funciones de resumen determinan cómo se crea un registro desde una unidad de registro. El resumen de mensaje se utiliza para crear un Código de Autentificación de Mensaje (MAC) el cual es encriptado con el mensaje para proporcionar integridad y para prevenir contra repeticiones de ataques. SSL soporta los siguientes:

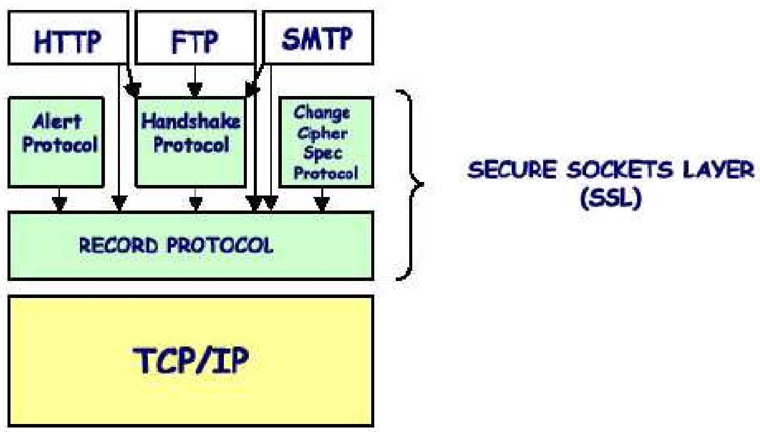
- Sin resumen (elección Null)
- MD5, un hash de 128-bits
- Secure Hash Algorithm (SHA-1), un hash de 160-bits

1.7.2.4 Protocolo de Secuencia de Acuerdos

La secuencia de acuerdos utiliza tres protocolos:

- El Protocolo **SSL Record** especifica la forma de encapsular los datos transmitidos y recibidos.
- El protocolo **SSL Handshak**e, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad.
- El Protocolo de **Alerta SSL** para transferir los mensajes de error SSL entre el cliente y el servidor.[9]

Gráfico # 5. PROTOCOLO DE SECUENCIA DE ACUERDOS

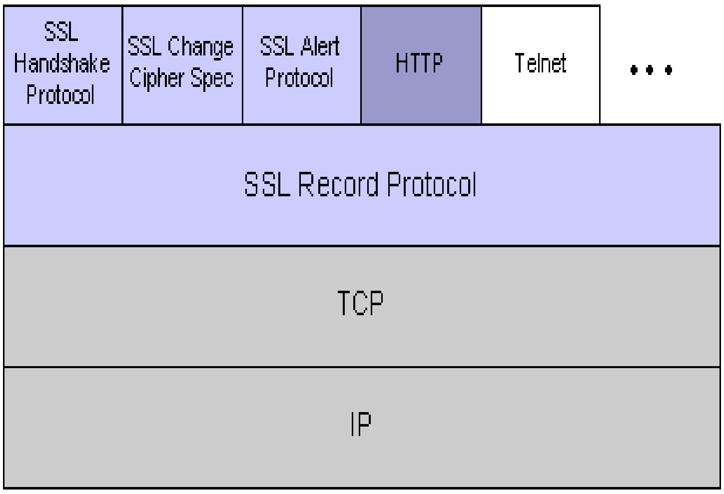


FUENTE: http://www.seguridad%20SSL.html

9 http://www.modssl.org/doc/2.8/ ; Ultimo acceso: Viernes 11 de Julio del 2003

Estos protocolos, al igual que los datos del protocolo de aplicación, se encapsulan dentro del Protocolo de Registro SSL, como se muestra en el gráfico # 6. Un protocolo encapsulado es transferido como dato por la capa de protocolo más baja, la cual no examina el dato.

Gráfico # 6. PROTOCOLO SSL APILADO



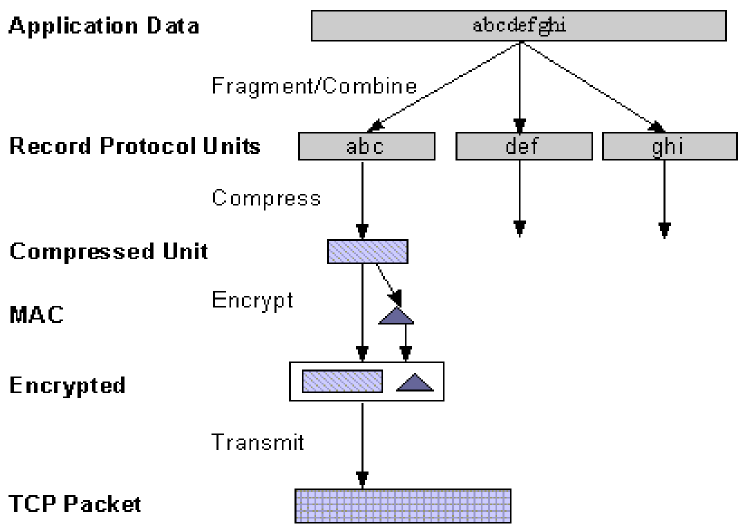
FUENTE: http://www.modssl.org/doc/2.8/

La encapsulación del control de protocolos SSL por el protocolo de registro significa que si una sesión activa es renegociada, los protocolos de control serán transmitidos de forma segura. Si no hubiera sesión anteriormente, se utiliza el juego de cifrado Null, que significa que no hay encriptación y que los mensajes no tienen resúmenes de integridad hasta que se haya establecido la sesión.

1.7.2.5 Transferencia de Datos

El Protocolo de Registro SSL, que se muestra en la figura Nº 6, es utilizado para la transferencia de aplicaciones y el Control de datos SSL entre el cliente y el servidor, la posible fragmentación de estos datos en unidades más pequeñas, o para combinar múltiples mensajes de protocolos superiores en unas unidades simples. Puede comprimir, adjuntar resúmenes de firmas, y encriptar esas unidades antes de transmitirlas, utilizando el protocolo de transporte seguro subyacente. (Nota: actualmente la mayoría de las implementaciones de SSL carecen de soporte para compresión).

Gráfico # 7. PROTOCOLO DE REGISTRO SSL



FUENTE: http://www.modssl.org/doc/2.8/

1.7.2.6 Asegurando las comunicaciones HTTP

Un uso común de SSL es para la seguridad de las Comunicaciones vía web HTTP entre el navegador y el servidor. Este caso no excluye el uso de HTTP no-seguros. La versión segura es principalmente HTTP normal sobre SSL (llamada esta combinación HTTPS), pero con una clara diferencia: se usa el esquema https en lugar de http y también se utiliza un puerto diferente en el servidor (por defecto el 443). Esto es principalmente lo que proporciona mod_ssl para el servidor Apache.[10]

10 http://www.modssl.org/doc/2.8/ ; Ultimo acceso: Viernes 11 de Julio del 2003

CAPITULO II

ARQUITECTURA DEL SERVIDOR WEB SSL

2.1 INTRODUCCIÓN

Los ordenadores personales y los paquetes de software de aplicaciones proliferan comercialmente. Estos ordenadores, también conocidos como estaciones de trabajo programables, están conectados a las Redes de Area Local (LAN), mediante las cuales, los grupos de usuarios y profesionales comparten aplicaciones y datos.

Las nuevas tecnologías de distribución de funciones y datos en una red, permiten desarrollar aplicaciones distribuidas de una manera transparente, de forma que múltiples procesadores de diferentes tipos (ordenadores personales de gama baja, media y alta, estaciones de trabajo, minicomputadoras o incluso mainframes), puedan ejecutar partes distintas de una aplicación. Si las funciones de la aplicación están diseñadas adecuadamente, se pueden mover de un procesador a otro sin modificaciones, y sin necesidad de retocar los programas que las invocan.

En general una aplicación necesita de una serie de datos para funcionar. Estos datos pueden ser obtenidos del usuario a través de ventanas, páginas web, modo texto, de ficheros, bases de datos o de otros programas. Estos datos se procesan y generan un resultado que pueden ser mostrados al usuario, enviados a otro programa, almacenados en bases de datos etc.

Para no tener que programar directamente sobre el hardware entre otras cosas, las aplicaciones se sirven del sistema operativo, pero además existen herramientas que nos permiten olvidarnos de una serie de tareas complejas y que son comunes en muchas aplicaciones, como por ejemplo el acceso a bases de datos, la obtención de datos del usuario, la representación de los resultados, etc.

2.2 MODELO CLIENTE / SERVIDOR

La arquitectura cliente / servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. Es el modelo de interacción más común entre aplicaciones en una red. No forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo.

Los principales componentes del esquema cliente / servidor son: Clientes, Servidores y la infraestructura de comunicaciones. En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios y en el cliente permanece sólo lo particular de cada usuario. [11]

2.2.1 Estructura del cliente

Los Clientes interactúan con el usuario, usualmente en forma gráfica y con frecuencia se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad. Los clientes realizan generalmente funciones como:

- Manejo de la interface del usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos o sentencias a un servidor web de aplicaciones, como son los navegadores: Internet Explorer, Nestcape Navigator, Konkeror, etc.

11 http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/CAP0312.HTM ; Ultimo acceso: Martes 15 de Julio del 2003

2.2.2 Estructura del servidor

Los Servidores proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o externa.
- Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le compila e interpreta. Los clientes suelen situarse en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Como ejemplos de servidores pueden citarse servidores de ventanas como X-windows, servidores de archivos como NFS, servidores para el manejo de bases de datos (como los servidores de SQL), servidores de diseño y manufactura asistidos por computador, servidores web, etc.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas Cliente / servidor actuales, se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración. Entre las principales características de la arquitectura cliente / servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfase única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfase externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente. [12]

2.2.3 Componentes esenciales de la infraestructura cliente / servidor

Una infraestructura Cliente / servidor consta de tres componentes esenciales, todos ellos de igual importancia y estrechamente ligados:

12 http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/CAP0312.HTML ; Ultimo acceso: Martes 15 de Julio del 2003

2.2.3.1 Plataforma operativa

La plataforma deberá soportar todos los modelos de distribución Cliente / servidor, todos los servicios de comunicación, y deberá utilizar preferentemente componentes estándar de la industria para los servicios de distribución.

Los desarrollos propios deben coexistir con las aplicaciones estándar y su integración deberá ser imperceptible para el usuario. Igualmente, podrán acomodarse programas escritos utilizando diferentes tecnologías y herramientas.

2.2.3.2 Entorno de desarrollo de aplicaciones

Debe elegirse después de la plataforma operativa. Aunque es conveniente evitar la proliferación de herramientas de desarrollo, se garantizará que el enlace entre éstas y el middleware no sea excesivamente rígido. Será posible utilizar diferentes herramientas para desarrollar partes de una aplicación. Un entorno de aplicación incremental, debe posibilitar la coexistencia de procesos cliente y servidor desarrollados con distintos lenguajes de programación y/o herramientas, así como utilizar distintas tecnologías (por ejemplo, lenguaje procedural, lenguaje orientado a objetos, multimedia), y que han sido puestas en explotación en distintos momentos del tiempo.

2.2.3.3 Gestión de sistemas

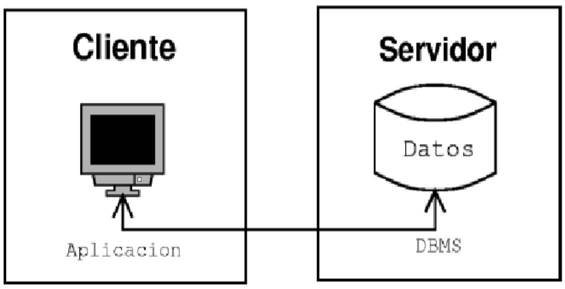
Estas funciones aumentan considerablemente el costo de una solución, pero no se pueden evitar. Siempre deben adaptarse a las necesidades de la organización, y al decidir la plataforma operativa y el entorno de desarrollo.[13]

2.3 ARQUITECTURA DE SOFTWARE DE DOS CAPAS

Las arquitecturas de dos capas consisten de tres componentes distribuidos en dos capas:

1. Cliente.- solicitante de servicios
2. Servidor.- proveedor de servicios

Gráfico # 8. APLICACIONES DE DOS CAPAS



FUENTE: http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/CAP03.HTML

13 http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/CAP0312.HTML ; Ultimo acceso: Martes 15 de Julio del 2003

Los tres componentes son:

1. Interfaz de usuario al sistema.- Tales como una sesión, entradas de texto, desplegado de menús, etc.
2. Administración de procesamiento.- Tales como la ejecución de procesos, el monitoreo de los mismos y servicios de procesamiento de recursos.
3. Administración de bases de datos.- Tales como los servicios de acceso a datos y archivos.

El diseño de dos capas coloca la interfaz de usuario exclusivamente en el cliente y la administración de base de datos en el servidor y divide la administración de procesos entre el cliente y/o el servidor, creando únicamente dos capas.

2.4 ARQUITECTURA DE SOFTWARE DE TRES CAPAS

La arquitectura de software de tres capas surgió para solventar las limitaciones de la arquitectura de dos capas. La tercera capa se localiza entre la interfaz de usuarios (cliente) y el administrador de datos (servidor). Esta capa intermedia provee de servicios para la administración de procesos (tal como el desarrollo, monitoreo y alimentación de procesos) que son compartidos por múltiples aplicaciones.

Gráfico # 9. APLICACIÓN DE TRES CAPAS

FUENTE: http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/CAP03.HTML

El servidor de la capa intermedia, también conocido como servidor de aplicaciones, centraliza la lógica de las aplicaciones, haciendo que la administración de cambios sea más sencilla. Por lo tanto esta arquitectura se divide en tres niveles (capas) que son: presentación, aplicación y datos.

El nivel de presentación es la interfaz que interactúa con el usuario. Ejemplos: modo terminal (solo caracteres), interfaz gráfica (ventanas), una página web (browser), una realidad virtual, etc. El nivel de aplicación son los algoritmos que permiten tratar los datos y hacer lo que realmente deseamos que haga. Ejemplos: Servidores de aplicaciones como: Web Apache Server, Java – tomcat, etc. El nivel de datos se encarga de cargar y guardar los datos que utiliza nuestra aplicación y del formato en el que éstos se mantienen en el sistema de ficheros. Ejemplos: Servidores de bases de datos como: Interbase, Oracle, Sybase, SQL server, etc.

En arquitecturas más simples, cualquier cambio en la lógica, implica rescribir todas las aplicaciones que dependan de ésta. En algunas ocasiones, la capa intermedia se divide en dos o más unidades con diferentes funciones, en estos casos la arquitectura se refiere como multi-capas.

2.5 CLIENTE SERVIDOR EN TCP / IP

Un protocolo es el conjunto de reglas que dos ordenadores conocen y utilizan para comunicarse entre sí. Por tanto, previo a que exista una comunicación entre dos procesos como son un cliente y un servidor, estos deben de estar de acuerdo en el protocolo (algo así como el idioma) que van a utilizar.

Se supone que el cliente es un cliente web y el servidor es un servidor web. El protocolo que ambos procesos deben utilizar será el "hypertext transfer protocol" (http). Si es así, ambos procesos hablan el mismo 'idioma' y podrán comunicarse. El siguiente paso como ya sabemos, es que el cliente inicie la conexión. Para ello previamente el servidor debe estar escuchando. En todo caso, generalmente cuando un proceso servidor esta corriendo, está escuchando desde el primer momento por lo que se llama "un puerto TCP 'bien conocido'".

Ahora se necesita que el proceso cliente sepa 'donde debe llamar a la puerta'. Para ello al menos necesita saber el nombre de la máquina que tiene el proceso servidor, y a qué puerto llamar, ya que en una misma máquina puede haber varios procesos escuchando (y por tanto servidores) por distintos puertos al mismo tiempo.

Con el nombre de la máquina y un servidor DNS, se puede obtener la dirección IP de la máquina que tiene el servidor. Esto se debe hacer ya que un nombre como "http://www.iib.uam.es/" sirve para identificar una máquina, pero las propias máquinas necesitan una dirección IP tipo 150.244.14.6 para poderse identificar, y por tanto comunicarse entre ellas.

Como se ha comentado, el puerto TCP por el que escucha el proceso servidor es un puerto 'bien conocido', y depende del servicio que se preste. Como ejemplos, un servidor web escucha por el puerto 80, un servidor de ficheros escucha por el puerto 21 y un servidor de DNS por el 53.

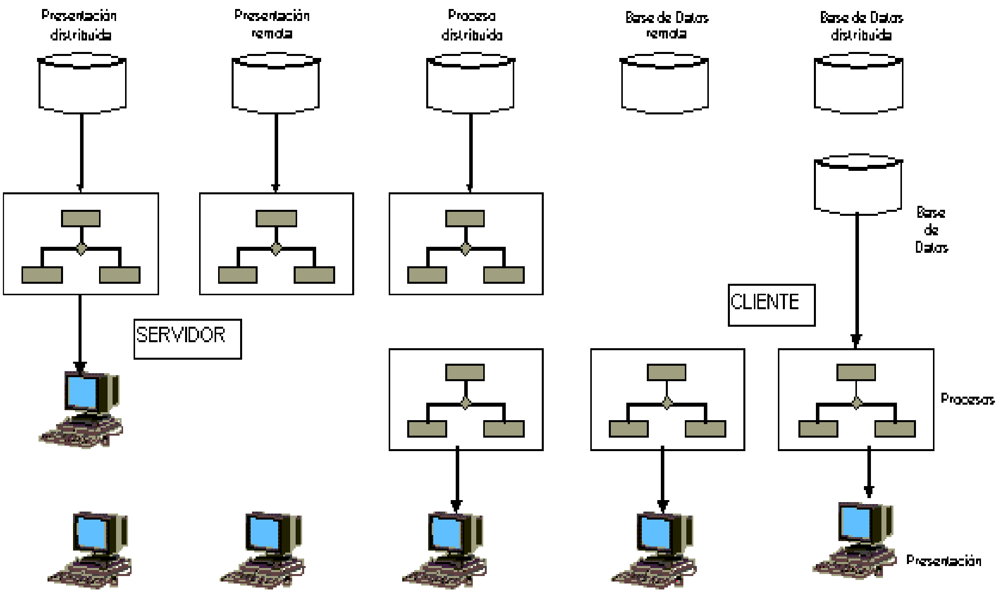
Así que el cliente web inicia una conexión con un servidor web (por tanto con el protocolo http) que está en la dirección http://www.iib.uam.es/ (o que es lo mismo 150.244.14.6) por el puerto 80 que es el habitual de los servicios web.

Cuando al servidor le llega la petición, obtiene en la misma petición la dirección IP del cliente y el puerto que el cliente se ha asignado para iniciar la sesión (el cliente asignará un puerto distinto, no de los más conocidos, a cada conexión que inicie). Si acepta la petición, ya puede existir una conexión entre ambos procesos (cliente y servidor, cada uno con su propio puerto) que se encuentran en distintas máquinas (cada una con su dirección IP).[14]

2.6 CARACTERÍSTICAS FUNCIONALES

Esta arquitectura se puede clasificar en cinco niveles, según las funciones que asumen el cliente y el servidor, tal y como se puede ver en la figura # 10.

Gráfico # 10. CARACTERÍSTICAS FUNCIONALES DEL MODELO CLIENTE / SERVIDOR



FUENTE: http://www2.iib.uam.es/bioinfo/curso/perl/mw/soft_de_base.es.html

14 http://www2.iib.uam.es/bioinfo/curso/perl/mw/soft_de_base.es.html ; Ultimo acceso: Martes 22 de Julio del 2003

2.6.1 Primer nivel

El cliente asume parte de las funciones de presentación de la aplicación, ya que siguen existiendo programas en el servidor, dedicados a esta tarea. Dicha distribución se realiza mediante el uso de productos para el "maquillaje" de las pantallas del mainframe. Esta técnica no exige el cambio en las aplicaciones orientadas a terminales, pero dificulta su mantenimiento. Además, el servidor ejecuta todos los procesos y almacena la totalidad de los datos. En este caso se dice que hay una presentación distribuida o embellecimiento.

2.6.2 Segundo nivel

La aplicación está soportada directamente por el servidor, excepto la presentación que es totalmente remota y reside en el cliente. Los terminales del cliente soportan la captura de datos, incluyendo una validación parcial de los mismos y una presentación de las consultas. En este caso se dice que hay una presentación remota.

2.6.3 Tercer nivel

La lógica de los procesos se divide entre los distintos componentes del cliente y del servidor. El diseñador de la aplicación debe definir los servicios y las interfaces del sistema de información, de forma que los papeles de cliente y servidor sean intercambiables, excepto en el control de los datos, que es responsabilidad exclusiva del servidor. En este tipo de situaciones se dice que hay un proceso distribuido o cooperativo.

2.6.4 Cuarto nivel

El cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada. En esta situación se dice que hay una gestión de datos remota.

2.6.5 Quinto nivel

El reparto de tareas es como en el anterior y además el gestor de base de datos divide sus componentes entre el cliente y el servidor. Las interfaces entre ambos, están dentro de las funciones del gestor de datos y, por lo tanto, no tienen impacto en el desarrollo de las aplicaciones. En este nivel se da lo que se conoce como bases de datos distribuidas.

2.7 CARACTERÍSTICAS FÍSICAS

El diagrama del punto anterior da una idea de la estructura física de conexión entre las distintas partes que componen una arquitectura cliente / servidor. La idea principal consiste en aprovechar la potencia de los ordenadores personales para realizar, sobre todo, los servicios de presentación y según el nivel, algunos procesos o incluso algún acceso a datos locales. De esta forma se descarga al servidor de ciertas tareas para que pueda realizar otras más rápidamente.

También existe una plataforma de servidores que sustituye al ordenador central tradicional y que da servicio a los clientes autorizados. Incluso a veces el antiguo ordenador central se integra en dicha plataforma como un servidor más. Estos servidores suelen estar especializados por funciones (seguridad, cálculo, bases de datos, comunicaciones, etc.), aunque, dependiendo de las dimensiones de la instalación se pueden reunir en un servidor una o varias de estas funciones.

2.8 CARACTERÍSTICAS LÓGICAS

Una de las principales aportaciones de esta arquitectura a los sistemas de información, es la interfaz gráfica de usuario. Gracias a ella se dispone de un manejo más fácil e intuitivo de las aplicaciones mediante el uso de un dispositivo tipo ratón.

En esta arquitectura los datos se presentan, editan y validan en la parte de la aplicación cliente. En cuanto a los datos, cabe señalar que en la arquitectura cliente / servidor se evitan las duplicidades (copias y comparaciones de datos), teniendo siempre una imagen única y correcta de los mismos, disponible en línea para su uso inmediato.

Todo esto tiene como fin que el usuario de un sistema de información soportado por una arquitectura cliente / servidor, trabaje desde su estación de trabajo con distintos datos y aplicaciones, sin importarle dónde están o dónde se ejecuta cada uno de ellos.

2.9 VENTAJAS E INCONVENIENTES

2.9.1 Ventajas

a) Aumento de la productividad:

- Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos.
- Mediante la integración de las aplicaciones cliente / servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes.
- Una interfaz gráfica de usuario consistente, reduce el tiempo de aprendizaje de las aplicaciones.

b) Menores costos de operación:

- Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión.
- Se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- Proporcionan un mejor acceso a los datos.

c) Mejora en el rendimiento de la red:

- Las arquitecturas cliente / servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso.
- La existencia de varias UCPs proporciona una red más fiable: una falla en uno de los equipos, no significa necesariamente que el sistema deje de funcionar.
- En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros, con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema.

2.9.2 Inconvenientes

- Hay una alta complejidad tecnológica al tener que integrar una gran variedad de productos.
- Requiere un fuerte rediseño de todos los elementos involucrados en los sistemas de información (modelos de datos, procesos, interfaces, comunicaciones, almacenamiento de datos, etc.). Además, en la actualidad existen pocas herramientas que ayuden a determinar la mejor forma de dividir las aplicaciones entre la parte cliente y la parte servidor.
- Es más difícil asegurar un elevado grado de seguridad en una red de clientes y servidores que en un sistema con un único ordenador centralizado. Se deben hacer verificaciones en el cliente y en el servidor. También se puede recurrir a otras técnicas como el encriptamiento.
- A veces, los problemas de congestión de la red pueden degradar el rendimiento del sistema por debajo de lo que se obtendría con una única máquina (arquitectura centralizada).
- El quinto nivel de esta arquitectura (bases de datos distribuidas) es técnicamente muy complejo y en la actualidad, hay muy pocas implantaciones que garanticen un funcionamiento totalmente eficiente. [15]

15 http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/CAP0312.HTM ; Ultimo acceso: Martes 15 de Julio del 2003

2.10 INTRODUCCIÓN A LAS APLICACIONES WEB

Las aplicaciones web son una herramienta de Internet que está en constante desarrollo y que se perfilan como el futuro de las aplicaciones convencionales tal y como las conocemos. El avance de las tecnologías ha permitido que Internet se comunique con sus usuarios de modo que cada vez sea mayor la interacción real a través de la Red. Las aplicaciones web es aquel software que interactúa con el usuario a través de la red y se caracterizan por ser sitios web activos que permiten la modificación de sus contenidos y características por sus usuarios. Una aplicación web puede ser meramente una interfaz que muestre y permita la actualización de la base de datos de una empresa.

Los sitios web auto administrables, plataformas de comercio electrónico, portales personalizados para cada cliente en función de sus gustos, son pequeños ejemplos de lo que se puede ver en la Red. El avance de las plataformas y de las tecnologías también se dirige en esa dirección.

Para que se pueda considerar un sitio web como aplicación web es necesario que exista un concepto de dinamismo, de no existir se limitará a considerar un repositorio de datos o sitio web estático. Una de las controversias generadas acerca de la valoración de las aplicaciones web viene suscitada por el entorno de ejecución de las mismas.

Hay quien defiende que las aplicaciones web deben modificar el estado del servidor que las proporciona entendiendo por aplicación web “aquella aplicación accesible mediante Internet, que modifica el estado del servidor que la aloja”; esto dejaría fuera del grupo de las aplicaciones web aquellas realizadas en script de cliente e incluso algunos applets.

Otras corrientes defienden que es aplicación web toda aquella que ofrezca una interacción con el usuario siendo esta proporcionada por un servidor web, es decir, aunque la ejecución se realice en la máquina cliente, es el servidor el que proporciona esa información.

Una aplicación web, posee un servidor de Internet (ISP) y un cliente ambos se comunican a través de la red. El servidor puede proporcionar al cliente páginas web planas, o lo que es lo mismo código HTML. Esto será siempre igual independientemente del tipo de usuario y de la petición del mismo.

Con las nuevas tecnologías se puede lograr que el servidor interprete la petición y el contexto (usuario, zona geográfica, etc.) en el que se realiza la misma, generando código a la medida de la petición mediante procesos, interactuando con otro servidor, consultando una base de datos, etc. E incluso puede que el cliente esté preparado para procesar la información que le transmita el servidor en función de los deseos del usuario.

2.10.1 Entorno de implementación

Una aplicación web puede ser programada de modo que se ejecute en el entorno del servidor o en el entorno del cliente. De ser implementada en el entorno del cliente se encuentra con la ventaja de:

- La carga computacional reside en los clientes por lo que se rebaja de la máquina servidora.

Por otra parte, se tiene como desventajas:

- El tiempo de descarga será mayor.
- Se desconoce las características del equipo cliente.
- El código terminará residiendo en la máquina de un cliente y en algunos casos podrá ser modificado con fines destructivos.
- Los recursos del equipo cliente pueden requerir de descargue de información extra (plug-ins), lo que hará que se tarde más o que incluso el cliente desista de visitar el sitio.

En el caso de que se realice la implementación en el servidor las ventajas serán las siguientes:

- En un equipo, se tiene un conocimiento del mismo y se puede instalar lo que se crea conveniente, se puede limitar a enviar exclusivamente los datos requeridos, lo que hará más rápido el tiempo de descarga.
- El código en este caso estará en la máquina salvo de posibles modificaciones e intrusiones, siempre que esté seguro de que este servidor está a salvo de ataques.

Como desventaja principal se puede citar:

- La máquina que realiza la carga computacional puede aumentar mucho más que en el caso de encontrarse con una implementación en el cliente.[16]

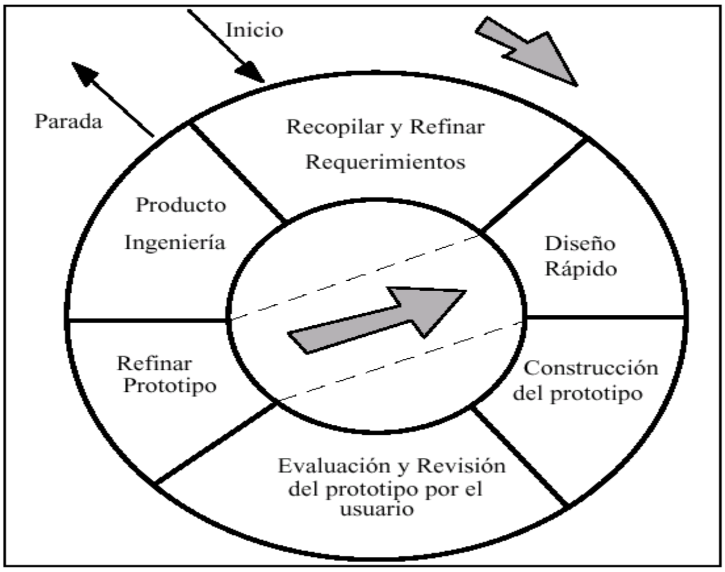
2.11 PARADIGMA CONSTRUCCIÓN DE PROTOTIPOS

La construcción de prototipos es un proceso que facilita al programador la creación de un método de software a conseguir. El método tomará una de las 3 formas siguientes: Un prototipo en papel que describa la interrelación hombre-maquina de forma que facilita al usuario la comprensión como producirá tal interrelación; un prototipo que funcione e implemente algunos subconjuntos de la función requerida al software deseado o un programa existente que ejecute parte o toda la función deseada pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

16 http://www.zope.org ; Ultimo acceso: Viernes 8 de Agosto del 2003

La secuencia de sucesos para el paradigma de construcción de prototipos se muestra en el gráfico # 11

Gráfico # 11. CONSTRUCCIÓN DE PROTOTIPOS



FUENTE: http://www.paradigmadeconstrucciondeprototipos.html

Como todos los métodos de desarrollo de software la construcción de prototipos comienza con la recolección de los requerimientos del sistema. El técnico y el cliente se reúnen y definen los objetivos globales para el software, identifican todos los requerimientos conocidos y perfilan las áreas donde será necesario una mayor definición. Luego se produce un diseño rápido. El diseño rápido se enfoca sobre la representación de los aspectos del software visibles al usuario por ejemplo: métodos de entrada y formatos de salida. El diseño rápido conduce a la construcción de un prototipo.

El prototipo es evaluado por el cliente-usuario y se utiliza para refinar los requerimientos del software a desarrollar. Se produce un proceso interactivo en el que prototipo es afinado para que satisfaga las necesidades del cliente al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de todo lo que hay que hacer idealmente, el prototipo sirve como un mecanismo para identificar los requerimientos del software.

2.11.1 Características

- Este paradigma ayuda al cliente a brindar requisitos paso a paso.
- También facilita al programador ir probando algoritmos no explotados con anterioridad, de los que no tiene seguridad de su eficiencia.
- Consiste en la creación de prototipos

2.11.2 Ventajas

- Son reales y tangibles.
- Permite al cliente aclarar lo que quiere que haga el sistema.

- Siente que es oído y tenido en cuenta para el diseño.
- Asegura que el trabajo se está haciendo bien y cumpliendo los requerimientos del cliente.

2.11.3 Desventajas

- El cliente puede creer que el sistema ya está listo y pedir su entrega rápida.
- Crea expectativas más allá de lo que realmente puede hacer.
- Se dificulta la dirección y control del proceso de desarrollo más que en el método clásico.
- La presión por entregar rápido el producto compromete la calidad.
- Se dificulta mantener el entusiasmo del cliente después de aprobado el prototipo porque creerá que se desperdicia el tiempo en detalles insignificantes.

2.12 COMERCIO ELECTRÓNICO

Hoy en día, gran parte de la actividad comercial ha podido transformarse gracias a redes de conexión por computadoras como Internet, esta transformación facilita hacer transacciones en cualquier momento, de cualquier lugar del mundo. Todo lo que esta alrededor de esta nueva forma de hacer negocios es lo que se ha llamado comercio electrónico, sin duda la gran variedad de actividades que giraban alrededor del quehacer comercial se ha tenido que juntar con las nuevas técnicas cibernéticas. Así hoy tanto un comerciante, un banquero, un abogado o una matemático puede hablar de comercio electrónico enfocándose a la parte que le corresponde.

Existen diferentes niveles de hacer comercio electrónico, y su clasificación aún esta por formarse, sin embargo, la parte más visible es la que cualquier usuario en una computadora personal puede ver, esto es hacer comercio electrónico se convierte a comprar o vender usando una conexión por Internet en lugar de ir a la tienda.

La forma de hacer esto es muy similar a lo que tradicionalmente se hace, por ejemplo: en la tienda uno entra al establecimiento, de forma electrónica se prende la computadora y una vez conectado a Internet entra a la página del negocio, enseguida un comprador revisa los productos que posiblemente compre y los coloca en una carrito, de la misma forma en la computadora se navega por la página del negocio y con el browser se revisa los productos que éste vende, al escoger éstos se colocan en un carrito virtual, que no es nada mas que un archivo del usuario.

Una vez elegidos los productos de compra se pasa a la caja, donde se selecciona un sistema de pago y se facturan los productos al comprador. De forma similar en la computadora se pueden borrar productos que no se quieren comprar o añadir nuevos, una vez elegidos éstos se procede a una parte de la página que toma los datos y solicita el método de pago, generalmente se lleva a cabo con tarjeta de crédito.

En la parte tradicional de comprar al pagar en la caja termina el proceso, en la parte por computadora aún tiene que esperarse que sean enviados los productos. A pesar de esto las ventajas que ofrece el comercio electrónico son magníficas, ya que es posible comprar en un relativo corto tiempo una gran cantidad de productos sin necesidad de moverse de lugar, es decir al mismo tiempo se puede comprar una computadora, un libro, un regalo, una pizza, hacer una transacción bancaria etc., de la forma tradicional se llevaría al menos un día completo y eso si los negocios esta en la misma ciudad, si no, el ahorro de tiempo que representa comprar por Internet es incalculable.

Al efectuar una operación comercial por Internet se presentan nuevos problemas, por ejemplo cómo saber que la tienda virtual existe verdaderamente, una vez hecho el pedido cómo saber que no se cambia la información, cuando se envía el número de tarjeta de crédito cómo saber si este permanecerá privado, en fin, para el comerciante también se presentan problemas similares, cómo saber que el cliente es honesto y no envía información falsa, etc. Todos estos problemas pueden ser resueltos de manera satisfactoria si se implementan protocolos de comunicación segura usando criptografía.[17]

17 http://www.paradigmadeconstrucciondeprototipos.html ; Ultimo acceso: Viernes 8 de Agosto del 2003

CAPITULO III

IMPLEMENTACIÓN DEL SERVIDOR WEB SSL

3.1 SISTEMA OPERATIVO RED HAT LINUX

Linux es una versión de UNIX de libre distribución, inicialmente desarrollada por Linux Torvals en la Universidad de Helsinki, en Finlandia. Fue desarrollado con la ayuda de programadores y expertos de Unix a lo largo y ancho del mundo, gracias a la presencia de Internet. Linux no es un sistema operativo que se actualiza cada dos o tres años, por el contrario es un sistema dinámico que se encuentra en constante actualización y se ha convertido en el sistema operativo para los negocios, educación, y provecho personal.

Cualquier habitante del mundo puede acceder a Linux y desarrollar nuevos módulos o cambiarlo ya que es totalmente gratuito, no necesita comprar licencias para poder trabajarlo, simplemente la forma más fácil de conseguir es “bajarlo de Internet”.

Linux es un clónico del sistema operativo UNIX que corre en ordenadores Intel 80386 y 80486. Soporta un amplio rango de software, desde TEX a X Windows al compilador GNU C/C++ a TCP/IP. Es una implementación de UNIX versátil, distribuida gratuitamente en los términos de la Licencia GNU.

3.1.1 Características generales

- Es un sistema operativo multitarea y multiusuario, similar a Unix en muchos aspectos pero completamente independiente de este.
- Multi-procesador desde la versión 2.0 del núcleo, Linux soporta múltiples procesadores, distribuyendo las tareas en distintos procesadores.
- Se distribuye bajo la licencia GPL de GNU, lo cual le ha permitido desarrollarse y distribuirse con rapidez.
- Es portado para numerosas plataformas computacionales. Ejemplos: Intel, Macintosh, Alpha y SPARC.
- Linux está especialmente orientado al trabajo en redes pues existen numerosas aplicaciones, nativas o portadas para él, que implementan casi todos los protocolos utilizados para la comunicación, tanto el lado cliente como el servidor. Ejemplos: FTP, HTTP, TCP/IP, PPP, UUCP, SMTP, SNMP, gopher, wais, news, IPX, SMB/CIFS, POP, IMAP, etc. Gracias a esto una máquina Linux puede proveer servicios de correo electrónico, resolución de nombres, news, Web, acceso remoto, compartir recursos a través de la red, etc.
- Las versiones actuales del kernel soportan numerosos sistemas de ficheros como FAT16/32, NTFS e iso9660 (para discos compactos). Posee un file system propio denominado ext2.

- Ofrece numerosas posibilidades para la programación en diversos lenguajes, tanto interpretados como compilados. Ejemplos: C, C++, Java, Pascal, PHP, SmallTalk, FORTRAN, LISP, Perl, Tcl/Tk y Python. También en Linux se dispone de varios shells como bash, csh, ksh y otros.
- Sobre Linux se implementan diversos ambientes conocidos como interfaces X que son interfaces gráficas entre ellas se destacan KDE, GNOME y WindowMaker..
- Las máquinas Linux se pueden interconectar y relacionar fácilmente con otras y con sistemas operativos diferentes como OS/2, Apple Machintosh, Windows 9x y NT.
- No requiere necesariamente de grandes recursos de hardware. Sin interfaz gráfica puede ejecutarse correctamente en máquinas con 16 MB de RAM o menos, contando además con espacio Swap1.14 en el disco duro. [18]

3.2 SERVIDOR WEB APACHE

Hoy en día el servidor Apache es el número uno indiscutible del mercado. Esto se debe a que es gratuito, su excelente rendimiento y su gran flexibilidad. **Apache** debe su nombre a su origen: consiste en una versión parcheada (**A**Pa**tC**Hy **s**Erver) del servidor de la NCSA.

18 http://www.linux.cu/manual/basico-html/footnode.html ; Ultimo acceso: Lunes 4 de Agosto del 2003

Este servidor es el producto más ampliamente conocido del proyecto Apache que trata de construir un servidor http robusto, potente, disponible en todas las plataformas, gratuito y con disponibilidad de su código fuente.

Se diseñó originalmente para correr en máquinas Unix y sus múltiples variantes; sin embargo, hoy en día está disponible para la práctica total de sistemas operativos. El servidor web Apache es uno de los mayores triunfos del software libre.

3.2.1 Características generales

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características se destacan:

- Multiplataforma
- Es un servidor de web conforme al protocolo HTTP/1.1
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.

- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

3.2.2 Requisitos

Los siguientes requisitos son necesarios para la compilación e instalación de Apache:

- Espacio en disco. Aproximadamente se debe tener un disco con 12MB de espacio libre disponible para realizar la compilación. Una vez realizada la instalación, Apache ocupa alrededor de 3MB.
- Compilador ANSI-C. Se debe estar seguro que se tiene un compilador ANSI-C instalado. Se recomienda el compilador GCC y G++. Si no se tiene el compilador, se puede obtener de la siguiente dirección: http://www.gnu.org/.
- Perl 5 (opcional). Es necesario tener instalado para dar soporte a algunos scripts como apxs o dbmanage.[19]

19 http://barrapunto.com ; Ultimo acceso: Lunes 4 de Agosto del 2003

3.3 INSTALACIÓN DEL SERVIDOR WEB APACHE CON SEGURIDADES

Para la instalación de este servidor es necesario tener instalado en el computador Red Hat Linux 7.2 como sistema Operativo e Interbase 6.0 como servidor de bases de datos con la respectiva licencia proporcionada por PETROECUADOR. Además hay que bajarse de Internet los paquetes que se muestran en la Tabla # 2 que son completamente gratuitos y se debe ubicarlos en un directorio específico. Para crear un directorio se debe ubicar en el directorio raíz que se simboliza con el signo / y digitar el siguiente comando:

mkdir /ssltesis.

Tabla # 2. SOFTWARE USADO EN LA INSTALACIÓN DEL SERVIDOR WEB SSL

PAQUETES	**DISTRIBUCIÓN**
Red Hat Linux 7.2	CD's PETROECUADOR
Interbase 6.0	CD PETROECUADOR con licencia
Apache_1.3.27.tar.gz	http://www.apache.org/
Mod_ssl-2.8.14-1.3.27.tar.gz	http://www.modssl.org/
Openssl-0.9.7a.tar.gz	http://www.openssl.org/
mm-1.1.x.tar.gz	http://www.engelschall.com/sw/mm/
Gzip-1.2.4.tar.gz	http://www.gnu.org/
perl-5.6.0.tar.gz	http://www.perl.com/
Php-4.3.1.tar.gz	http://www.php.net/

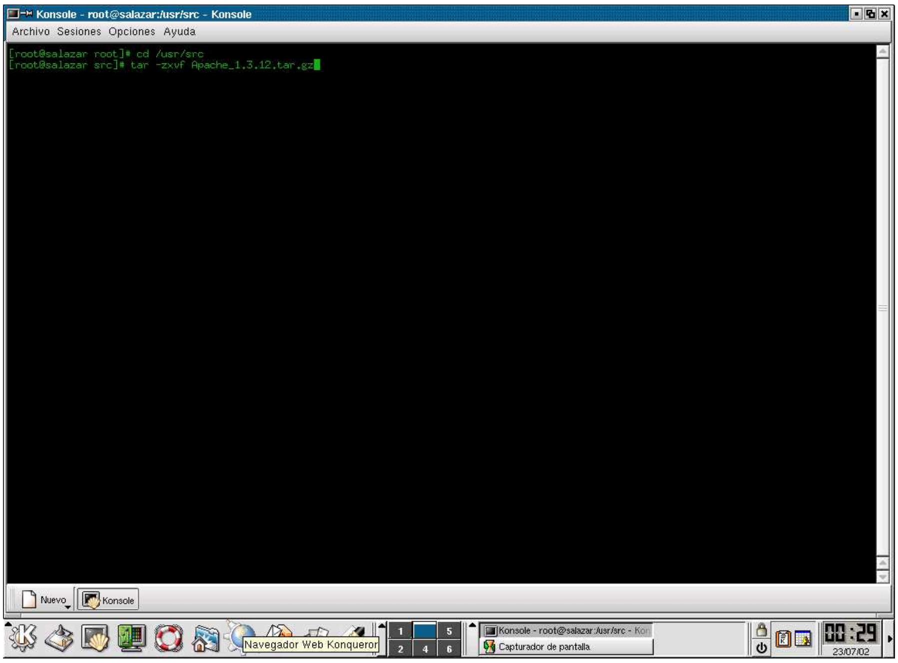
FUENTE: http://www.cambuddys.com/phpssldso.html

3.3.1 Intalación del Interbase 6.0

La instalación del Interbase 6.0 es un servidor de bases de datos el cual se instala primero para después configurarlo conjuntamente con los demás paquetes. También se puede instalar en otro servidor cuando las bases son demasiado grandes y necesitan de espacio suficiente en disco. Para la instalación se puede escoger los paquetes RPM o TAR con las licencias adquiridas por PETROECUADOR, en este caso se procede a instalar mediante paquetes RPM.

1. Ingresar al sistema Linux como usuario "root" e iniciar una consola de trabajo.

Gráfico # 12. CONSOLA DE TRABAJO



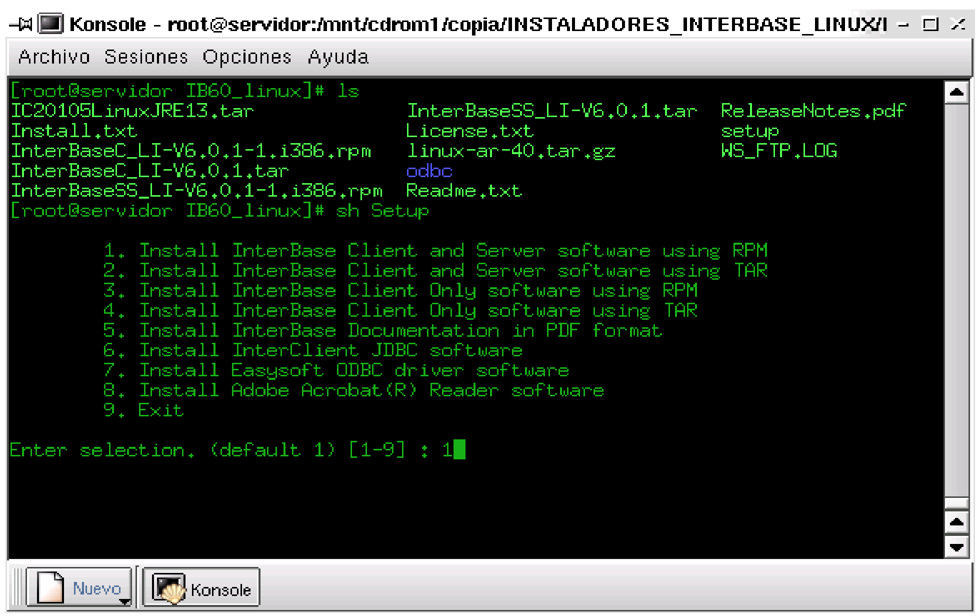
FUENTE: La autora

2. Verificar si existe instalado alguna versión anterior de Interbase con el siguiente comando: rpm -q -a | grep Interbase

3. Si al realizar la ejecución del comando anterior no se visualizó ningún resultado, continuar con el paso 5 y si por el contrario se desplegó una lista con la versión del Interbase instalada, se procede a desinstalar escribiendo el siguiente comando: rpm -e Interbase_5.6

4. Reiniciar el equipo e insertar el CD en la unidad de CD_ROOM y realizar un montaje del dispositivo, digitando el siguiente comando: mount /mnt/cdrom

Gráfico # 13. OPCIONES PARA INSTALAR EL INTERBASE 6.0



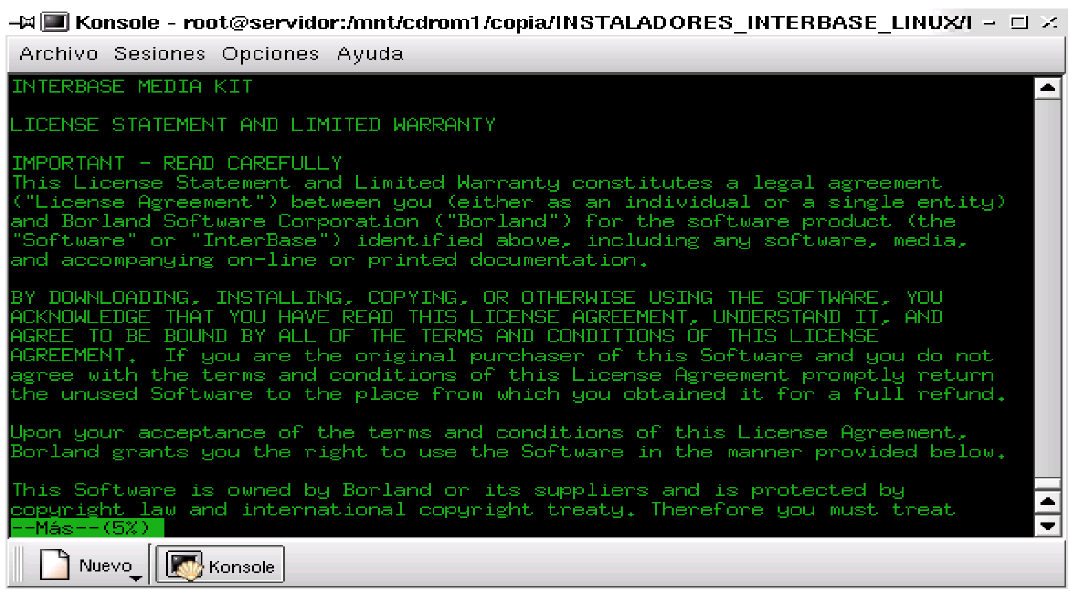
FUENTE: La autora

5. Hay que escoger la opción 1 del menú que se despliega en pantalla y aceptar los términos propuestos para el manejo del Interbase e inclusive escoger el path por default en donde se encontrará instalado el Interbase que es: /opt/interbase/bin.

   6.1 Para salir de la instalación una vez que se ha aceptado todos los requerimientos del menú desplegado escoger la opción número 9.

6. Ubicarse en el directorio /opt/interbase/bin para aceptar el contrato e ingresar las licencias de Interbase, digitando el siguiente comando: ./iblicence

Gráfico # 14. CONTRATO DE LA LICENCIA DE INTERBASE 6.0



FUENTE: La autora

7. Ingresar la licencia de Interbase utilizando una serie de opciones que se detallan a continuación:

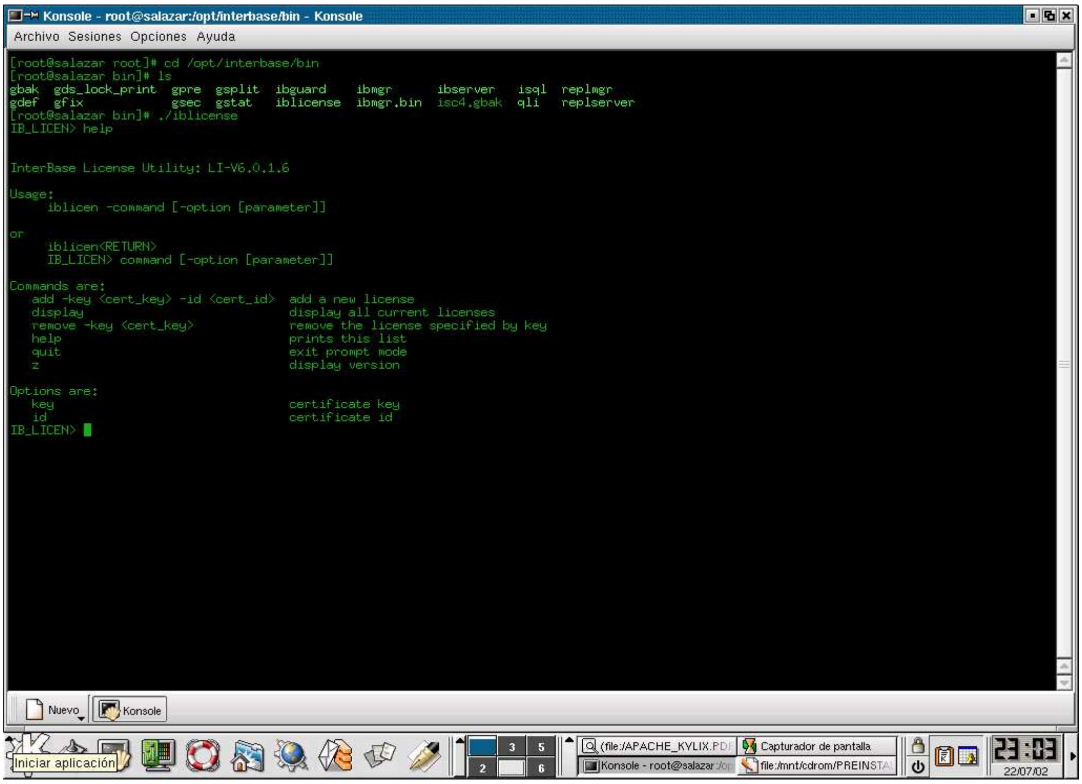
   Help ....Permite ver todas las opciones de manejo

   Display..Permite obtener un listado de las licencias.

   Quit...Permite salir del archivo

   Add –key <clave> -id <clave>......Permite ingresar una nueva licencia . Se puede ingresar licencias en evaluación (eval).

Gráfico # 15. INGRESO DE LICENCIAS DEL INTERBASE 6.0



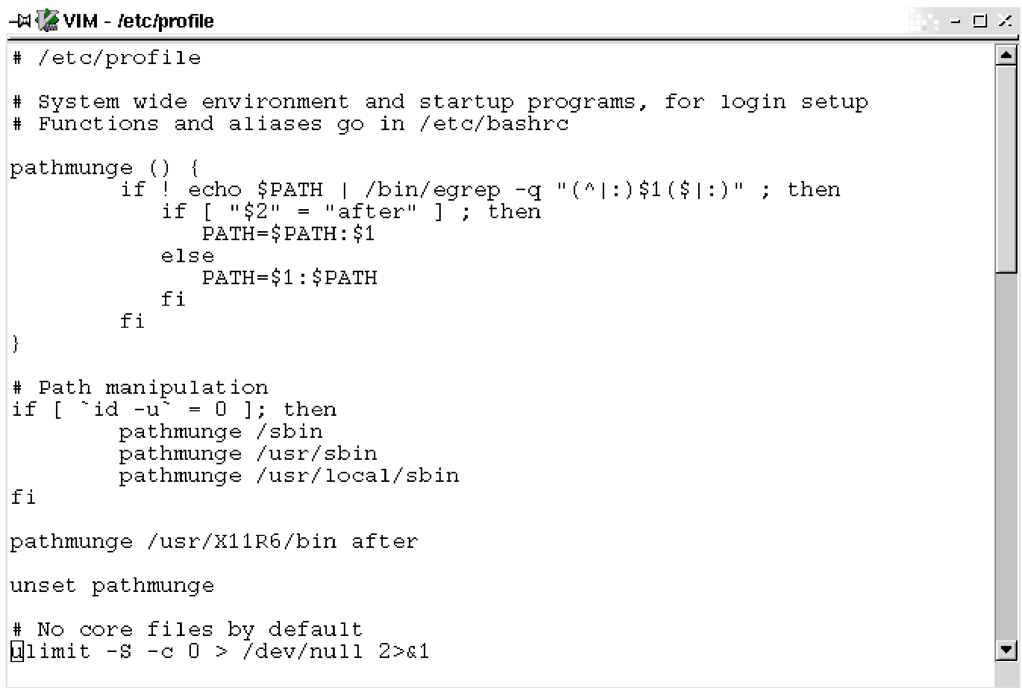
FUENTE: La autora

8. Ubicarse en la carpeta /etc y dentro de esta crear un archivo denominado profile luego digitar el siguiente código:

   PATH =$PATH :/opt/interbase/bin

   export PATH

Gráfico # 16. ARCHIVO PROFILE



VIM - /etc/profile

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

pathmunge () {
        if ! echo $PATH | /bin/egrep -q "(^|:)$1($|:)" ; then
           if [ "$2" = "after" ] ; then
              PATH=$PATH:$1
           else
              PATH=$1:$PATH
           fi
        fi
}

# Path manipulation
if [ `id -u` = 0 ]; then
        pathmunge /sbin
        pathmunge /usr/sbin
        pathmunge /usr/local/sbin
fi

pathmunge /usr/X11R6/bin after

unset pathmunge

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1
```

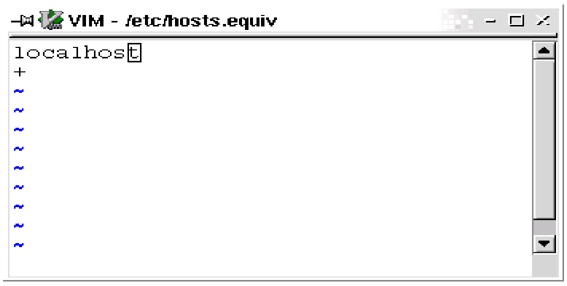
FUENTE: La autora

9. Después ubicarse en el directorio /etc/Hosts.equiv, pulsar la tecla F4 y verificar que exista el siguiente código:

   localhost

   +

Gráfico # 17. ARCHIVO HOSTS.EQUIV

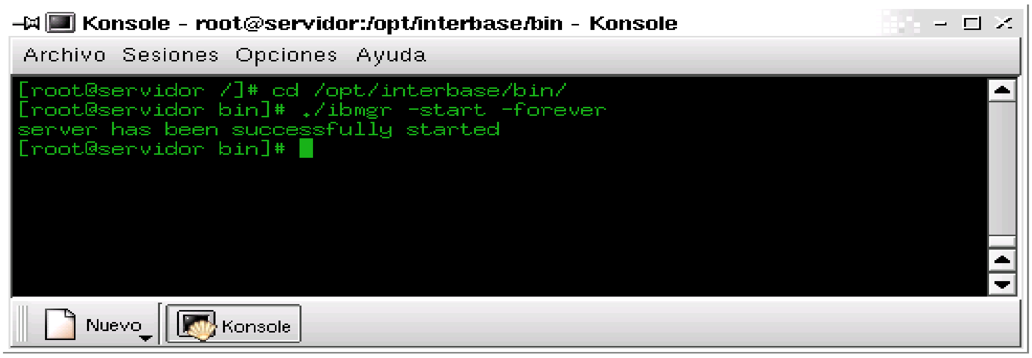


FUENTE: La autora

10. Ahora será necesario copiar una librería denominada libncurses4.0.so del CD de Interbase en el directorio /usr/lib y reiniciar el equipo.

11. Una vez que se ubique en el directorio /op/interbase/bin ingresar el siguiente código para iniciar el servidor: ./ibmgr –start –forever

Gráfico # 18. INICIO DEL SERVIDOR DE INTERBASE 6.0



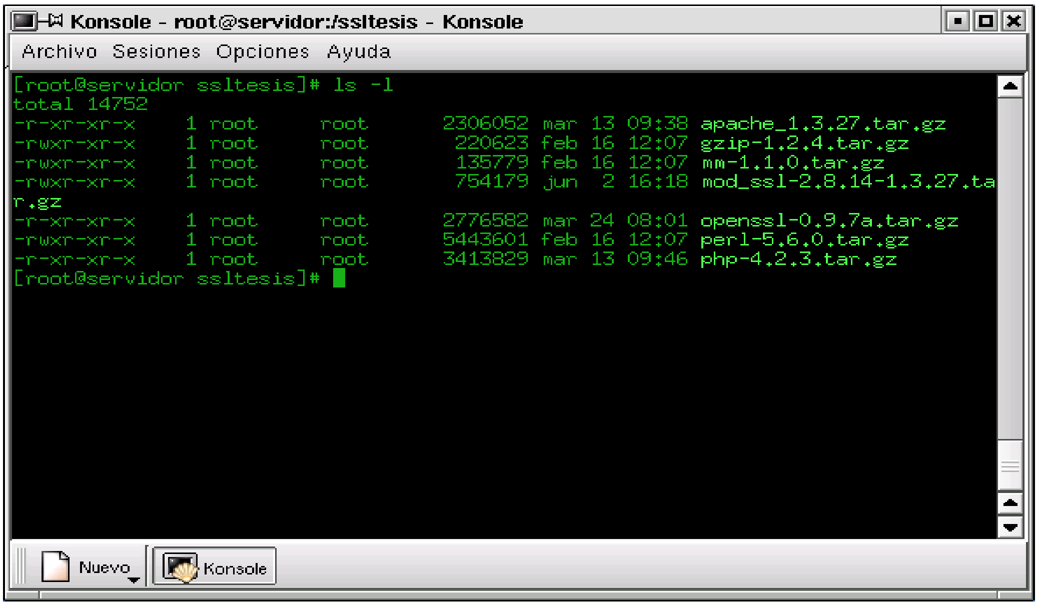
FUENTE: La autora

Finalmente el servidor de bases de datos se encuentra instalado y funcionando.

3.3.2 Instalación del servidor SSL

Para empezar con la instalación se debe ingresar al sistema Operativo Linux, tener una consola(ventana) con los paquetes como se muestra en el gráfico # 19.

Gráfico # 19. CONSOLA DE TRABAJO



FUENTE: La autora

1. Ubicarse en el directorio /ssltesis para descomprimir cada uno de los paquetes con el comando tar. Al ejecutar este comando se crearán directorios individuales con sus respectivos nombres identificativos como se muestra a continuación:

   tar -zxvf gzip-1.2.4.tar.gz

   tar -zxvf perl-5.6.0.tar.gz

tar -zxvf apache_1.3.27.tar.gz

tar -zxvf mod_ssl-2.8.12-1.3.27.tar.gz

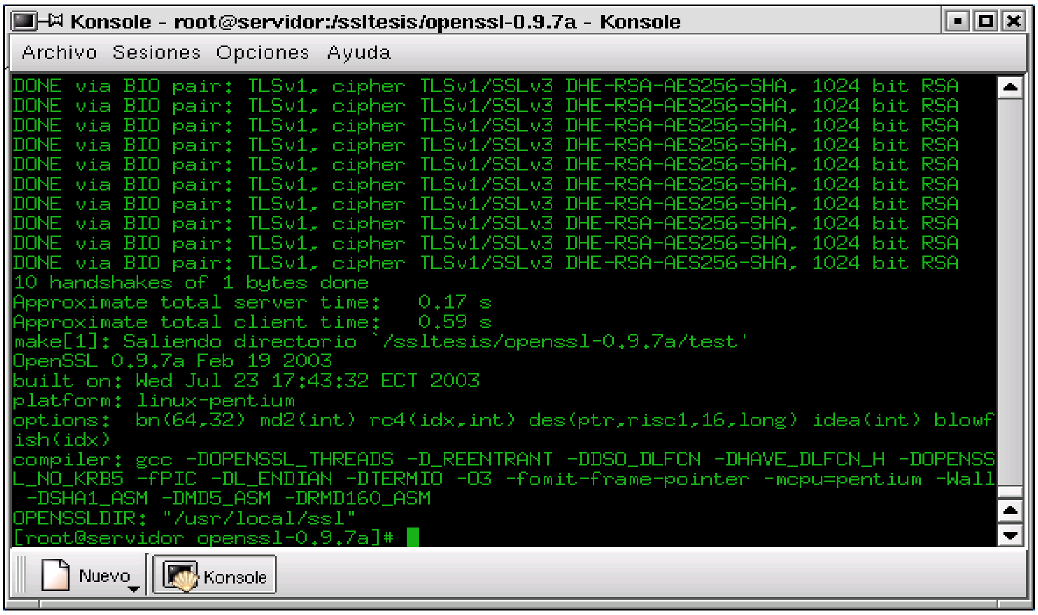
tar -zxvf openssl-0.9.7a.tar.gz

NOTA: Para ingresar a los directorios usar el comando cd y su nombre, para salir de éstos digitar el comando cd ..

2. Ingresar al directorio openssl-0.9.7a para instalar y configurar el paquete OpenSSL digitando el siguiente código:

   sh config -fPIC

Gráfico # 20. INSTALACION DEL PAQUETE OPENSSL



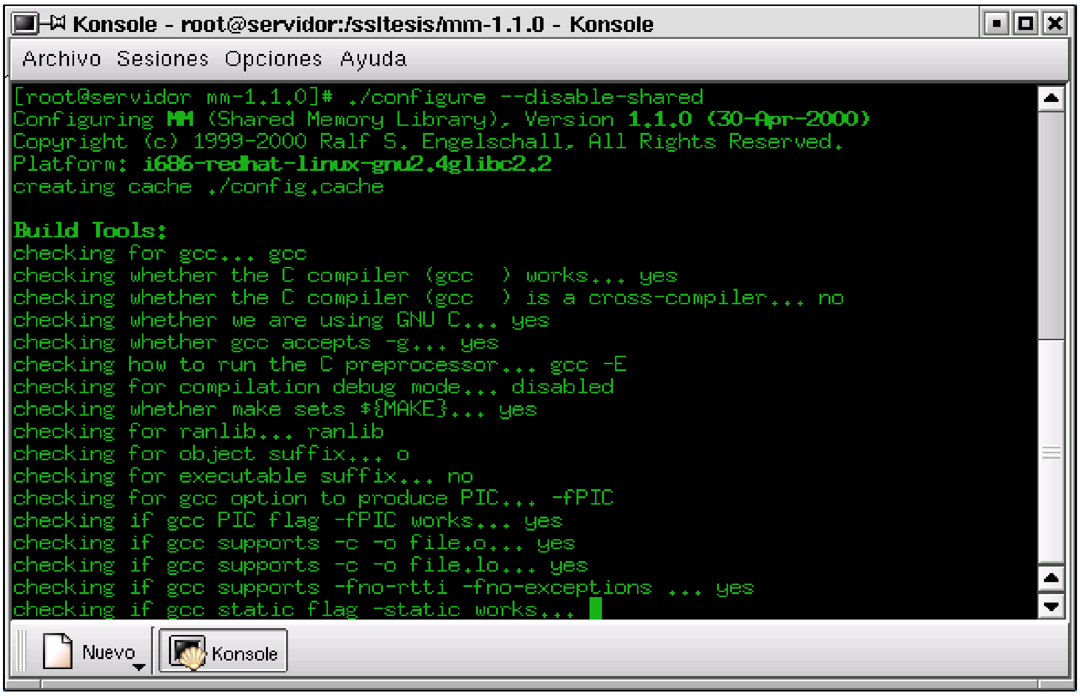
FUENTE: La autora

Para finalizar la instalación de éste paquete se debe digitar los comandos: make y luego make test.

3. Ubicarse en el directorio mm-1.1.x para instalar y configurar el paquete MM Shared Memory digitando el siguiente código:

   ./configure –disable-shared

Gráfico # 21. INSTALACION DEL PAQUETE MM SHARED MEMORY



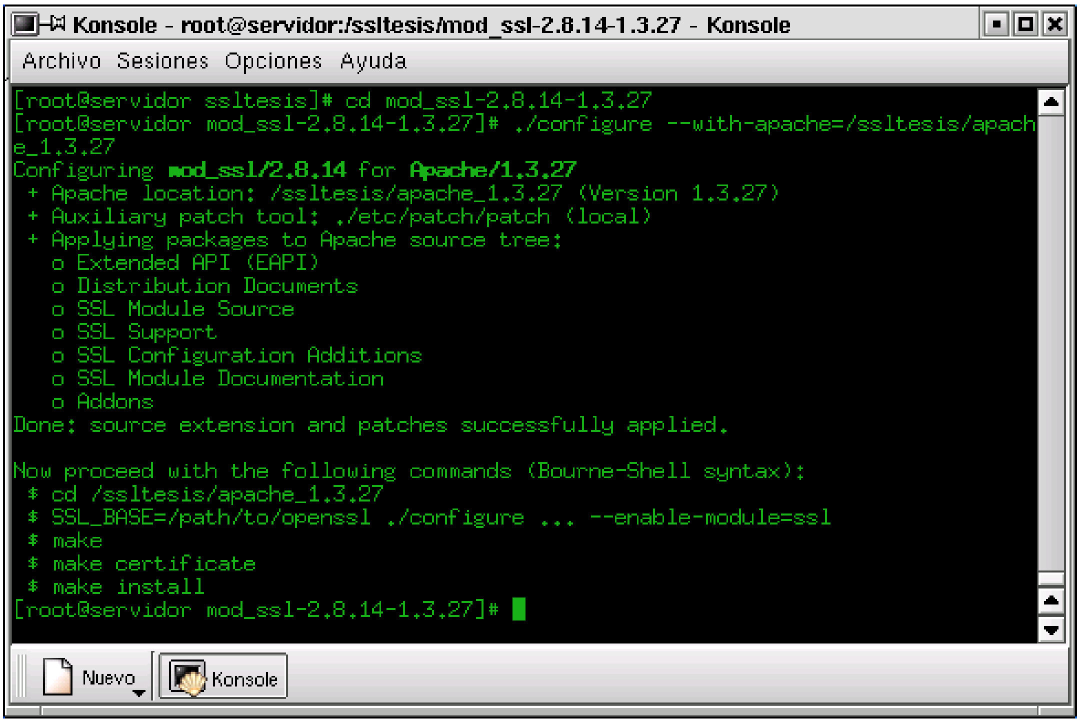
FUENTE: La autora

Posteriomente digitar el comando make para finalizar esta instalación.

4. Ingresar al directorio mod_ssl-2.8.12-1.3.27 para instalar y configurar el paquete MOD_SSL digitando el siguiente código:

./configure --with-apache=/apache_1.3.27 --with-crt=/usr/local/apache/conf/ ssl.crt/ server.crt --with-key=/usr/local/apache/conf/ssl.key/server.key

Gráfico # 22. INSTALACION DEL PAQUETE MOD_SSL

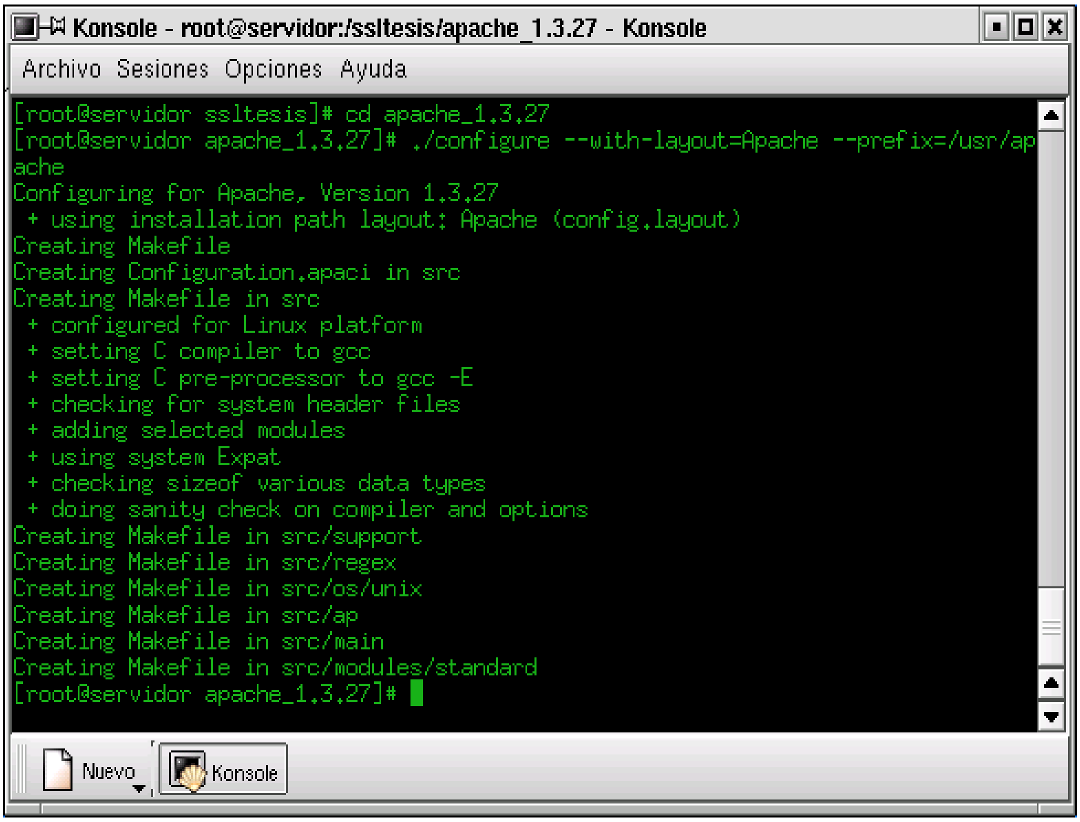


FUENTE: La autora

5. Ubicarse en el directorio apache_1.3.27 para instalar y configurar el apache mediante el siguiente código:

./configure --with-layout=Apache --prefix=/usr/apache

Gráfico # 23. INSTALACION DEL PAQUETE APACHE

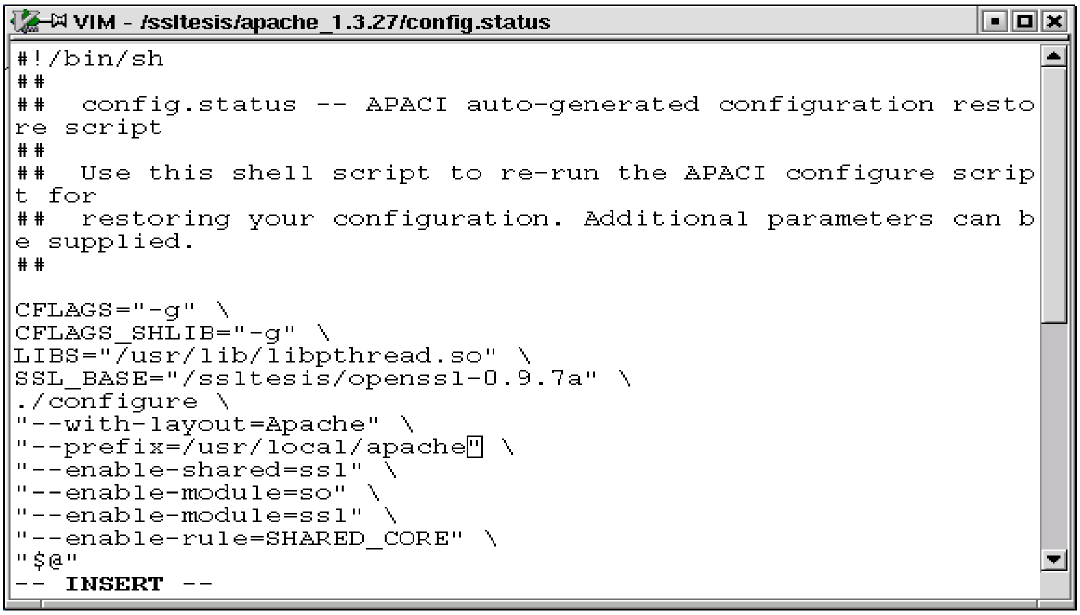


FUENTE: La autora

Luego digitar el comando make y después make install para finalizar la instalación.

6. Ingresar al archivo config.status con el comando gvim para recompilar el Apache escribiendo en éste archivo el código que se muestra en el gráfico # 24.

Gráfico # 24. RECOMPILACION DEL PAQUETE APACHE



```
VIM - /ssltesis/apache_1.3.27/config.status
#!/bin/sh
##
##  config.status -- APACI auto-generated configuration resto
re script
##
##  Use this shell script to re-run the APACI configure scrip
t for
##  restoring your configuration. Additional parameters can b
e supplied.
##

CFLAGS="-g" \
CFLAGS_SHLIB="-g" \
LIBS="/usr/lib/libpthread.so" \
SSL_BASE="/ssltesis/openssl-0.9.7a" \
./configure \
"--with-layout=Apache" \
"--prefix=/usr/local/apache" \
"--enable-shared=ssl" \
"--enable-module=so" \
"--enable-module=ssl" \
"--enable-rule=SHARED_CORE" \
"$@"
-- INSERT --
```

FUENTE: La autora

Luego se debe presionar la tecla ESC para grabar y salir de este archivo con los siguientes comandos: :wq!

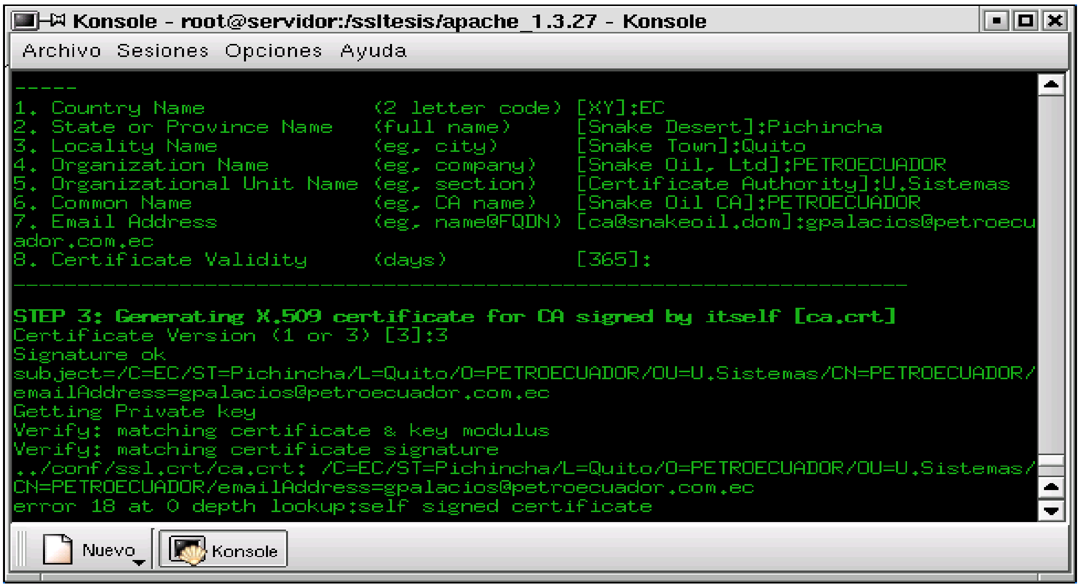
7. Después de haber grabado digitar el siguiente código:

   chmod 655 config.status

   ./config.status

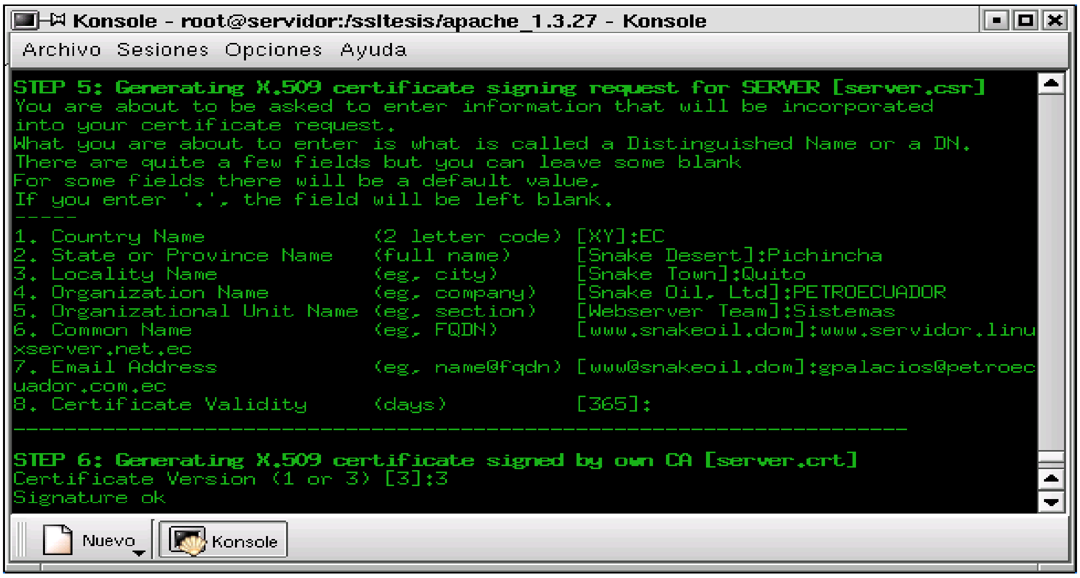
   Finalmente digitar el comando make y luego make certificate TYPE=custom para crear el certificado, ingresando los datos correspondientes tanto para la Autoridad Certificadora como para el servidor.

Gráfico # 25. DATOS PARA LA AUTORIDAD CERTIFICADORA



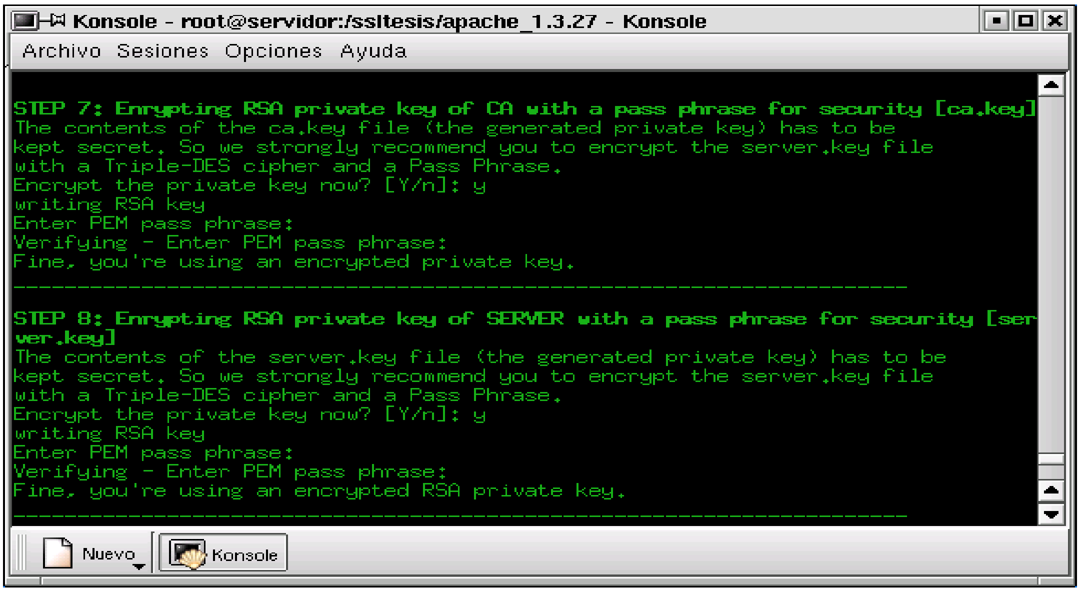
FUENTE: La autora

Gráfico # 26. DATOS PARA EL SERVIDOR



FUENTE: La autora

Gráfico # 27. INGRESO DE CLAVES PARA LA AUTORIDAD CERTIFICADORA Y PARA EL SERVIDOR



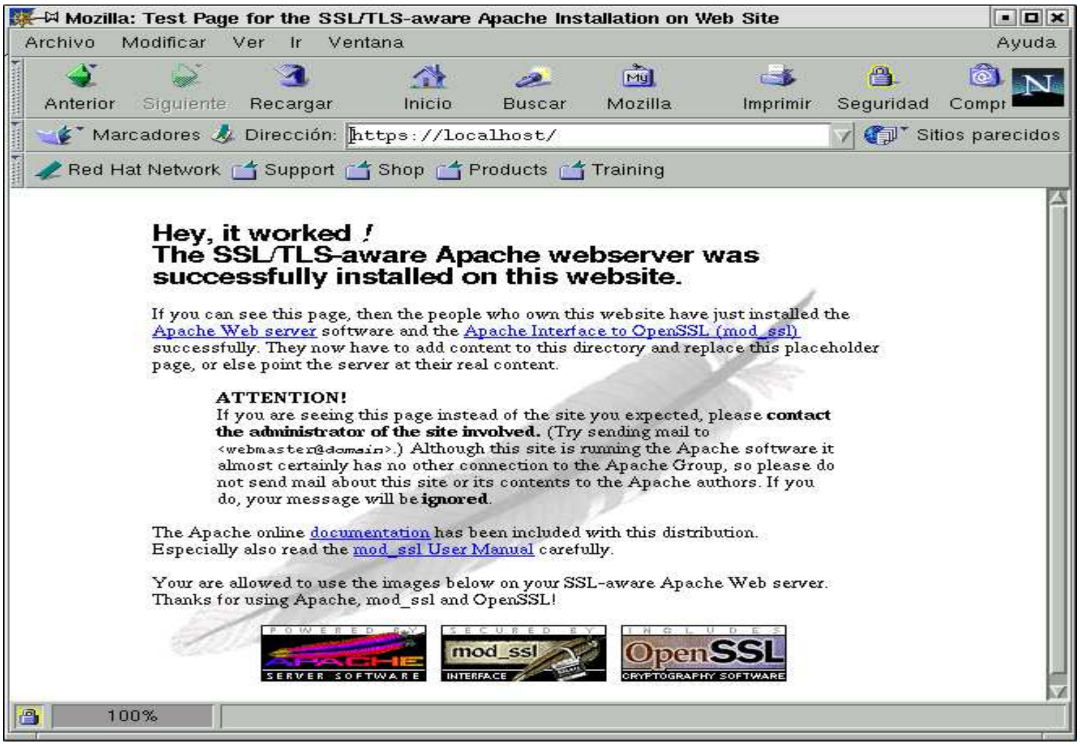
FUENTE: La autora

Finalmente se debe digitar el comando make install para concluir con la creación de los certificados.

8. Probar el funcionamiento del Apache sin SSL con el protocolo http y el puerto 80, para lo cual se debe ingresar al directorio /usr/local/apache/bin/ y digitar ./apachectl start para iniciar el servidor. Ingresar a Nescape Navigator y digitar http://localhost luego parar el apache escribiendo ./apachectl stop.

9. Probar el funcionamiento del Apache con SSL mediante el protocolo https y el puerto 443, ubicándose en el directorio /usr/local/apache/bin/ y digitar ./apachectl startssl luego escribir la misma clave que se ingresó en el momento de la configuración para iniciar el servidor seguro. Ingresar a Nescape Navigator y digitar https://localhost

Gráfico # 28. INICIO DEL SERVIDOR SSL



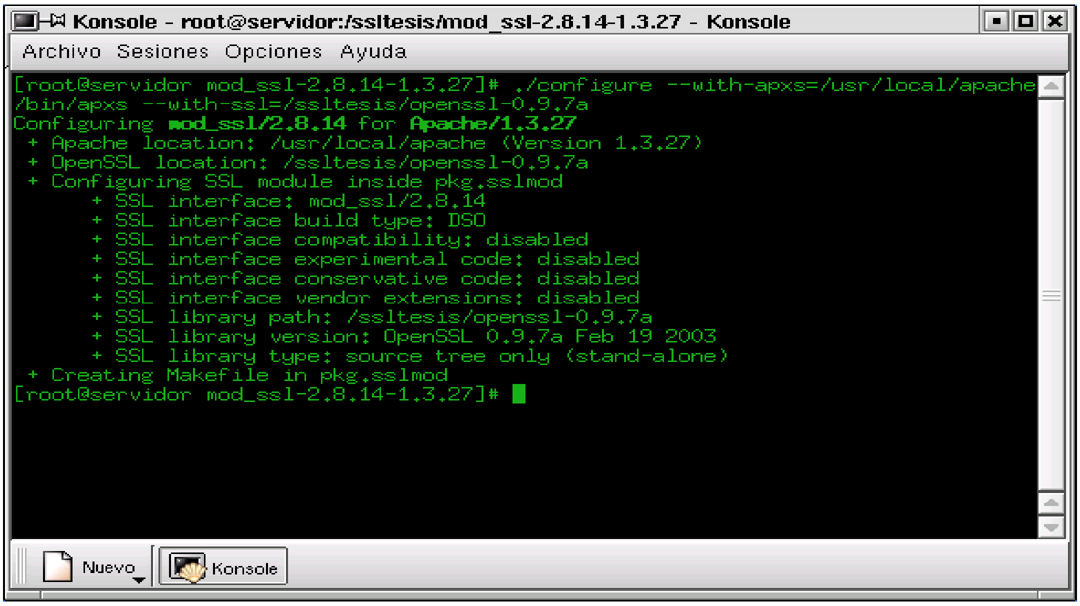
FUENTE: La autora

Luego parar el apache digitando ./apachectl stop, cerrar la ventana del Navegador y continuar con la instalación de los demás paquetes.

10. Ubicarse en el directorio mod_ssl-2.8.12-1.3.27 para configurar el APXS y posteriormente instalar el paquete PHP mediante la actualización de la librería libssl.so y digitar el siguiente código:

    ./configure –with-apxs=/usr/local/apache/bin/apxs -- with-ssl=/openssl-0.9.7a

Gráfico # 29. ACTUALIZACION DEL APXS



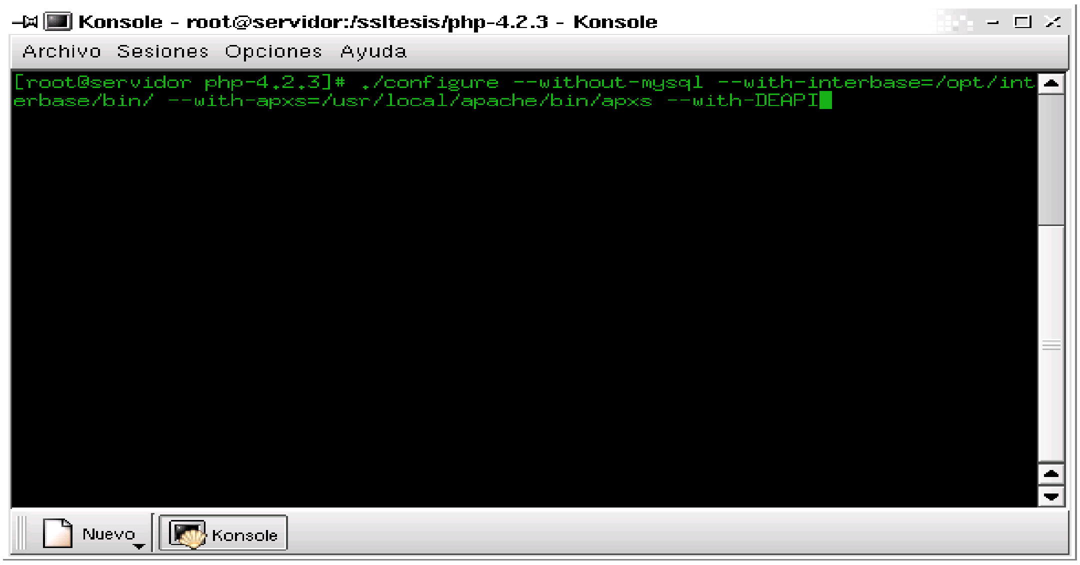
FUENTE: La autora

Después digitar make, luego make install y finalmente make distclean.

11. Ubicarse en el directorio php-4.3.1 para compilar Php4 with Interbase 6. 0 y apxs digitando el siguiente código:

./configure --without-mysql --with-interbase=/opt/interbase --with-apxs=/usr/local/apache/bin/apxs --with-DEAPI

Gráfico # 30. INSTALACIÓN DE LOS PAQUETES PHP4 E INTERBASE 6.0



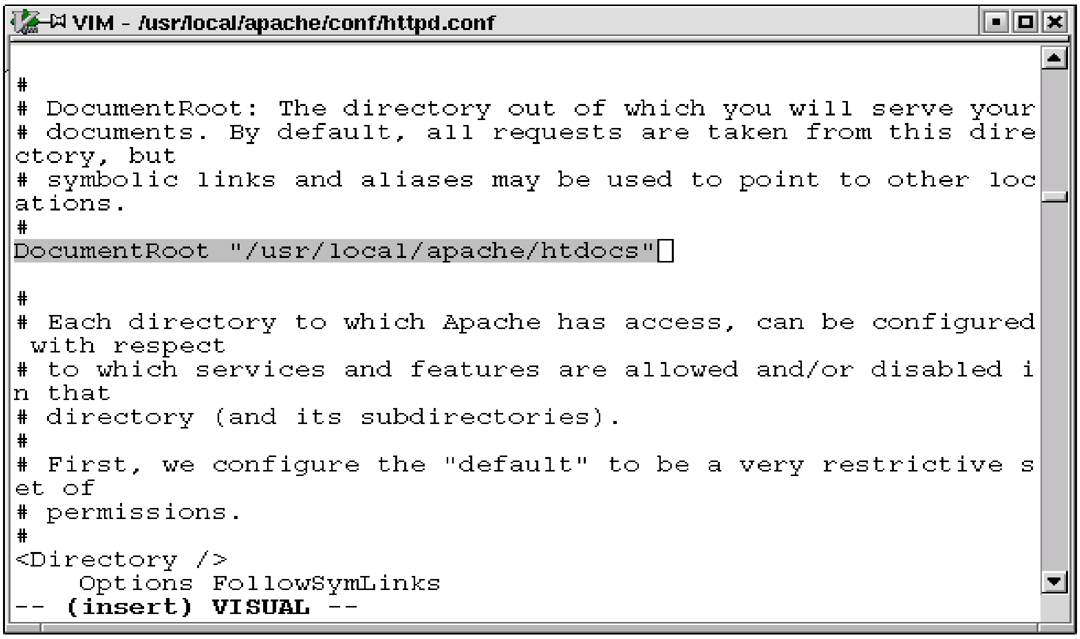
FUENTE: La autora

Posteriormente digitar make y luego make install para finalizar la instalación.

12. Ingresar al directorio /usr/local/apache/conf/ para revisar el archivo httpd.conf, usando el comando gvim. Aquí se puede modificar cualquier opción de acuerdo a las necesidades que se requiera para su correcto funcionamiento. Ejemplo: Se puede modificar la ruta para guardar las páginas web de /usr/local/apache/htdocs/ a "/var/www/html" como se indica en el gràfico # 31.[20]

20 http://www.cambuddys.com/index.php ; Ultimo acceso: Viernes 29 de Agosto del 2003

Gráfico # 31. ARCHIVO HTTP.CONF



```
VIM - /usr/local/apache/conf/httpd.conf

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this dire
ctory, but
# symbolic links and aliases may be used to point to other loc
ations.
#
DocumentRoot "/usr/local/apache/htdocs"

#
# Each directory to which Apache has access, can be configured
 with respect
# to which services and features are allowed and/or disabled i
n that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive s
et of
# permissions.
#
<Directory />
    Options FollowSymLinks
-- (insert) VISUAL --
```

FUENTE: La autora

3.3.3 Ejecución del servidor SSL.

Para iniciar la ejecución el servidor SSL se debe reiniciar la máquina y realizar los siguientes pasos:

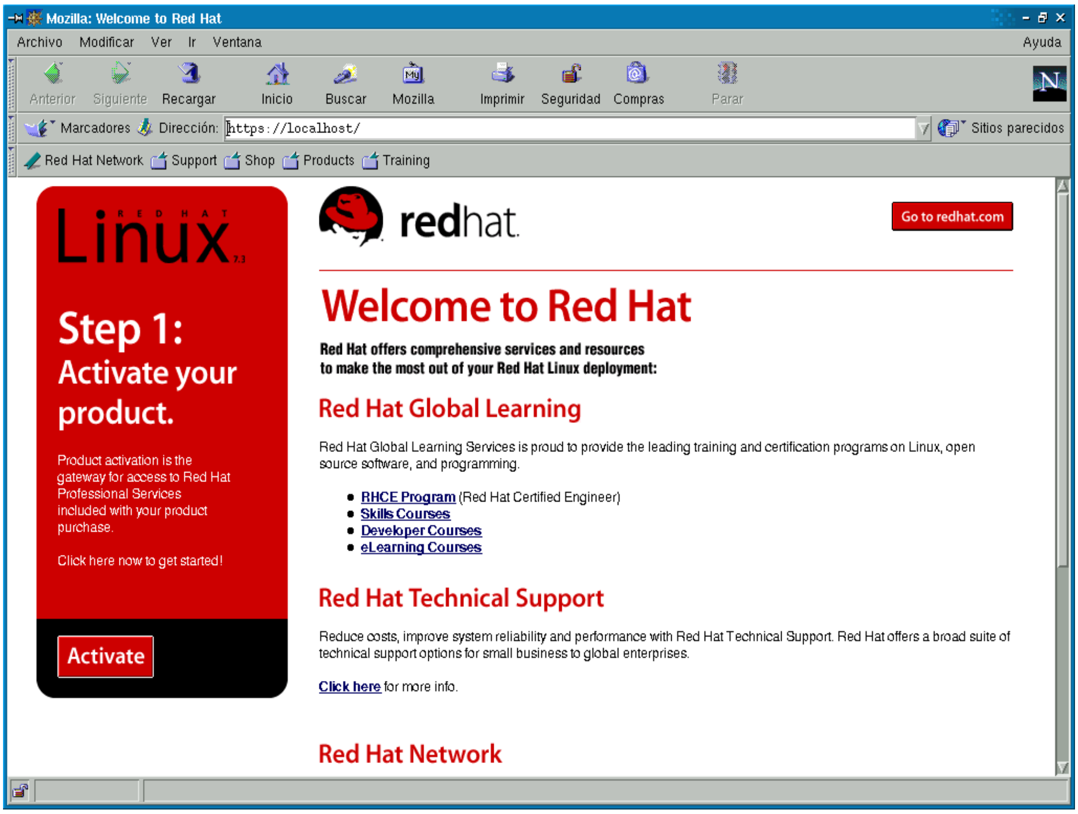
1. Ingresar al directorio /opt/interbase/bin para levantar el Interbase 6.0 con la siguiente línea de código: ./ibmgr –start –forever

2. Ingresar al directorio /usr/local/apache/bin para levantar el Apache con la siguiente línea de código: ./apachectl startssl

Al ejecutar esta instrucción hay que ingresar la misma contraseña con la que se configuró el servidor.

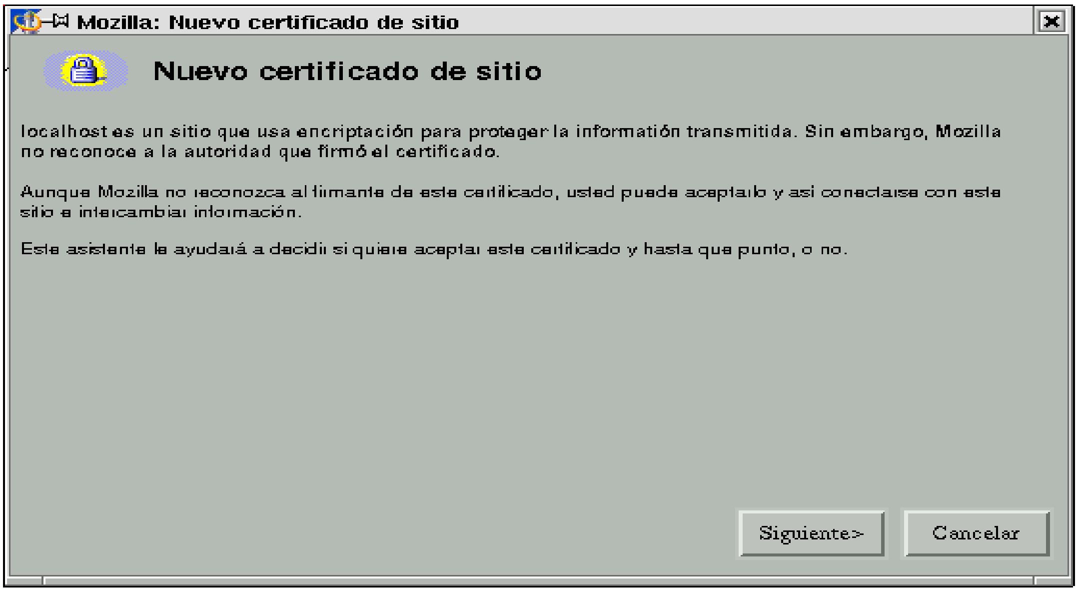
3. Ubicarse con el mouse en la barra de herramientas para escoger la opción K, luego Internet y finalmente Nestcape Navigator. En la ventana del navegador digitar la siguiente línea: https://localhost/ presionar enter y aceptar todos los datos del certificado como se muestra a continuación:

Gráfico # 32. NESTCAPE NAVIGATOR



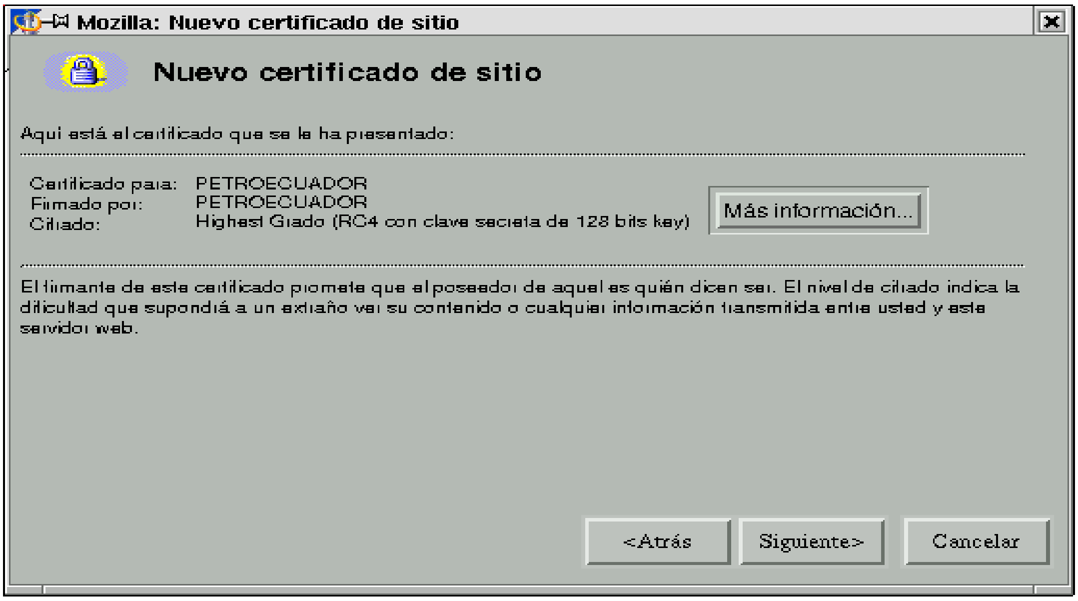
FUENTE: La autora

Gráfico # 33. GENERACIÓN DEL CERTIFICADO (PARTE I)



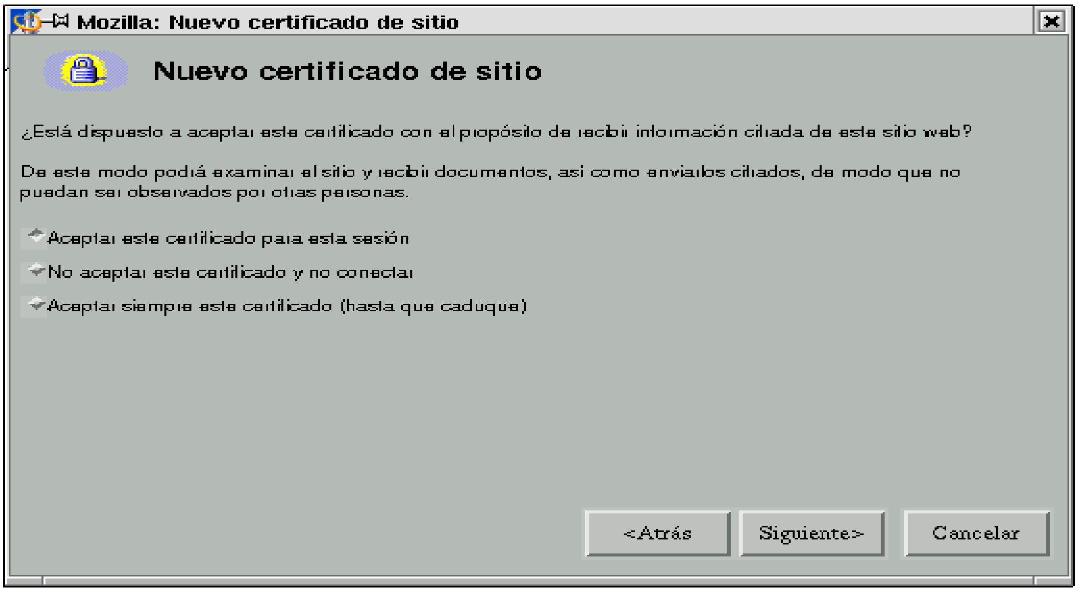
FUENTE: La autora

Gráfico # 34. GENERACIÓN DEL CERTIFICADO (PARTE II)



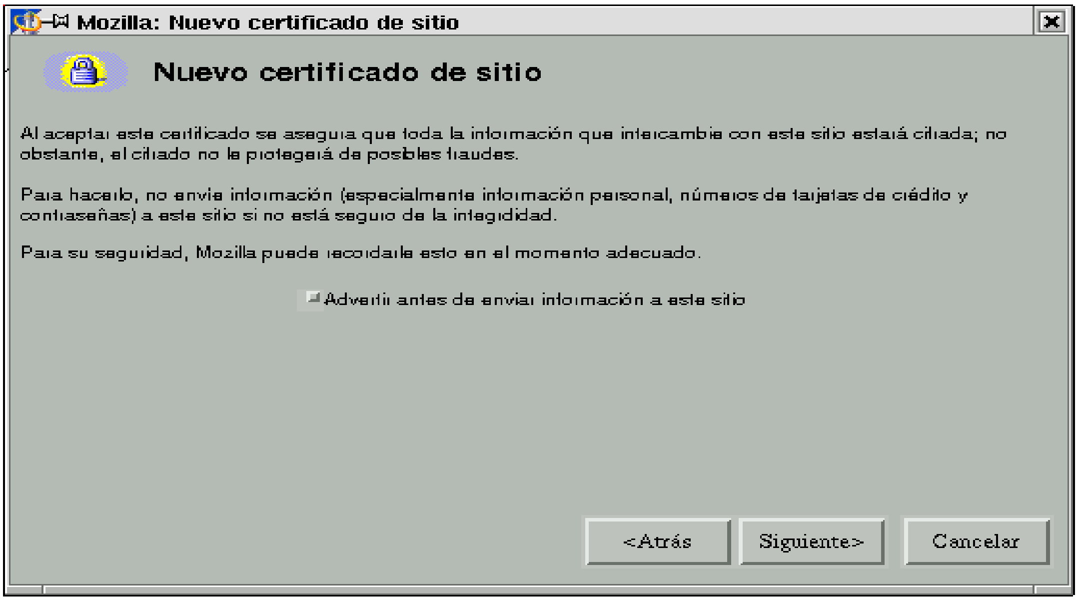
FUENTE: La autora

Gráfico # 35. GENERACIÓN DEL CERTIFICADO (PARTE III)



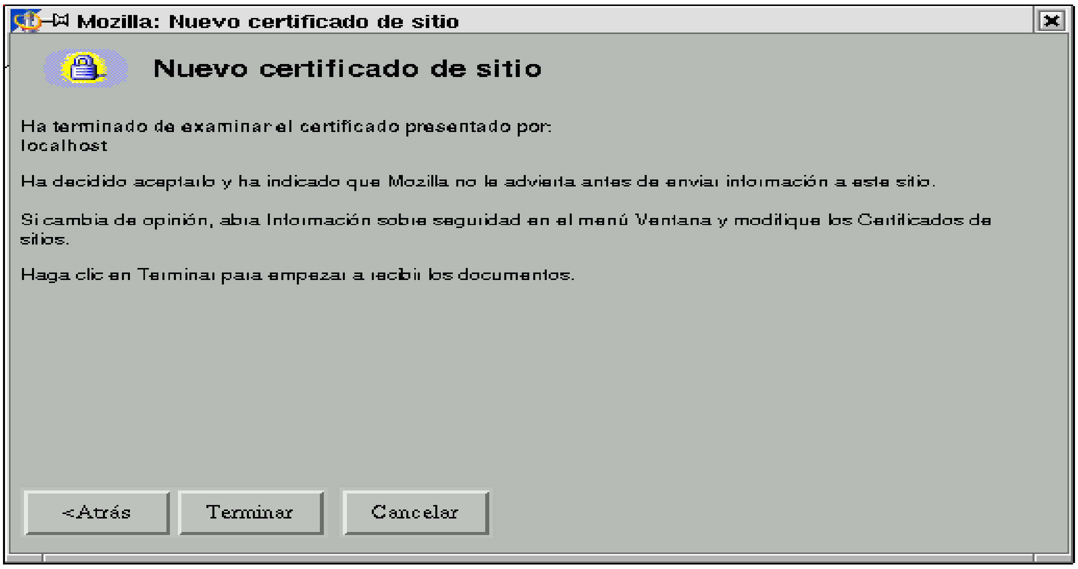
FUENTE: La autora

Gráfico # 36. GENERACIÓN DEL CERTIFICADO (PARTE IV)



FUENTE: La autora

Gráfico # 37. GENERACIÓN DEL CERTIFICADO (PARTE V)

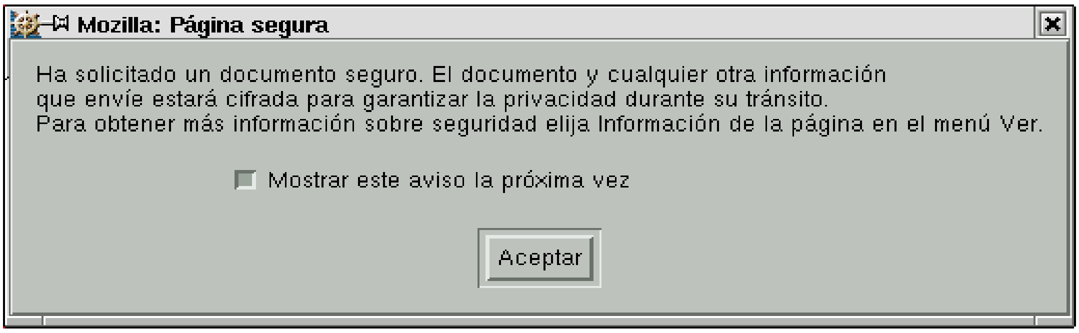


FUENTE: La autora

Gráfico # 38. GENERACIÓN DEL CERTIFICADO (PARTE VI)

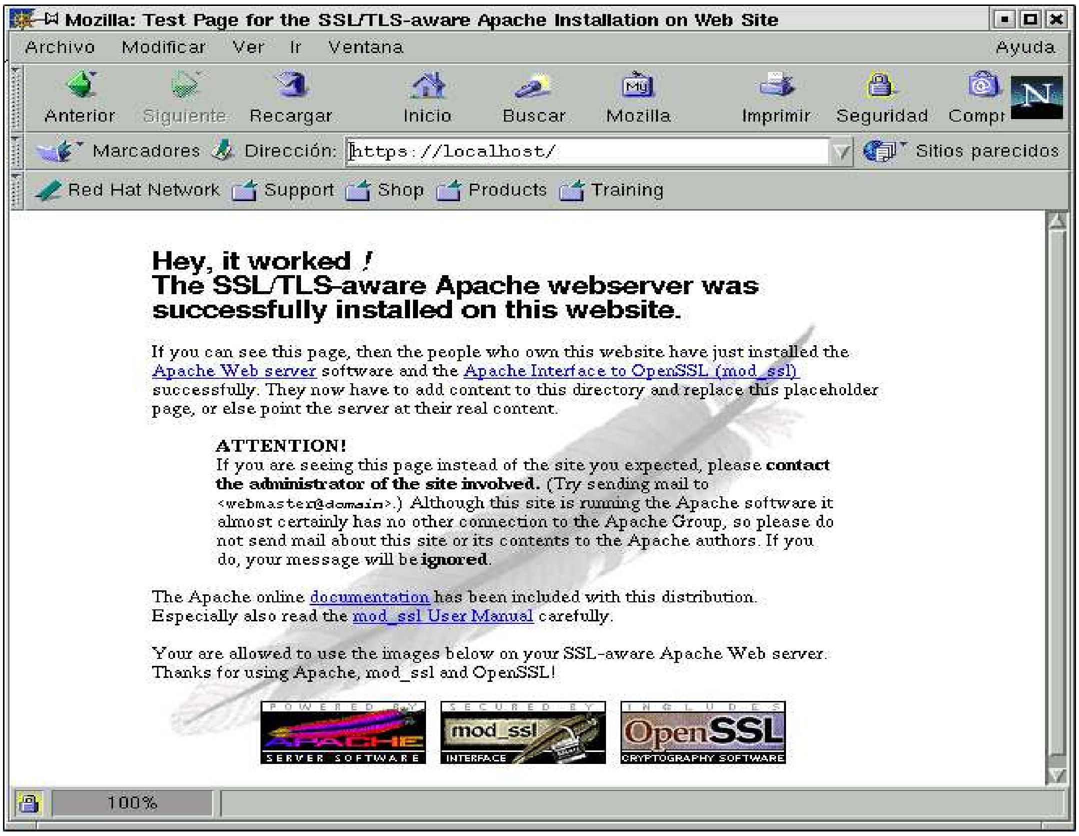
FUENTE: La autora

Gráfico # 39. AVISO DE TRANSMISIÓN DE INFORMACIÓN SEGURA



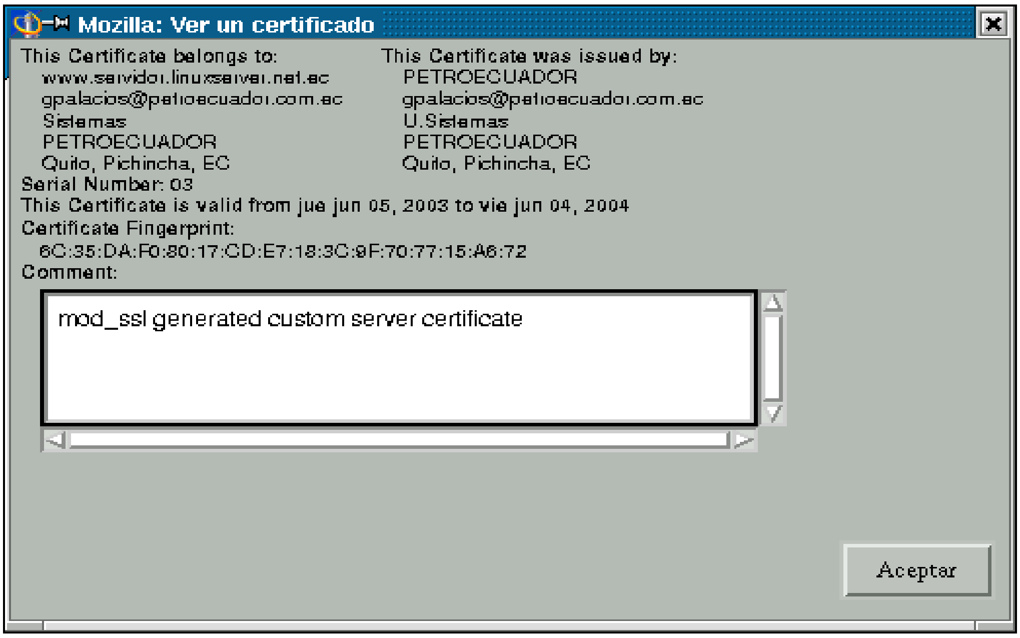
FUENTE: La autora

Gráfico # 40. PAGINA INDEX.HTML



FUENTE: La autora

Gráfico # 41. CERTIFICADO



FUENTE: La autora

Esta página se denomina index.html y por default se carga al inicio del sitio del servidor ssl; también se puede digitar https://localhost443/ para verificar que exactamente esta utilizando el puerto 443 de seguridad. Además se puede observar el candado cerrado en la parte inferior izquierda, lo que quiere decir que toda la información está siendo encriptada correctamente, si se desea se puede dar clic en este candado y se obtendrá información del sitio y su certificado.

4. A partir de esta página se puede ejecutar cualquier página web, ubicándola en la ruta /usr/local/apache/htdocs/, también aquí se puede crear directorios personales

y dentro de éstos páginas web que obviamente van a ser enriptadas y transmitidas por este canal seguro.

5. Para salir de este proyecto se debe cerrar la ventana del navegador, abrir una consola de trabajo para parar el servidor apache en la misma ubicación /usr/local/apache/bin con la instrucción: ./apachectl stop y finalmente parar el Interbase en la dirección /opt/interbase/bin con la instrucción: ./ibmgr –shut –passw masterkey.[21]

3.3.4 Pasos previos para comprar un certificado a la Autoridad certificadora Verising

Cuando ya se tiene un servidor compatible SSL, se debe decidir a qué autoridad de certificación (AC) se va a comprar un certificado digital, lo cual se puede ver en los navegadores más comunes como son: Netscape Comunicator e Internet Explorer. Entre ellas se cuentan VeriSign, Thawte, AT&T, etc. Por comodidad se puede solicitar un certificado de prueba gratuito de 14 días de validez a VeriSign. Además esta empresa ofrece los siguientes tipos de servicios:

- Sitio de Comercio (con 40-bits de encriptación y Payflow Pro servicios de administración de pago "online").

21 https://localhost/ ; Ultimo acceso: Martes 2 de Septiembre del 2003

- Sitio de Comercio Pro (con 128-bits de encriptación y Payflow Pro servicios de administración de pago "online").
- Sitio Seguro (con 40-bit de encriptación)
- Sitio Seguro Pro (con 128-bit de encriptación)

Las Soluciones VeriSign incluyen:

- Un programa de garantía extendido que protege a los clientes de robo, corrupción, personificación o pérdida de uso de un certificado.
- El sello de los sitios seguros de VeriSign que permiten a los visitantes comprobar su información ID servidor y estado en tiempo real.
- Un proceso de autentificación, que asegura que VeriSign verifica la identidad de todos los sitios equipados con un ID servidor.

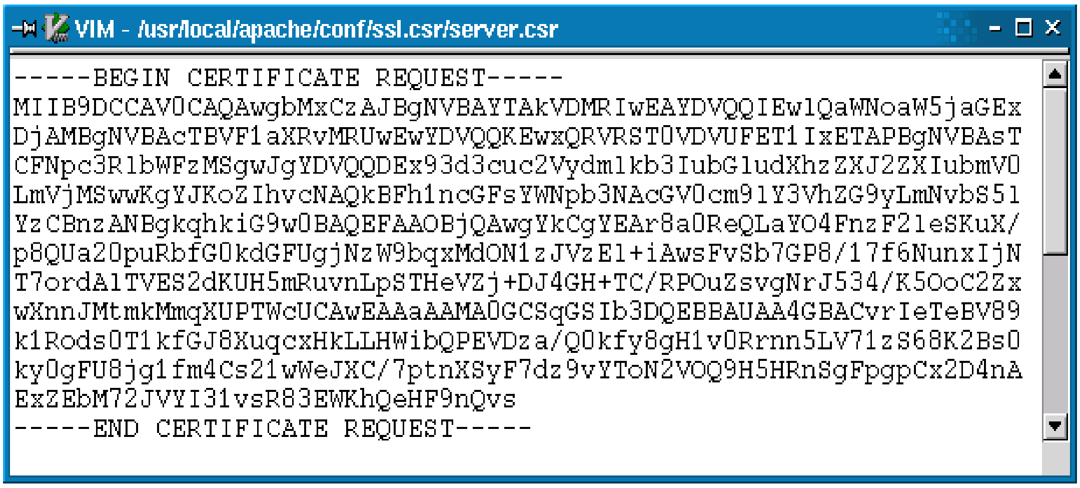
Una vez que se ha decidido en la adquisición de un certificado de un sitio seguro, ingresar a la página http://www.verisign.com/server/ para seleccionar un certificado y seguir paso a paso las instrucciones del proceso de compra.

1. Para los certificados de sitio seguros se puede elegir un **Two Year Option** y confirmar la localización de su servidor seguro, luego pulsar en **Continue.**
2. La página siguiente es **Preparing for Enrollment**. Esta página proporciona un resumen de la información que se necesita proporcionar a VeriSign. Leer esta

página y asegurarse que tiene la información necesaria, antes de continuar con el proceso. Cuando se termine, pulsar en el botón **Continue** en la parte inferior de la página.

3. La siguiente página es **CSR Wizard: Verify Distinguished Name**. Si no se tiene ya generado una clave y CSR hay que generarlo necesariamente, luego seleccionar **I have already prepared a CSR for this enrollment** y pulsar **Continue**.

Gráfico # 42. ARCHIVO CSR

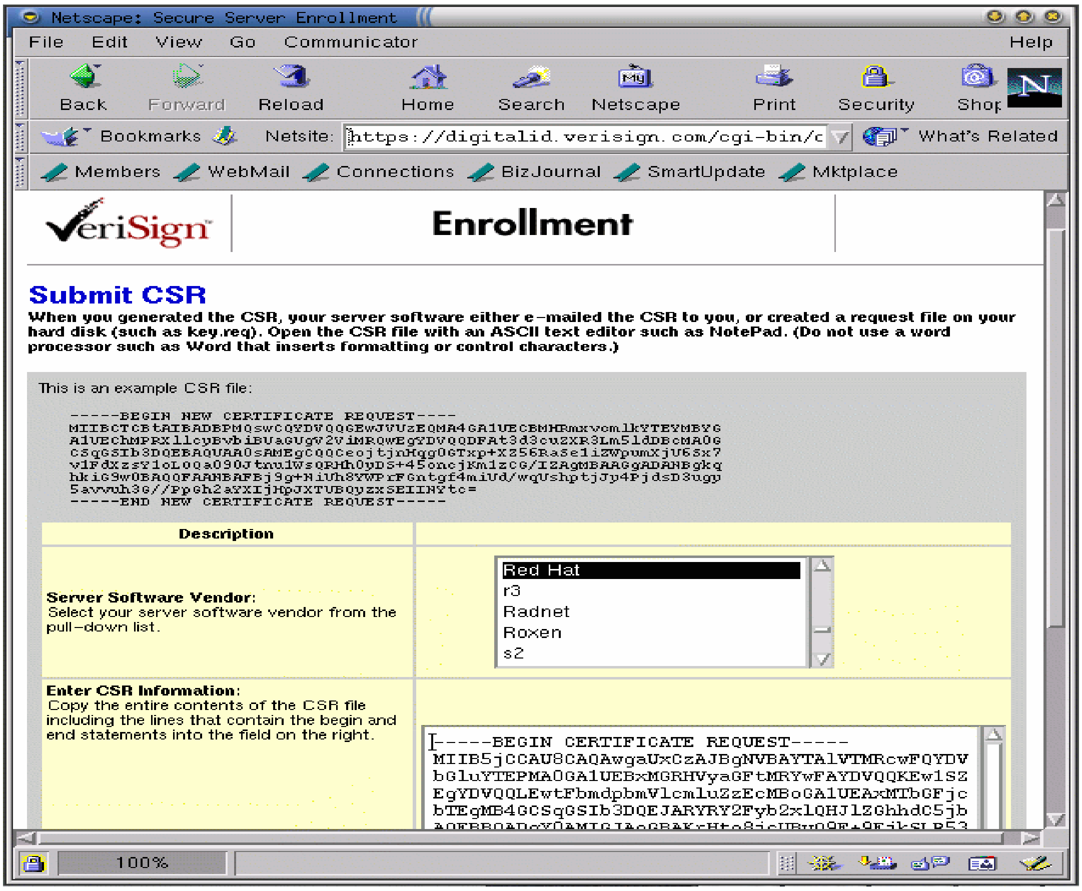


-----BEGIN CERTIFICATE REQUEST-----
MIIB9DCCAV0CAQAwgbMxCzAJBgNVBAYTAkVDMRIwEAYDVQQIEw1QaWNoaW5jaGEx
DjAMBgNVBAcTBVF1aXRvMRUwEwYDVQQKEwxQRVRST0VDVUFET1IxETAPBgNVBAsT
CFNpc3RlbWFzMSgwJgYDVQQDEx93d3cuc2Vydml1kb3IubG1udXhzZXJ2ZXIubmV0
LmVjMSwwKgYJKoZIhvcNAQkBFh1ncGFsYWNpb3NAcGV0cm91Y3VhZG9yLmNvbS51
YzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAr8a0ReQLaYO4FnzF2leSKuX/
p8QUa20puRbfG0kdGFUgjNzW9bqxMdON1zJVzE1+iAwsFvSb7GP8/17f6NunxIjN
T7ordAlTVES2dKUH5mRuvnLpSTHeVZj+DJ4GH+TC/RPOuZsvgNrJ534/K5OoC2Zx
wXnnJMtmkMmqXUPTWcUCAwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBACvrIeTeBV89
k1Rods0T1kfGJ8XuqcxHkLLHWibQPEVDza/Q0kfy8gH1v0Rrnn5LV71zS68K2Bs0
ky0gFU8jg1fm4Cs21wWeJXC/7ptnXSyF7dz9vYToN2VOQ9H5HRnSgFpgpCx2D4nA
ExZEbM72JVYI31vsR83EWKhQeHF9nQvs
-----END CERTIFICATE REQUEST-----

FUENTE: La autora

4. Seleccionar **Red Hat** desde la lista **Server Software Vendor** como se indica en el gráfico # 43.

Gráfico # 43. DEMANDA DE UN CERTIFICADO A VERISING



FUENTE: http://europe.redhat.com/documentation/rhl7/ref-guide-es/s1-securing-buycert.php3

5. Pegar el contenido del CSR generado en la caja de texto **Enter CSR Information**. Para lo cual hay que ingresar al directorio /etc/httpd/conf/ssl.csr para mostrar el contenido del fichero server.csr usando el comando **cat server.csr** y señalar el contenido del fichero con el botón izquierdo del ratón, luego ingresar a la caja de texto de la página Web y pulsar el botón central del ratón para pegar el texto señalado. Cuando este copiado y pegado el CSR hay que tener cuidado

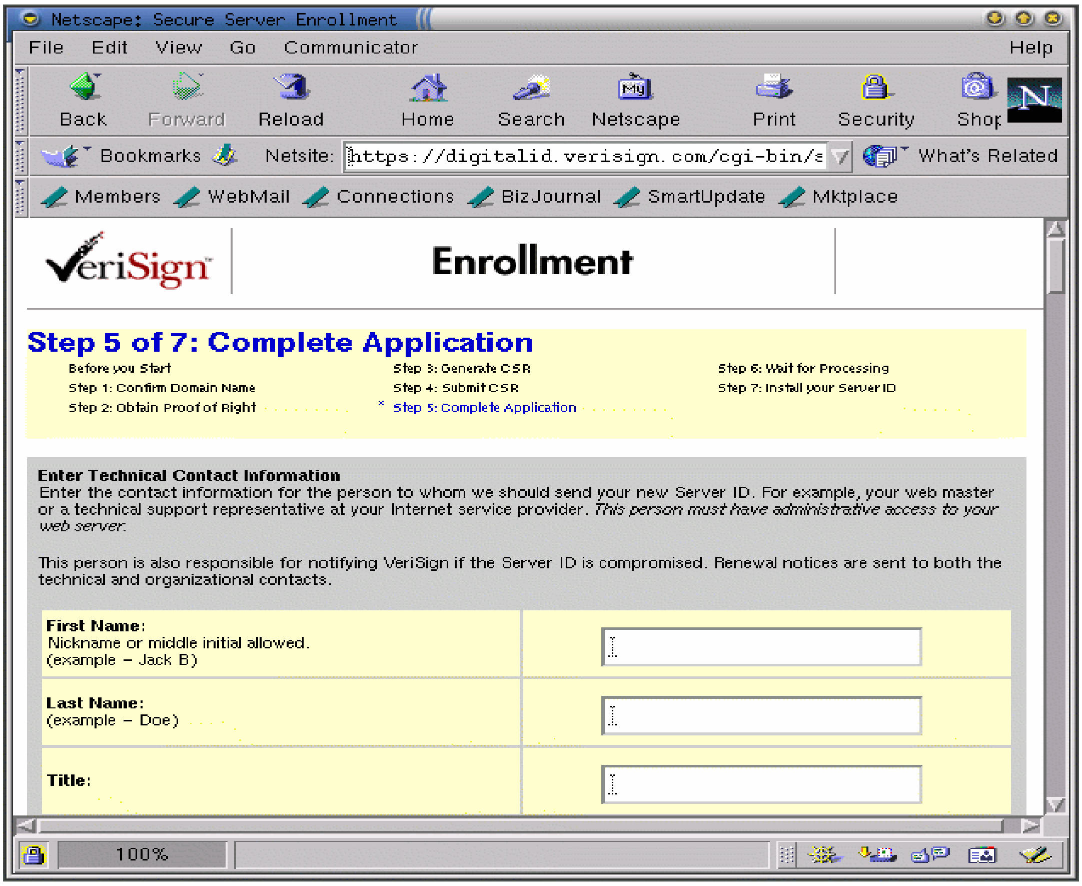
de no copiar cualquier espacio vacío o blanco extras antes o después del texto (incluyendo las líneas **-----BEGIN CERTIFICATE REQUEST-----** y **-----END CERTIFICATE REQUEST-----**), ya que las certificadoras rechazan CSRs que incluyen estos espacios, después de haber pegado el CSR pulsar el botón **Continue**.

6. El siguiente paso es para **Provide Proof of Right,** esto significa que necesita probar a VeriSign que su organización es legítima. VeriSign primero intenta emparejar el nombre organizacional proporcionado por la base de datos de Dun & Bradstreet. Si la organización fue encontrada se puede seleccionar, pero si no fue encontrada, seleccionar **My company and/or my company's correct address is not displayed in this list** y pulsar en **Continue**.
7. La manera más fácil para probar la identidad de la organización es proporcionar el número D-U-N-S, pero si no se tiene este número VeriSign ofrece otras alternativas. Una vez que se tiene los documentos requeridos, continuar con el proceso de matriculación.
8. Después de seleccionar la organización correcta de la lista de la base de datos de Dun & Bradstreet y haber pulsado en **Continue**, la siguiente página es **Confirm Domain Registration**. En esta página, VeriSign es verificado para ver si su dominio esta registrado.
9. El nombre de dominio debe ser registrado para la **Organization name listed in domain registry** debe ser la misma como para la **Organization name you entered**. Si no son los mismos, probablemente se necesita un nuevo CSR que

incluya la información correcta. En la mayoría de los casos, los dos campos será lo mismo, puede seleccionar **These organization names match** y entonces pulsar en **Continue**.

10. La siguiente página debe felicitar al pasar la verificación de la validación inicial de VeriSign. Pulsar en **Continue** y se obtendrá la página, **Complete Application**, como se muestra en el gráfico # 44.

Gráfico # 44. APLICACIÓN PARA CERTIFICADO DE VERISIGN



FUENTE: http://europe.redhat.com/documentation/rhl7/ref-guide-es/s1-securing-buycert.php3

11. Completar en la selección **Enter Technical Contact Information** con información acerca de su administrador o webmaster Red Hat Linux Apache/SSL Server.
12. Rellenar la sección **Enter Organizational Contact Information** con la información apropiada, de acuerdo con las instrucciones proporcionadas por VeriSign.
13. Completar en la selección **Enter Billing Contact Information** con información para la persona que será avisada para el propósito.
14. Teclear en "challenge phrase" y en "reminder question" sobre el área proporcionada. Puede preguntar por su "challenge phrase" si alguna vez necesita soporte desde VeriSing, asegurarse de grabarlo y guadarlo en alguna parte segura.
15. Leer el acuerdo de subscriptor al final de la página y pulsar en el botón **Continue** en la parte inferior de la página. Después de completar la matriculación con éxito, la información y pago que se ha proporcionado a VeriSign ellos autentificarán la identidad de la organización y emitirán el certificado. Cuando la aplicación ha sido aceptada, mandarán el certificado por e-mail a los contactos técnicos y organizacional proporcionados.
16. Grabar el certificado de VeriSign en el fichero server.crt en /etc/httpd/conf/ssl.crt/.

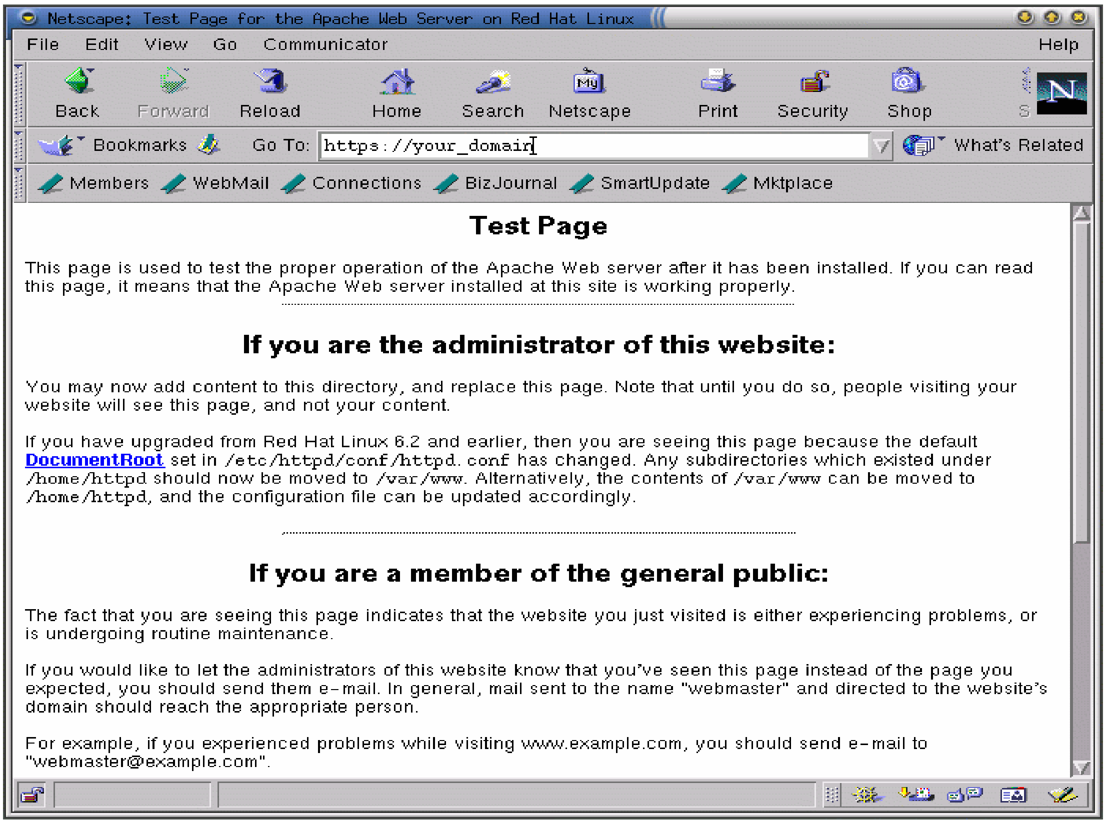
3.3.4.1 Comprobación del certificado

Cuando el servidor seguro es instalado por el programa de instalación Red Hat Linux una clave aleatoria y un certificado genérico son instalados. Puede conectarse al servidor seguro usando el certificado. Si se sigue las instrucciones proporcionadas por esta guía a comprado un certificado de una AC, debe tener un fichero llamado /etc/httpd/conf/ssl.key/server.key, que contiene la clave y un fichero llamado /etc/httpd/conf/ssl.crt/server.crt, que contiene el certificado comprobado. Si la clave y el certificado están en otra parte hay que moverlos a los directorios anteriormente especificados.

Ahora parar e iniciar el servidor. Si el fichero clave es encriptado, será preguntado por el password. Teclear el password y el servidor debe arrancar. Apuntar el browser Web a la página home del servidor. El URL accede a su Red Hat Linux Apache/SSL Server.

Si se usa un certificado AC el browser aceptará automáticamente el certificado y creara la conexión segura. El browser no reconoce automáticamente un test o un certificado self-signed, porque el certificado no esta firmado por la AC. Una vez que el browser acepta el certificado, su Red Hat Linux Apache/SSL Server mostrara la página de inicio por defecto.

Gráfico # 45. PÁGINA DE INICIO DEL CERTIFICADO DE VERISING



FUENTE: http://europe.redhat.com/documentation/rhl7/ref-guide-es/s1-securing-buycert.php3

Nota: Procesar páginas con SSL supone una sobrecarga para el servidor que puede reducir su rendimiento. Por este motivo, se recomienda que se aplique SSL de forma selectiva sólo a aquellas páginas que necesiten cifrado, como las páginas de pago. No aplicar a páginas de solo información comercial.[22]

22 http://europe.redhat.com/documentation/rh/7/ref-guide-es/s1-securing-buycert.php3/ ; Ultimo acceso: Martes 9 de Septiembre del 2003

CAPITULO IV

DESARROLLO DE LA APLICACIÓN VALIDACIÓN DE USUARIO Y CONTRASEÑA

4.1 DISEÑO DE LA APLICACIÓN

4.1.1 Introducción

Actualmente el desarrollo de aplicaciones web son una herramienta de Internet que está en constante desarrollo y que se perfilan como el futuro de las aplicaciones convencionales. El avance de la tecnología ha permitido que Internet se comunique con sus usuarios de modo que cada vez sea mayor la interacción real a través de la red. Las aplicaciones web se utilizan para dar a conocer actividades, publicar datos de interés general, de un tema específico o información confidencial para clientes.

4.1.2 Objetivos

- Diseñar una aplicación web, utilizando el servidor web SSL para validar el nombre y clave de los usuarios registrados en la base de datos de Interbase.
- Permitir al usuario registrar nuevos usuarios en la base de datos.

- Facilitar al usuario hacer consultas de usuarios de la base de datos.
- Realizar un registro de bitácora con los usuarios de tercer nivel de la base de datos.

4.1.3 Justificación

Para demostrar el funcionamiento del servidor seguro SSL se desarrolla la presente aplicación web validación de usuario y clave, la cual permitirá al servidor y al usuario autenticar y negociar entre ambas partes un algoritmo de encriptación y llaves criptográficas, antes de que se transmita o reciba cualquier información. Una vez en línea el navegador se conecta a un servidor seguro SSL usando su llave privada y genera una sesión segura de conexión con el usuario; el navegador decodifica la llave enviada por el servidor y si la descifra correctamente, se abre un canal o conexión segura para transmitir información que estará encriptada o protegida. Además en la página web se visualizará un candado cerrado en la parte inferior izquierda y se cambiará de http a https lo que indica que se está transmitiendo por el puerto 443.

4.1.4 Análisis del sistema

El análisis es transformar las políticas del usuario y el esquema del proyecto en una especificación estructurada. Se necesita modelar el ambiente del usuario con un diagrama de flujo de datos, diagramas de entidad-relación, además de los requerimientos del usuario se prepara un conjunto de presupuestos, cálculos de costos y beneficios.

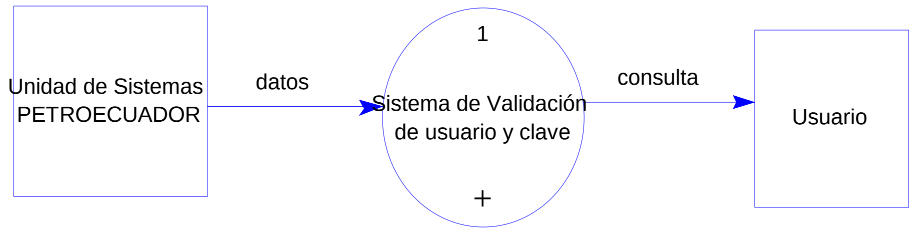
Para el modelado de la aplicación validación de usuario y clave se tiene en claro los procesos que se va a realizar como son: ingreso de nuevos usuarios, consultas de los mismos y su correspondiente registro de bitácora, el cual sirve para ingresar el nombre, diagnóstico y la solución a los problemas que ocurren en los equipos de cómputo.

4.1.5 Metodología de desarrollo del sistema

Para el desarrollo de este sistema en el cual los procesos están definidos claramente se utilizará la metodología estructurada de Edward Yourdon el mismo que cubre todo el ciclo de vida de desarrollo. Esta metodología integra las principales ideas del análisis estructurado y el diseño estructurado en un marco conceptual único y consistenete, además conjuga las técnicas y herramientas de modelado usadas dentro de una organización para obtener una buena comprensión del problema y diseñar una solución de buena calidad. El modelo del sistema está dividido en dos modelos generales que son: modelo esencial que indica lo que el sistema debe hacer para satisfacer las necesidades del usuario y el modelo ambiental que describe los límites del sistema, los estímulos que recibe y como reacciona a ellos.

4.1.6 Diagrama de contexto

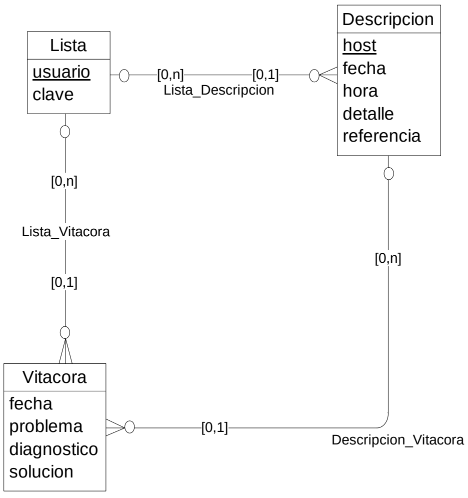
Gráfico # 46. DIAGRAMA DE CONTEXTO



FUENTE: La autora

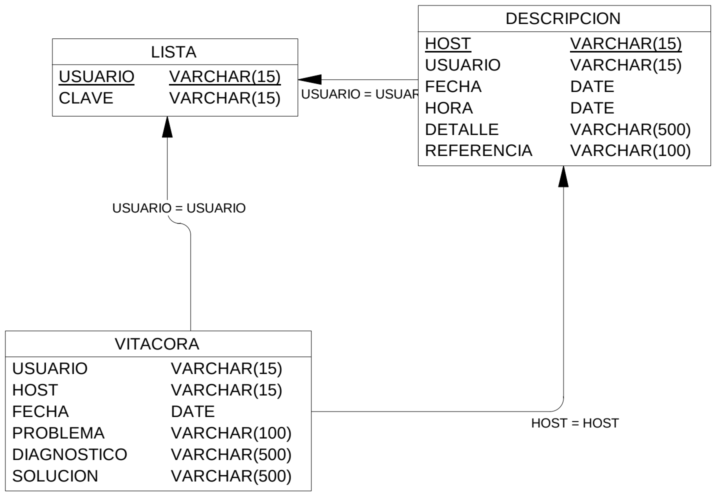
4.1.7 Modelos

Gráfico # 47. DIAGRAMA CONCEPTUAL DE DATOS



FUENTE: La autora

Gráfico # 48. DIAGRAMA FISICO DE DATOS

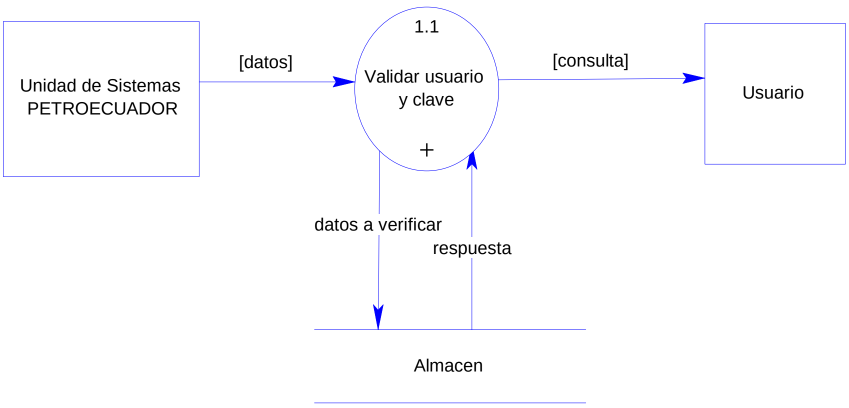


FUENTE: La autora

4.1.8 Definiciòn de procesos

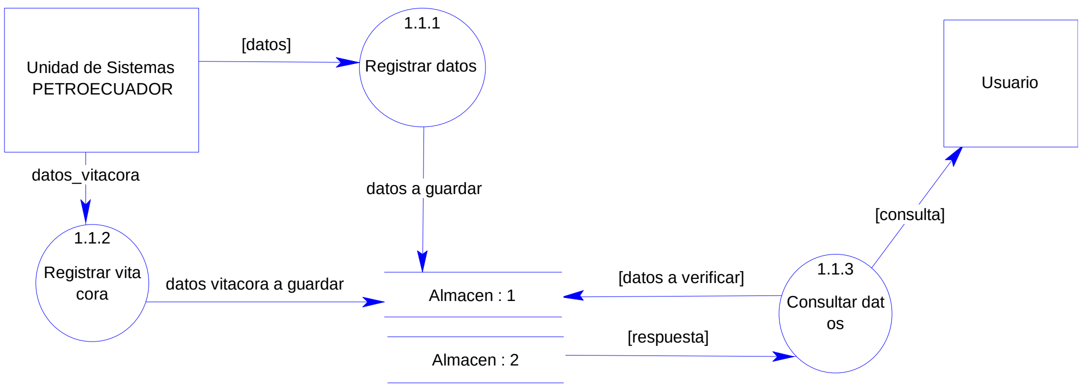
4.1.8.1 Diagrama de flujo de datos

Gráfico # 49. NIVEL 1



FUENTE: La autora

Gráfico # 50. NIVEL 2



FUENTE: La autora

4.1.8.2 Glosario de procesos

Sistema de Validación de usuario y clave [1]

Validar usuario y clave [1.1]

Registrar datos [1.1.1]   Registrar vitacora [1.1.2]   Consultar datos [1.1.3]

Tabla # 3. PROCESO [1] SISTEMA DE VALIDACIÓN DE USUARIO Y CLAVE

Name:	Sistema de Validación de usuario y clave
Code:	
SISTEMA_DE_VALIDACION_DE_USUARIO_Y_CLAVE	
Label:	
Number:	1
Lowest Level:	No

Descripción del Proceso

Proceso principal para el Sistema de validación del usuario y su clave.

FUENTE: La autora

Tabla # 4. PROCESO [1.1] SUBPROCESO VALIDAR USUARIO Y CLAVE

Name:	Validar usuario y clave
Code:	VALIDAR_USUARIO_Y_CLAVE
Label:	
Number:	1.1
Lowest Level:	No

Descripción del Proceso

Proceso que valida al usuario y su clave

FUENTE: La autora

Tabla # 5. PROCESO [1.1.1] SUBPROCESO REGISTRAR DATOS

Name:	Registrar datos
Code:	REGISTRAR_DATOS
Label:	
Number:	1.1.1
Lowest Level:	No

Descripción del Proceso

Proceso que sirve para registrar algunas sugerencias en la base de datos

FUENTE: La autora

Tabla # 6. PROCESO [1.1.2] SUBPROCESO REGISTRAR VITACORA

Name:	Registrar vitacora
Code:	REGISTRAR_VITACORA
Label:	
Number:	1.1.2
Lowest Level:	No

Descripción del Proceso

Proceso que sirve para registrar algunas sugerencias en la base de datos

FUENTE: La autora

Tabla # 7. PROCESO [1.1.3] SUBPROCESO CONSULTAR DATOS

Name:	Consultar datos
Code:	CONSULTAR_DATOS
Label:	
Number:	1.1.3
Lowest Level:	No

Descripción del Proceso

Proceso que sirve para realizar consultas de los usuarios existentes y sus sugerencias

FUENTE: La autora

4.1.8.3 Glosario de flujo de datos

Tabla # 8. FLUJO [1] CONSULTA_DATOS

Connected via	Connected to	Src	Dst
Consulta	Usuario (External Entity)	X	
Datos	Unidad de Sistemas PETROECUADOR (External Entity)		X

FUENTE: La autora

Tabla # 9. FLUJO [1.1] CONSULTA_DATOS

Connected via	Connected to	Src	Dst
Consulta	Usuario (External Entity)	X	
Datos	Unidad de Sistemas PETROECUADOR (External Entity)		X
datos a verificar	Almacen (Data Store)	X	
Respuesta	Almacen (Data Store)		X

FUENTE: La autora

Tabla # 10. FLUJO [1.1.1] DATOS_DATOS A GUARDAR

Connected via	Connected to	Src	Dst
Datos	Unidad de Sistemas PETROECUADOR (External Entity)		X
Datos a guardar	Almacen (Data Store)	X	

FUENTE: La autora

Tabla # 11. FLUJO [1.1.2] DATOS BITÁCORA A GUARDAR_ DATOS BITÁCORA

Connected via	Connected to	Src	Dst
Datos bitócora a guardar	Almacen (Data Store)	X	
Datos_bitácora	Unidad de Sistemas PETROECUADOR (External Entity)		X

FUENTE: La autora

Tabla # 12. FLUJO [1.1.3] CONSULTA_DATOS A VERIFICAR

Connected via	Connected to	Src	Dst
Consulta	Usuario (External Entity)	X	
Datos a verificar	Almacen (Data Store)	X	
Respuesta	Almacen (Data Store)		X

FUENTE: La autora

4.1.8.4 Glosario de almacenamiento de datos

Tabla # 13. LISTA DE ALMACENES DE INFORMACIÓN

Name	Code
Almacen	ALMACEN

FUENTE: La autora

4.1.8.5 Glosario de entidades

Tabla # 14. ENTIDADES

Name	Code
Unidad de Sistemas PETROECUADOR	UNIDAD_DE_SISTEMAS_PETROECUADOR
Usuario	USUARIO

FUENTE: La autora

4.1.8.6 Glosario de documentos de entrada y de salida

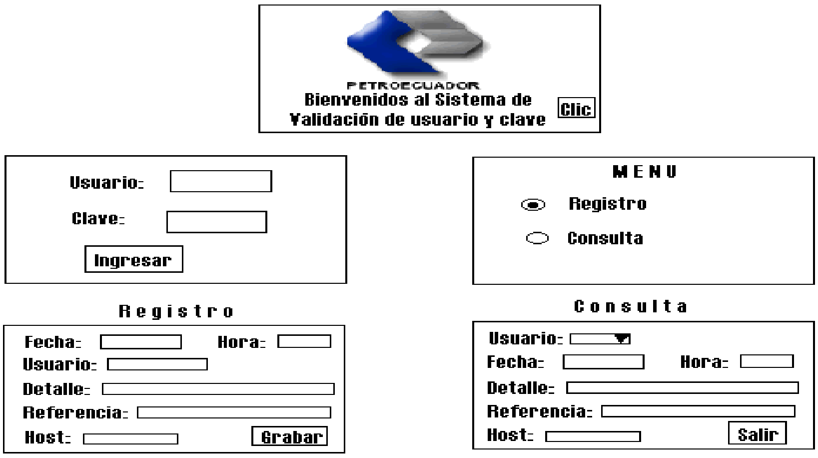
Tabla # 15. DOCUMENTOS DE ENTRADA Y SALIDA

Nº	NOMBRE	E / S	PROCESO
1	Datos	E	Ingresar información de los usuarios
2	Consulta	S	Consultar información de los usuarios
3	Datos a verificar	E	Validar información de los usuarios
4	Respuesta	S	Resultado de la validación de los usuarios
5	Datos a guardar	S	Guardar datos de nuevos usuarios
6	Datos_bitácora	E	Ingresar las actividades de los usuarios
7	Datos vitacora a guardar	S	Guardar las actividades de los usuarios

FUENTE: La autora

4.1.9 DEFINICION DEL PROTOTIPO

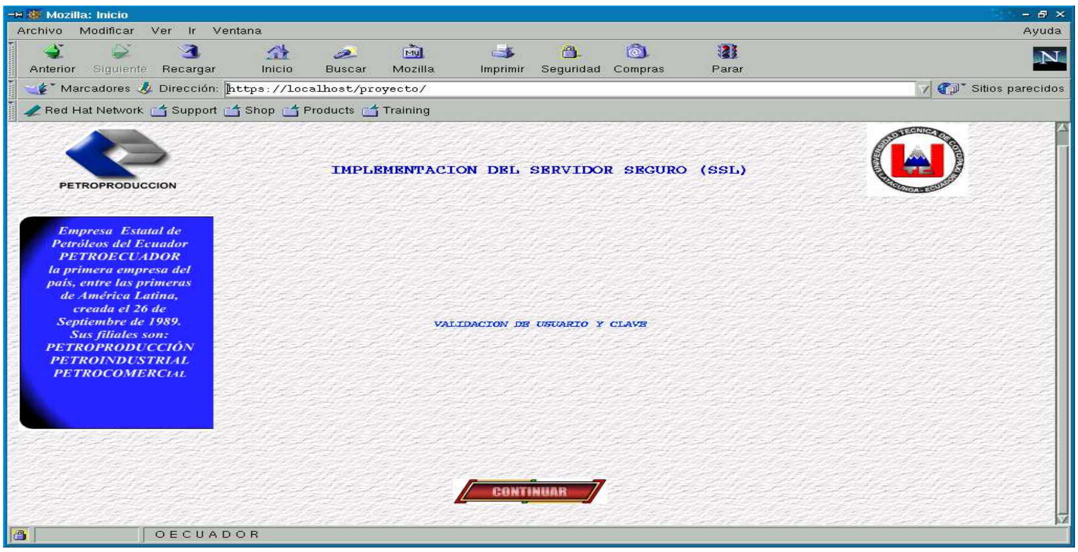
Gráfico # 51. PROTOTIPO



FUENTE: La autora

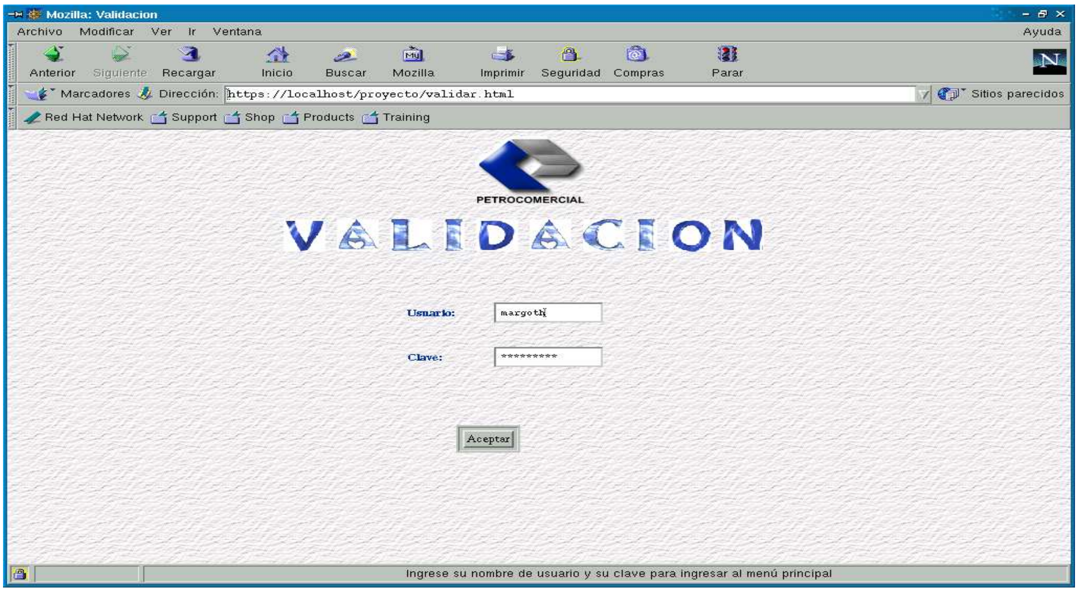
4.1.10 DEFINICIÒN DE PÀGINAS

Gráfico # 52. PÁGINA INDEX.HTML



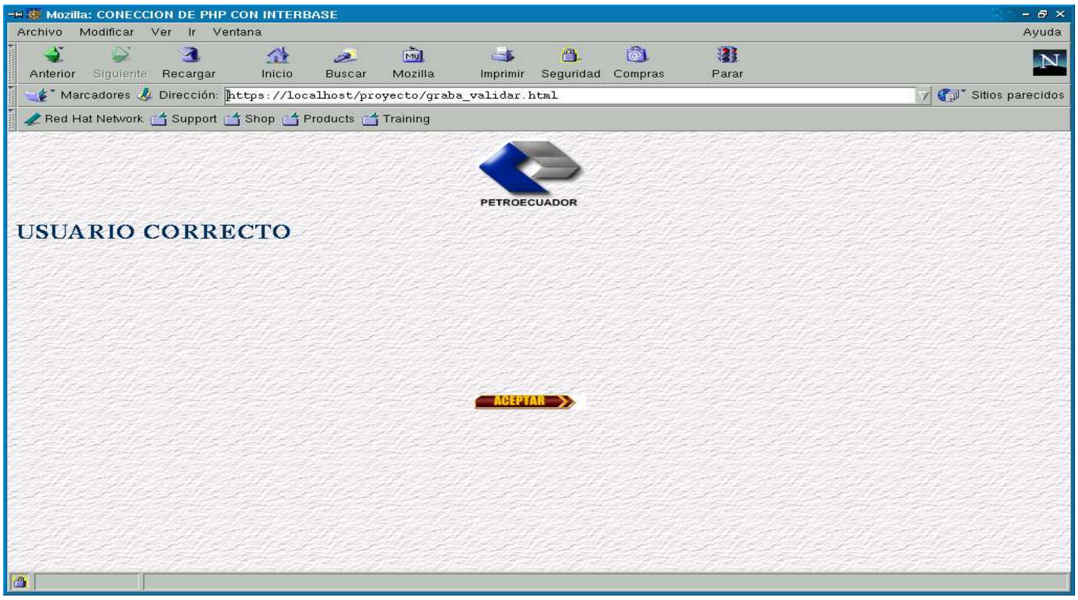
FUENTE: La autora

Gráfico # 53. PÁGINA VALIDAR.HTML



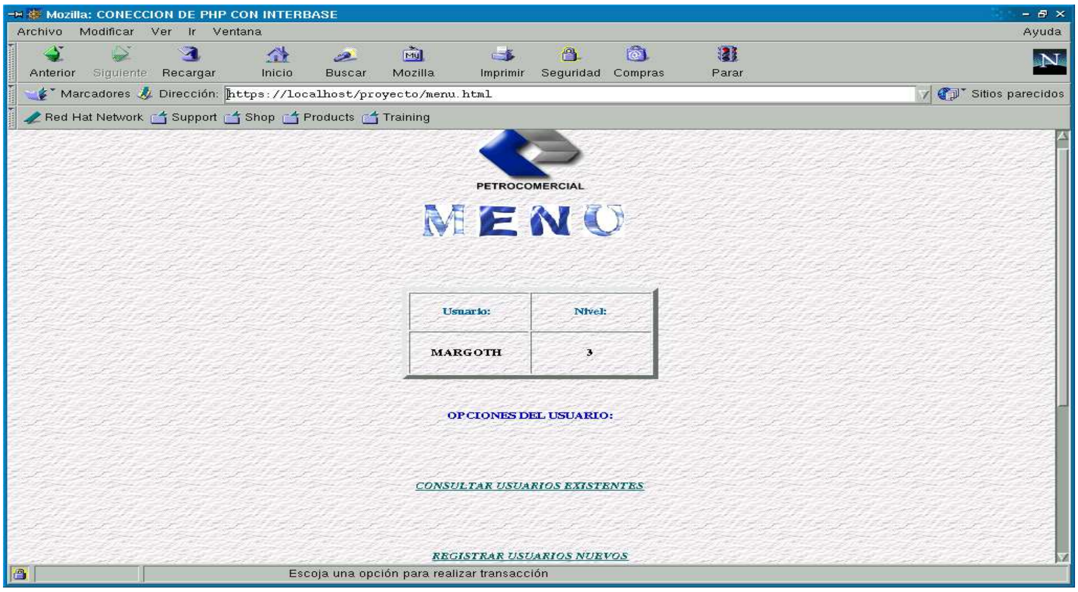
FUENTE: La autora

Gráfico # 54. PÁGINA GRABA_VALIDAR.HTML



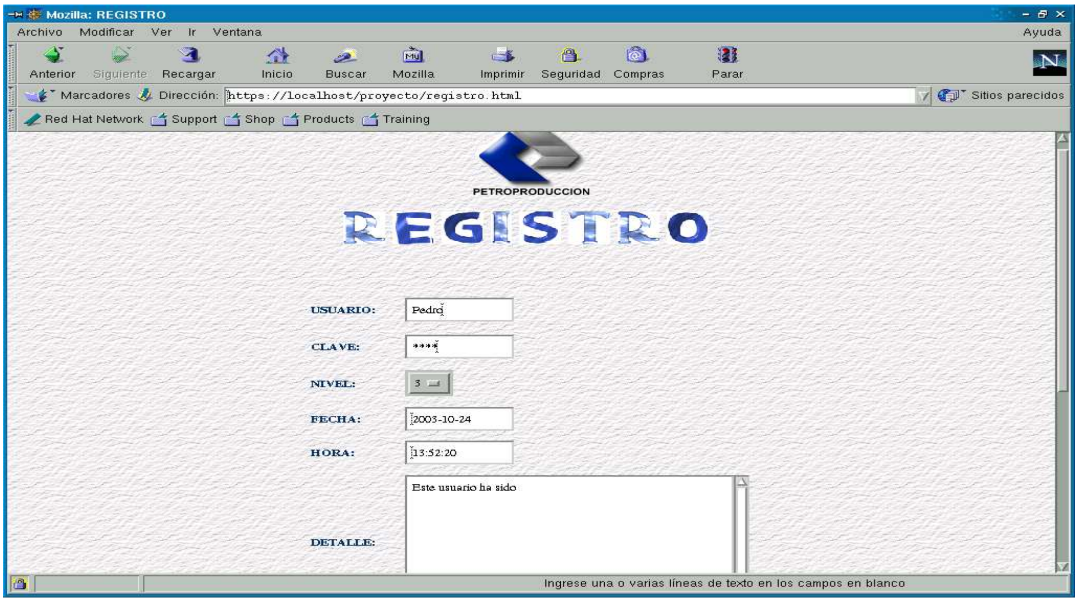
FUENTE: La autora

Gráfico # 55. PÁGINA MENU.HTML



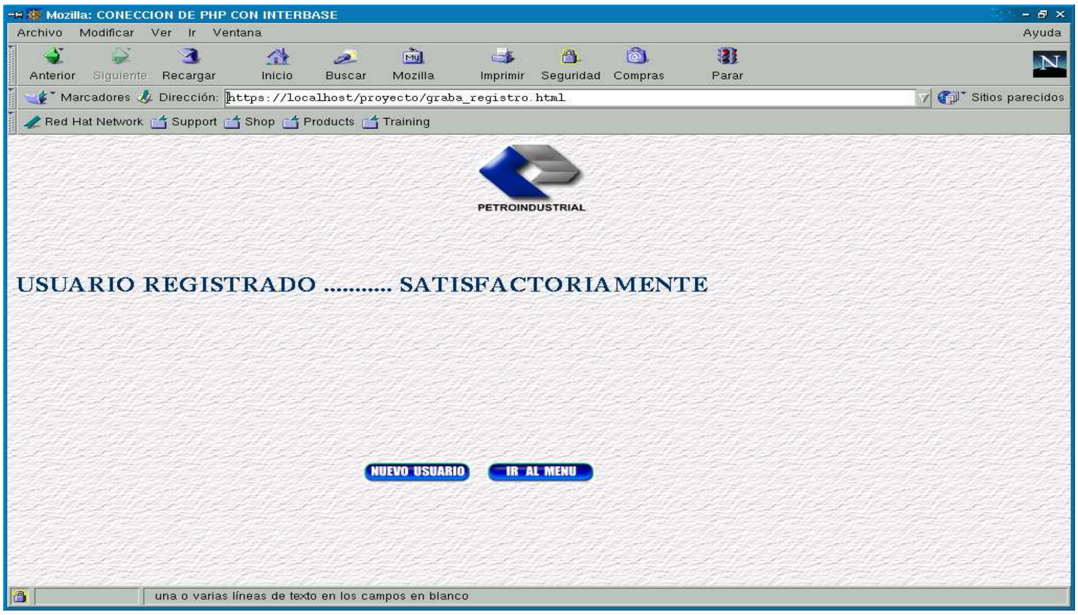
FUENTE: La autora

Gráfico # 56. PÁGINA REGISTRO.HTML



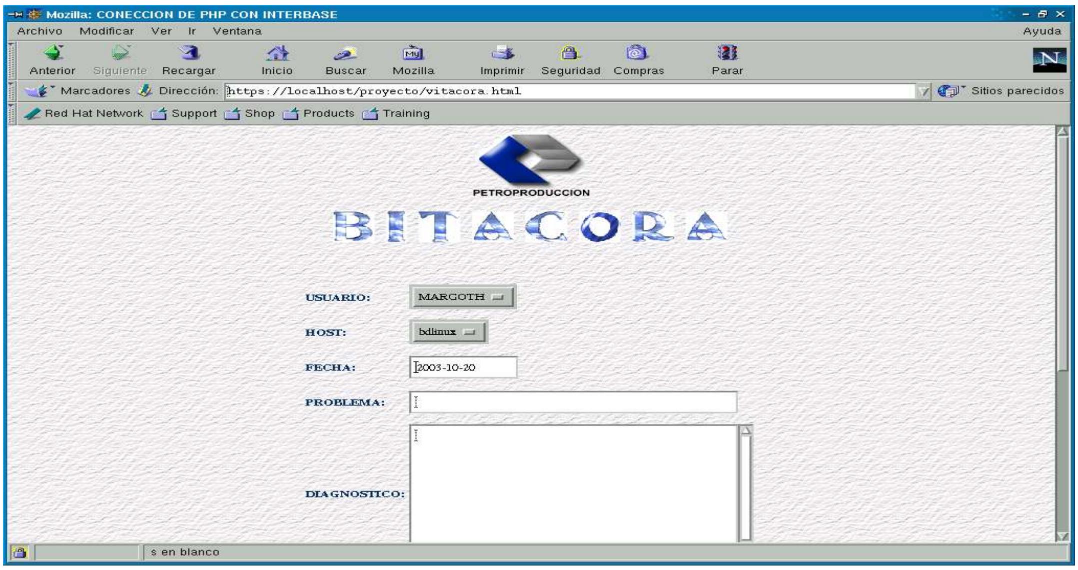
FUENTE: La autora

Gráfico # 57. PÁGINA GRABA_REGISTRO.HTML



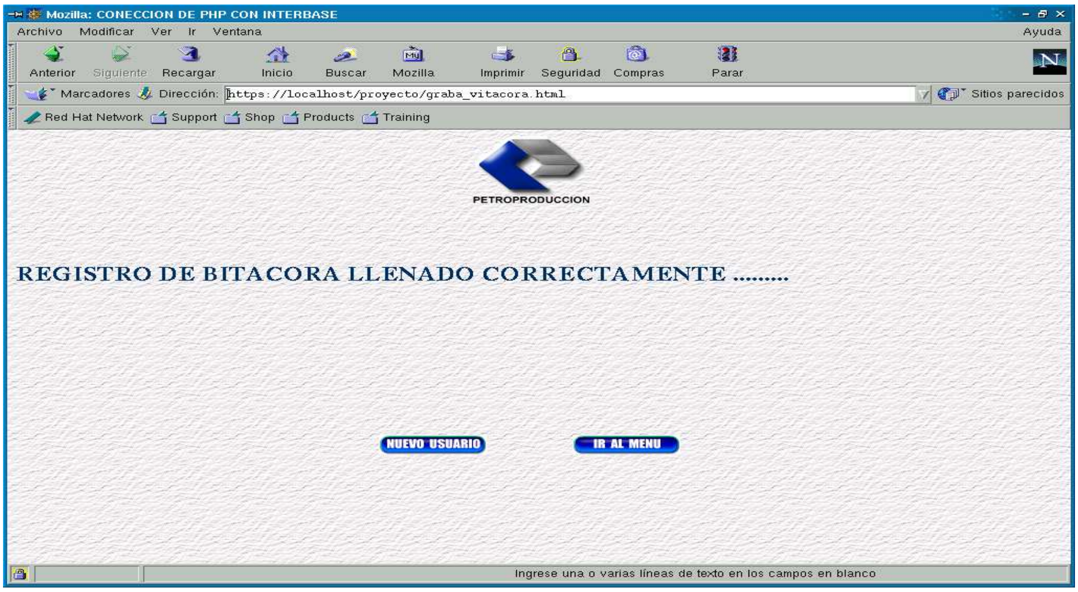
FUENTE: La autora

Gráfico # 58. PÁGINA BITÁCORA.HTML



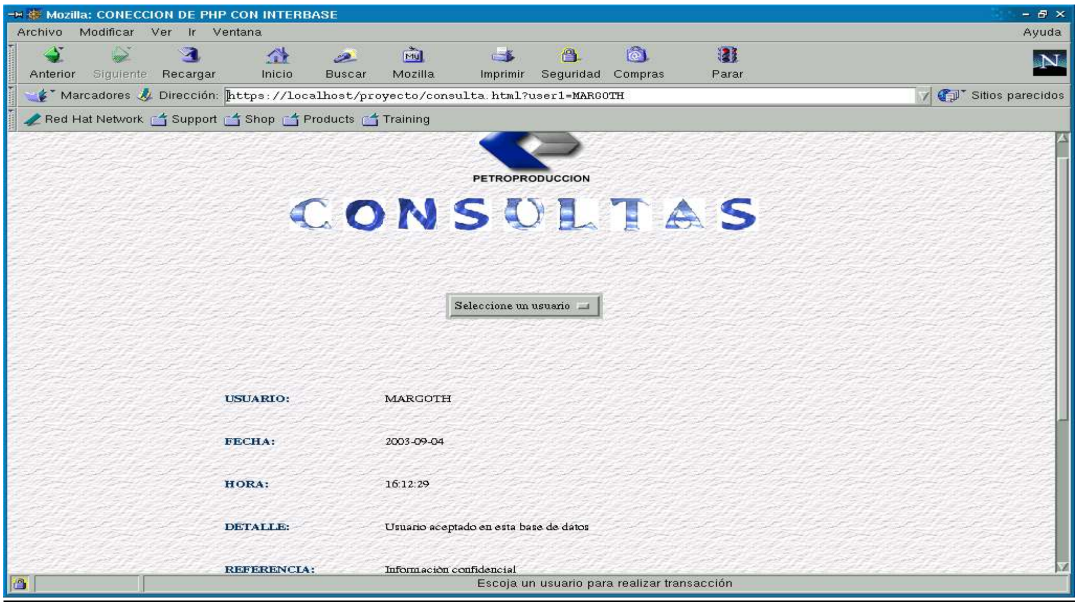
FUENTE: La autora

Gráfico # 59. PÁGINA GRABA_BITÁCORA.HTML



FUENTE: La autora

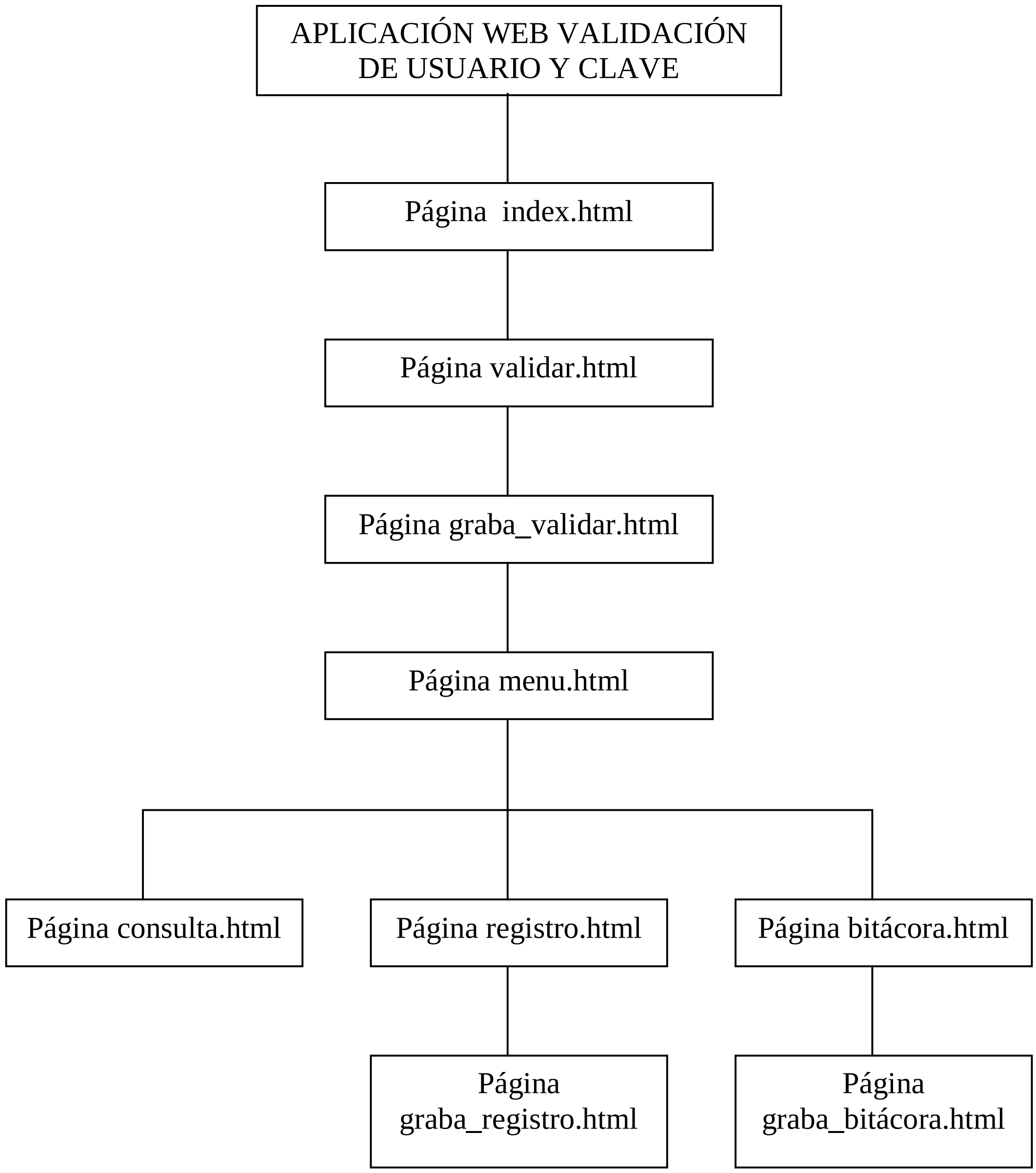
Gráfico # 60. PÁGINA CONSULTA.HTML



FUENTE: La autora

4.1.11 DIAGRAMA JERÁRQUICO DE PÁGINAS

Gráfico # 61. JERARQUÍA DE PÁGINAS



FUENTE: La autora

4.1.12 CARACTERISTICAS FÍSICAS DE ALMACENAMIENTO

Tabla # 16. TABLAS PRINCIPALES

Nº	CODIGO	NOMBRE/DESCRIPCION	TIPO ELM. (*1)	TIPO LOG. (*2)	TIPO FIS. (*3)	LONGITUD ENT.DEC.
1	Lista	Permite escoger campos de la tabla lista	T		A	
2	Descripción	Permite escoger campos de la tabla descripción	T		A	
3	Bitácora	Permite escoger campos de la tabla bitácora	T		A	
*1 A=ARCHIVO / T=TABLA / R=REGISTRO / C=CAMPO *2 K=CLAVE / M=MULTIVALUADO / S=LINEAL / P=FRASE *3 D=FECHA / N=NUMERICO / A=ALFABETICO / +=ALFANUMERICO						

FUENTE: La autora

Tabla # 17. TABLA LISTA

Nº	CODIGO	NOMBRE/DESCRIPCION	TIPO ELM. (*1)	TIPO LOG. (*2)	TIPO FIS. (*3)	LONGITUD ENT.DEC.
1	Usuario	Nombre del usuario	C	K	+	15
2	Clave	Clave asignada al usuario	C	S	+	15
3	Nivel	Nivel del usuario	C	S	+	15
*1 A=ARCHIVO / T=TABLA / R=REGISTRO / C=CAMPO *2 K=CLAVE / M=MULTIVALUADO / S=LINEAL / P=FRASE *3 D=FECHA / N=NUMERICO / A=ALFABETICO / +=ALFANUMERICO						

FUENTE: La autora

Tabla # 18. TABLA DESCRIPCION

Nº	CODIGO	NOMBRE/DESCRIPCION	TIPO ELM. (*1)	TIPO LOG. (*2)	TIPO FIS. (*3)	LONGITUD ENT.DEC.
1	Usuario	Nombre del usuario	C	K	+	15
2	Fecha	Fecha de registro del usuario	C	M	D	15
3	Hora	Hora de registro del usuario	C	M	D	15
4	Detalle	Detalle del usuario	C	S	+	500
5	Referencia	Referencia del usuario	C	S	+	100
6	Host	Nombre del servidor o host	C	S	+	15
*1 A=ARCHIVO / T=TABLA / R=REGISTRO / C=CAMPO *2 K=CLAVE / M=MULTIVALUADO / S=LINEAL / P=FRASE *3 D=FECHA / N=NUMERICO / A=ALFABETICO / +=ALFANUMERICO						

FUENTE: La autora

Tabla # 19. TABLA BITACORA

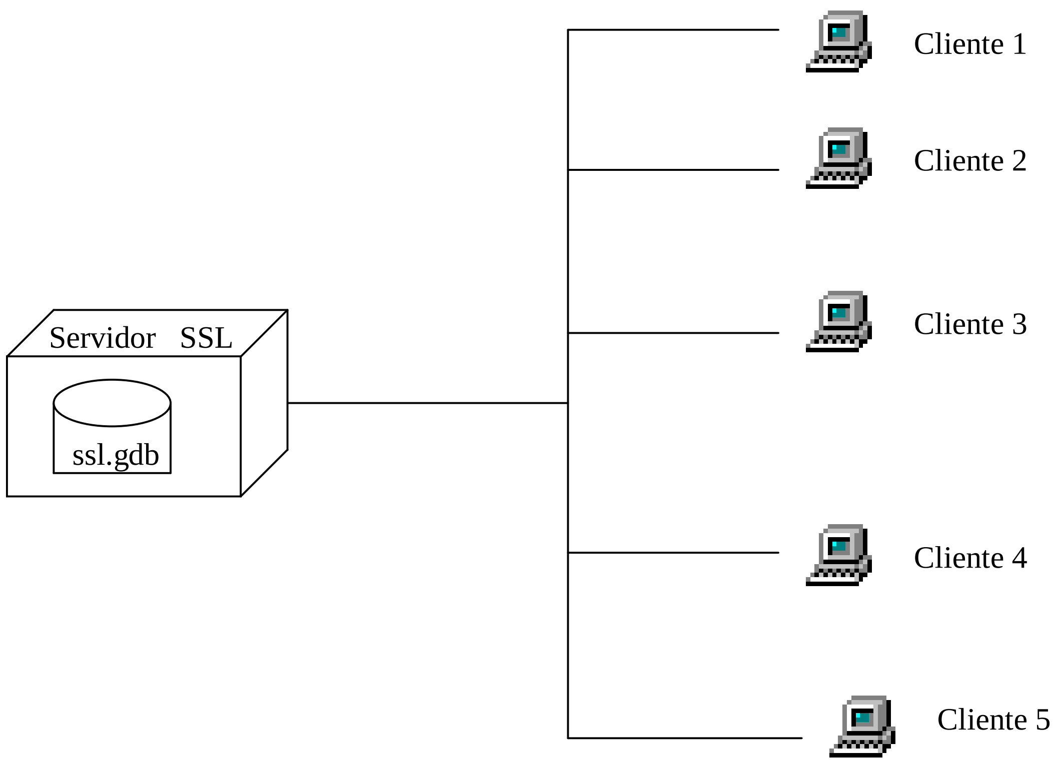
Nº	CODIGO	NOMBRE/DESCRIPCION	TIPO ELM. (*1)	TIPO LOG. (*2)	TIPO FIS. (*3)	LONGITUD ENT.DEC.
1	Usuario	Nombre del usuario	C	K	+	15
2	Fecha	Fecha de registro de bitácora	C	M	D	15
3	Problema	Nombre del problema	C	S	+	100
4	Diagnostico	Descripción del problema	C	S	+	500
5	Solucion	Solución del problema	C	S	+	500

6	Host	Nombre del servidor o host	C	S	+	15
*1 A=ARCHIVO / T=TABLA / R=REGISTRO / C=CAMPO *2 K=CLAVE / M=MULTIVALUADO / S=LINEAL / P=FRASE *3 D=FECHA / N=NUMERICO / A=ALFABETICO / +=ALFANUMERICO						

FUENTE: La autora

4.1.13 DIAGRAMA DE COMUNICACIÓN DE DATOS

Gráfico # 62. RED DE DATOS



FUENTE: La autora

4.1.14 REQUERIMIENTOS DE HARDWARE Y SOFTWARE

Tabla # 20. REQUERIMIENTOS DE HARDWARE

EQUIPO	CARACTERÍSTICAS
Dos computadores	Procesador 200 Mhz o superior Memoria RAM 64 MB Disco duro 64 MB Monitor Resolución VGA o superior Unidad de CD, teclado, mouse, tarjeta de red, impresora, cortapicos, regulador de voltaje, cables de conexión para red..

FUENTE: La autora

Tabla # 21. REQUERIMIENTOS DE SOFTWARE

PROGRAMAS	TIPO
Red Hat Linux 7.2	Sistema Operativo
Interbase 6.0	Base de datos
Php-4.3.1	Lenguaje de Programación
Dreamweaver 4 y editor gvim de Linux	Editores de páginas web
Flash MX	Editor gráfico de páginas web
Advanced Apache Web Server	Conectividad
Nestcape Navigator, Internet Explorer.	Navegadores

FUENTE: La autora

4.1.15 ESTIMACIONES SOBRE ARCHIVOS

Tabla # 22. ESTIMACIÓN DE ARCHIVOS

ARCHIVO	ESPACIO ASIGNADO (Bytes)
/usr/local/apache/htdocs/proyecto/botonindex.swf	4 KB
/usr/local/apache/htdocs/proyecto/botonaceptamenu.swf	4 KB
/usr/local/apache/htdocs/proyecto/botonregresaregistro.swf	8 KB
/usr/local/apache/htdocs/proyecto/botonregresavitacora.swf	8 KB
/usr/local/apache/htdocs/proyecto/botonvalidarotro.swf	8 KB
/usr/local/apache/htdocs/proyecto/botonmenu1swf	8 KB
/usr/local/apache/htdocs/proyecto/botonmenu2.swf	8 KB
/usr/local/apache/htdocs/proyecto/botonmenu3.swf	8 KB
/usr/local/apache/htdocs/proyecto/botonmenu4.swf	8 KB
/usr/local/apache/htdocs/proyecto/botonmenu5.swf	8 KB
/usr/local/apache/htdocs/proyecto/conectar.php	1 KB
/usr/local/apache/htdocs/proyecto/fondo.gif	3 KB
/usr/local/apache/htdocs/proyecto/logo_Petro.gif	14 KB
/usr/local/apache/htdocs/proyecto/Sello_UTC.gif	4 KB
/usr/local/apache/htdocs/proyecto/letraspetro.swf	8 KB
/usr/local/apache/htdocs/proyecto/letraspetro.fla	20 KB

/usr/local/apache/htdocs/proyecto/index.html	4 KB
/usr/local/apache/htdocs/proyecto/validar.html	4 KB
/usr/local/apache/htdocs/proyecto/graba_validar.html	4 KB
/usr/local/apache/htdocs/proyecto/menu.html	6 KB
/usr/local/apache/htdocs/proyecto/consulta.html	8 KB
/usr/local/apache/htdocs/proyecto/registro.html	7 KB
/usr/local/apache/htdocs/proyecto/graba_registro.html	5 KB
/usr/local/apache/htdocs/proyecto/vitacora.html	6 KB
/usr/local/apache/htdocs/proyecto/graba_bitácora.html	5 KB
/usr/local/apache/htdocs/proyecto/phpinfo.html	1 KB
/usr/local/apache/htdocs/proyecto/a.gif	6 KB
/usr/local/apache/htdocs/proyecto/b.gif	7 KB
/usr/local/apache/htdocs/proyecto/c.gif	6 KB
/usr/local/apache/htdocs/proyecto/d.gif	6 KB
/usr/local/apache/htdocs/proyecto/e.gif	6 KB
/usr/local/apache/htdocs/proyecto/f.gif	6 KB
/usr/local/apache/htdocs/proyecto/g.gif	7 KB
/usr/local/apache/htdocs/proyecto/h.gif	7 KB
/usr/local/apache/htdocs/proyecto/i.gif	4 KB
/usr/local/apache/htdocs/proyecto/j.gif	4 KB
/usr/local/apache/htdocs/proyecto/k.gif	7 KB

/usr/local/apache/htdocs/proyecto/l.gif	5 KB
/usr/local/apache/htdocs/proyecto/m.gif	9 KB
/usr/local/apache/htdocs/proyecto/n.gif	7 KB
/usr/local/apache/htdocs/proyecto/o.gif	7 KB
/usr/local/apache/htdocs/proyecto/p.gif	7 KB
/usr/local/apache/htdocs/proyecto/q.gif	6 KB
/usr/local/apache/htdocs/proyecto/r.gif	8 KB
/usr/local/apache/htdocs/proyecto/s.gif	6 KB
/usr/local/apache/htdocs/proyecto/t.gif	6 KB
/usr/local/apache/htdocs/proyecto/u.gif	7 KB
/usr/local/apache/htdocs/proyecto/v.gif	6 KB
/usr/local/apache/htdocs/proyecto/w.gif	7 KB
/usr/local/apache/htdocs/proyecto/x.gif	7 KB
/usr/local/apache/htdocs/proyecto/y.gif	6 KB
/usr/local/apache/htdocs/proyecto/z.gif	7 KB

FUENTE: La autora

4.1.16 ETAPA DE CONSTRUCCIÓN

Tabla # 23. MENSAJES DE ERROR DEL SISTEMA

CODIGO ERROR	**TEXTO DEL MENSAJE**	**CAUSA DEL ERROR**	**ACCION DE CORRECCION**
Window.alert	Incorrecto	Utilización de parámetros incorrectos	Debe ingresar el nombre del usuario

		en el ingreso del usuario y la clave	y su clave correctamente.
Window.alert	Incorrecto el usuario	Utilización de parámetros incorrectos en el ingreso del usuario	Debe ingresar el nombre del usuario correctamente.
Window.alert	Incorrecto la clave	Utilización de parámetros incorrectos en el ingreso de la clave	Debe ingresar la clave del usuario correctamente.
Window.alert	No ha validado un usuario	No ha ingresado ningún usuario, ni clave para validar.	Debe ingresar un usuario y una clave para validar.
window.alert	Este usuario ya existe, ingrese otro nombre	Repetición del mismo nombre de usuario.	Ingresar otro nombre de usuario
window.alert	Estos datos no son válidos, debe ingresar un nombre de usuario	Utilización de datos incorrectos o espacios en blanco.	Debe ingresar el nombre del usuario y sus datos correctamente.
Window.alert	Este usuario ya ha sido registrado, escoja otro nombre	Repetición del mismo nombre de usuario en el registro de vitacora.	Elija otro nombre de usuario disponible en la lista.
Window.alert	Datos no válidos. Seleccione otro usuario	Repetición de un usuario o espacios en blanco.	Escoja otro usuario disponible en la lista.

FUENTE: La autora

4.1.17 ETAPA DE IMPLANTACIÓN

Planes:

- Capacitar al usuario para el uso de la aplicación web validación de usuario y clave.
- Indicar al usuario las ventajas y opciones que presta la aplicación web.

- Esta aplicación tiene la posibilidad de realizar adecuaciones, según los requerimientos del usuario.

Control:

- Esta aplicación posee un manual del usuario en formato pdf e impreso en papel para consultas de su funcionamiento.
- Se prestará ayuda al usuario cada vez que lo necesite.

4.2 EJECUCIÓN DE LA APLICACIÓN VALIDACIÓN DE USUARIO Y CONTRASEÑA

4.2.1 Introducción

La presente aplicación web ha sido desarrollada con la finalidad de comprobar el funcionamiento del servidor web SSL, el cual ha sido instalado y configurado completamente. Esta aplicación valida el nombre y clave de los usuarios que se encuentran en una base de datos realizada en Interbase.

Además esta aplicación permite registrar nuevos usuarios en la base de datos, consultar los usuarios existentes y realizar el correspondiente registro de bitácora a los usuarios que se han registrado con la categoría de usuarios de tercer nivel.

4.2.2 Objetivos de la aplicación

- Validar el nombre y clave de los usuarios de la base de datos elaborada en Interbase.
- Permitir al usuario registrar nuevos usuarios en la base de datos.
- Facilitar al usuario hacer consultas de usuarios de la base de datos.
- Realizar un registro de bitácora con los usuarios de tercer nivel de la base de datos.

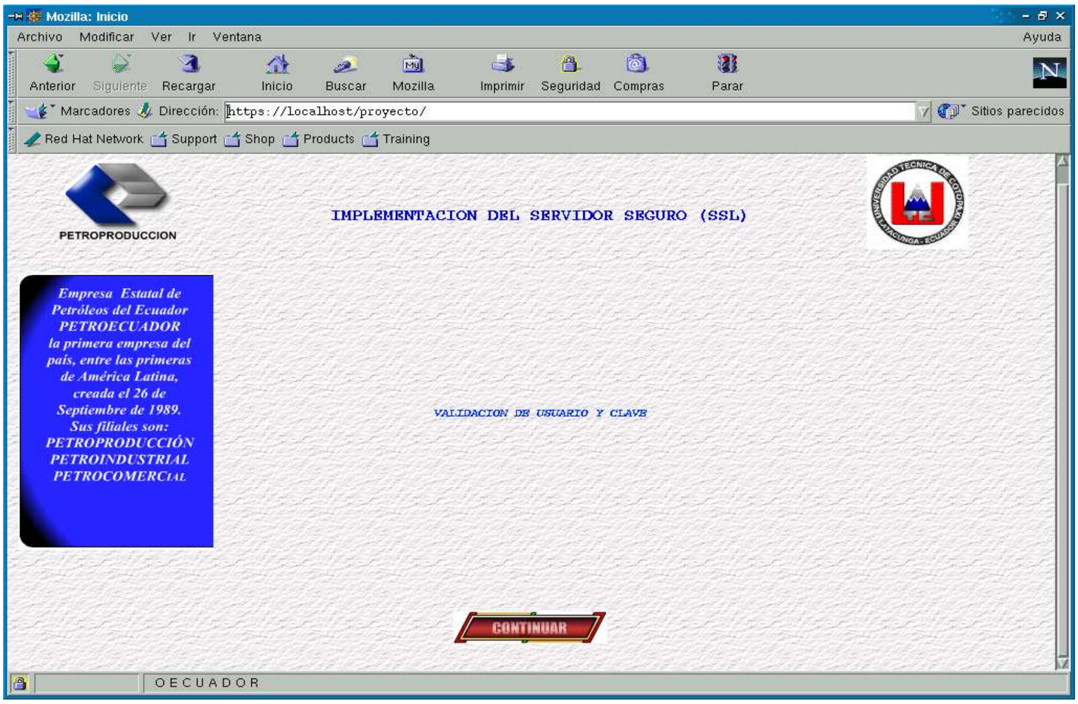
4.2.3 Ingreso a la aplicación

Para iniciar la ejecución de la aplicación se debe los siguientes pasos:

1. Ingresar al directorio /opt/interbase/bin para levantar el Interbase 6.0 con la siguiente línea de código: ./ibmgr –start –forever
2. Ingresar al directorio /usr/local/apache/bin para levantar el Apache con la siguiente línea de código: ./apachectl startssl
3. Ingresar la contraseña con la que se configuró el servidor.
4. Ubicarse con el mouse en la barra de herramientas para escoger la opción K, luego Internet y finalmente Nestcape Navigator. En la ventana del navegador digitar la siguiente línea: https://localhost/proyecto/ presionar enter y aceptar

todos los datos del certificado, los cuales indican que se va visualizar la aplicación con una conexión segura. Se observará la primera página index.html.

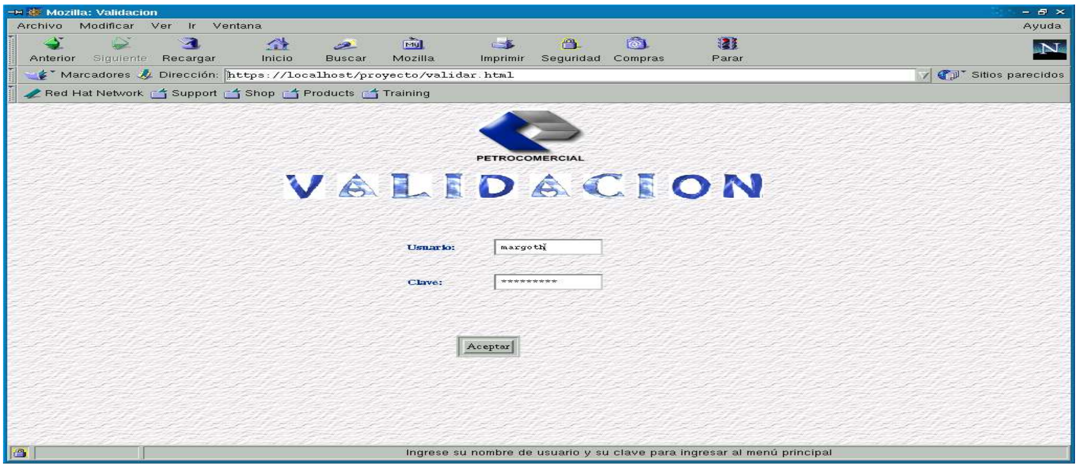
Gráfico # 63. PÁGINA INDEX.HTML



FUENTE: La autora

5. Se debe presionar el botón continuar que se localiza en la parte inferior de la página, el cual permitirá visualizar la página validar.html.

Gráfico # 64. PÁGINA VALIDAR.HTML



FUENTE: La autora

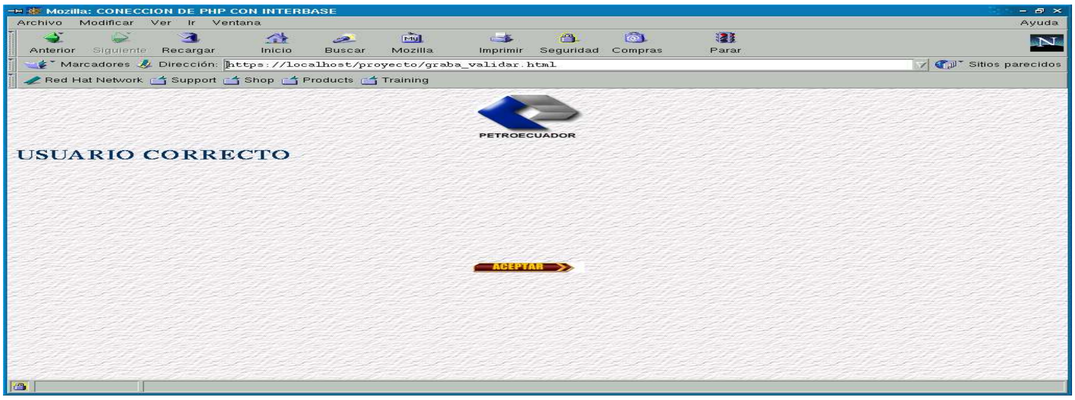
6. En esta página se debe ingresar el nombre y la clave del usuario que se desea validar, posteriormente presionar el botón aceptar. Si los datos que se ingresaron no fueron correctos aparecerá un mensaje indicando que esos datos son incorrectos. Presionar el botón Aceptar para regresar a la página anterior e ingresar nuevamente el nombre y clave del usuario.

Gráfico # 65. MENSAJE DE ERROR DE VALIDACIÓN

FUENTE: La autora

7. Cuando se ha ingresado los datos correctos aparecerá la página graba_validar.html la cual indica que los datos fueron validados correctamente.

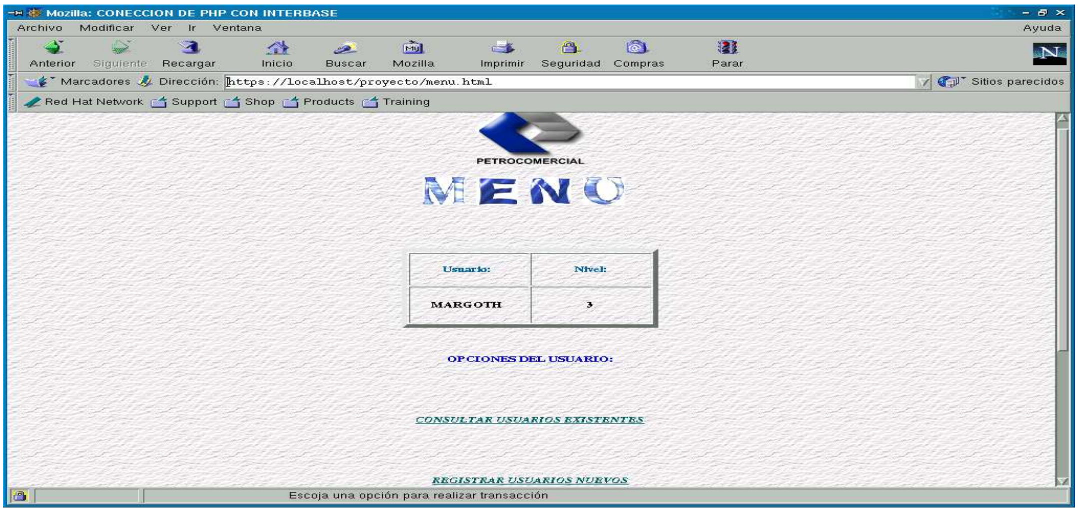
Gráfico # 66. PÁGINA GRABA_VALIDAR.HTML



FUENTE: La autora

8. Posteriormente presionar el botón aceptar para ingresar a la página menu.html.

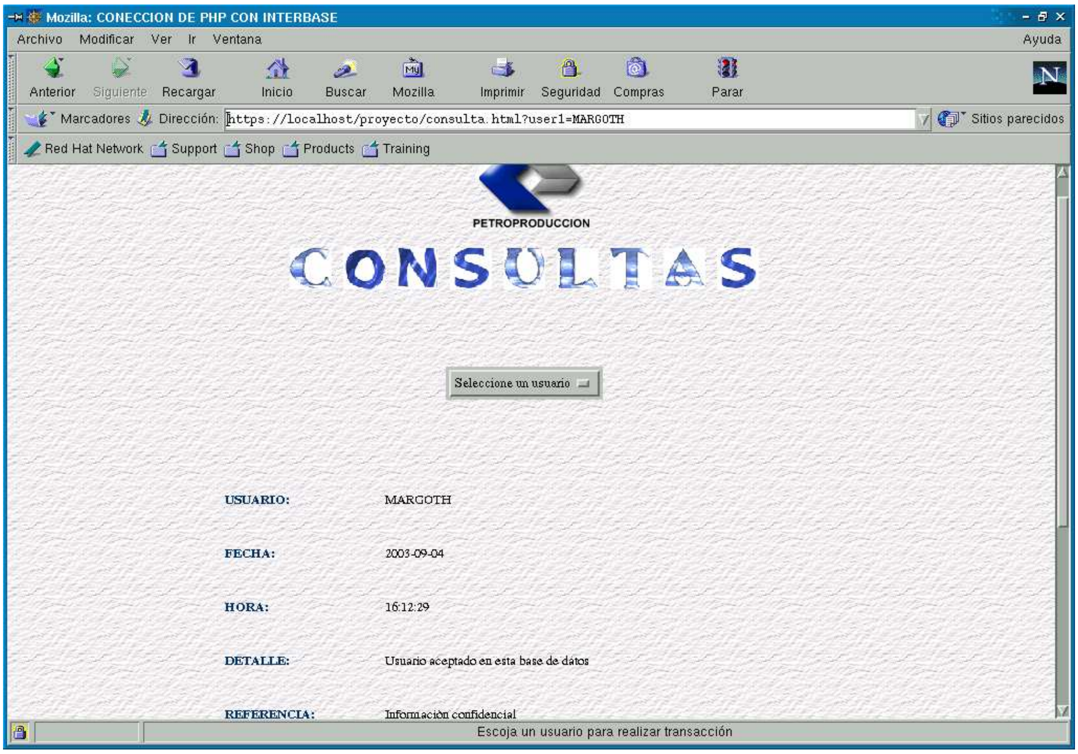
Gráfico # 67. PÁGINA MENU.HTML



FUENTE: La autora

9. En esta página en la parte superior indica el nombre del usuario el usuario y el nivel, cabe destacar que existen tres niveles: 1) el usuario solo tiene la opción de realizar consultas, 2) el usuario tiene las opciones de consultar y registrar nuevos usuarios, 3) el usuario tiene las tres opciones existentes de consultar, registrar nuevos usuarios e ingresar en la base de datos el registro de bitácora que no es más que los problemas, diagnóstico y soluciones a los equipos de cómputo dadas por el usuario. Si el usuario pertenece al nivel 1 tiene solo la opción de consultas como se muestra en el gráfico # 68.

Gráfico # 68. PÁGINA CONSULTA.HTML

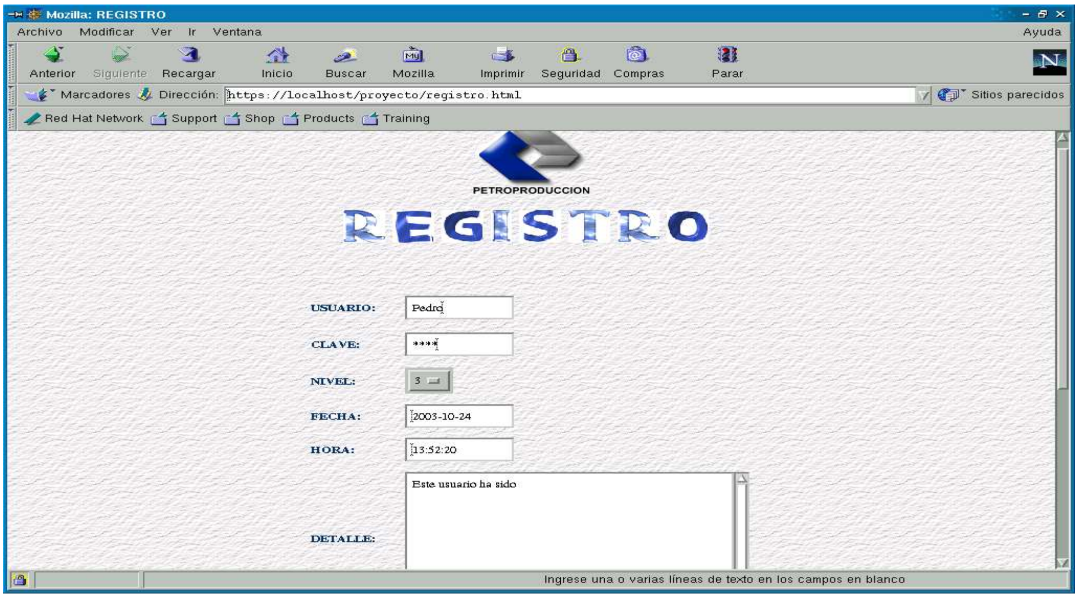


FUENTE: La autora

En esta página el usuario puede escoger del combo un nombre de todos los que se encuentran disponibles y observar sus datos de información.

10. Luego de realizar las consultas presionar el botón IR AL MENU para regresar al menú principal y escoger otra opción en caso de que el usuario lo disponga. Si el usuario es de nivel 2 también podrá acceder a la página de registro de nuevos usuarios como se indica en el gráfico # 69.

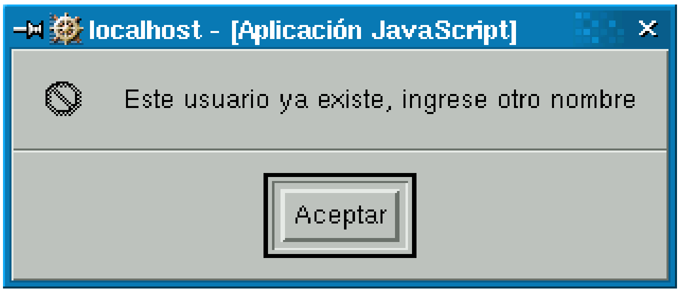
Gráfico # 69. PÁGINA REGISTRO.HTML



FUENTE: La autora

En esta página el usuario tiene la capacidad de ingresar nuevos usuarios en la base de datos. Además no puede ingresar nombres de usuarios repetidos o espacios en blanco, porque no se grabará en la base y se le presentará la siguiente página de error:

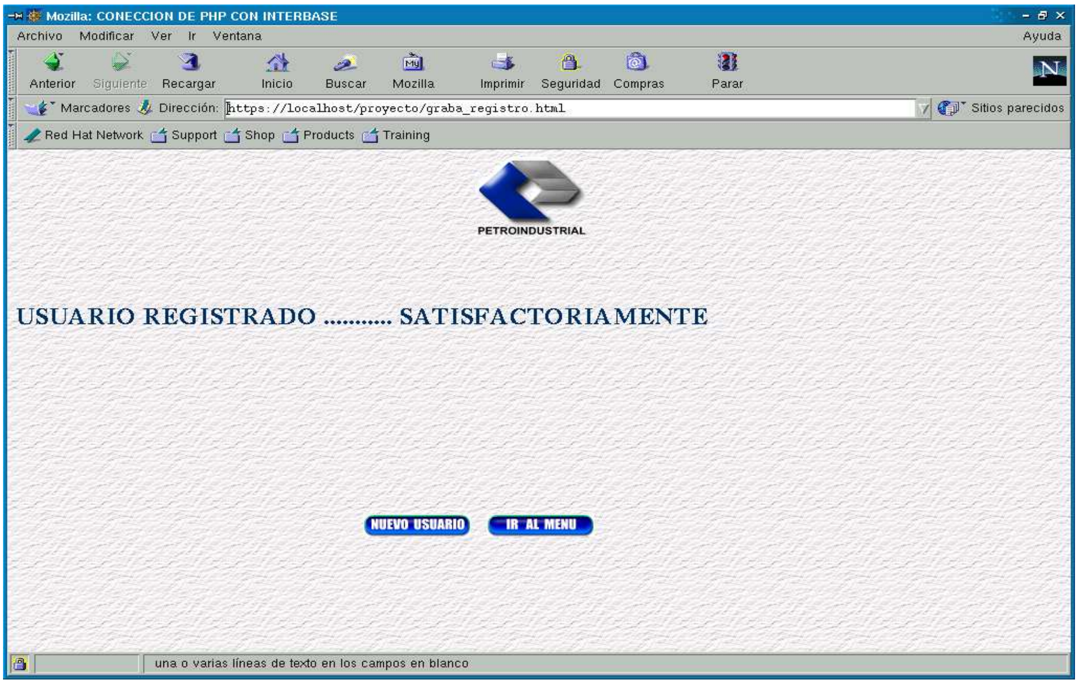
Gráfico # 70. MENSAJE DE ERROR DE REGISTRO



FUENTE: La autora

Si el nombre del usuario no se repite y los datos ingresados son correctos entonces se le presentará la página del gráfico # 71.

Gráfico # 71. PÁGINA GRABA_REGISTRO.HTML

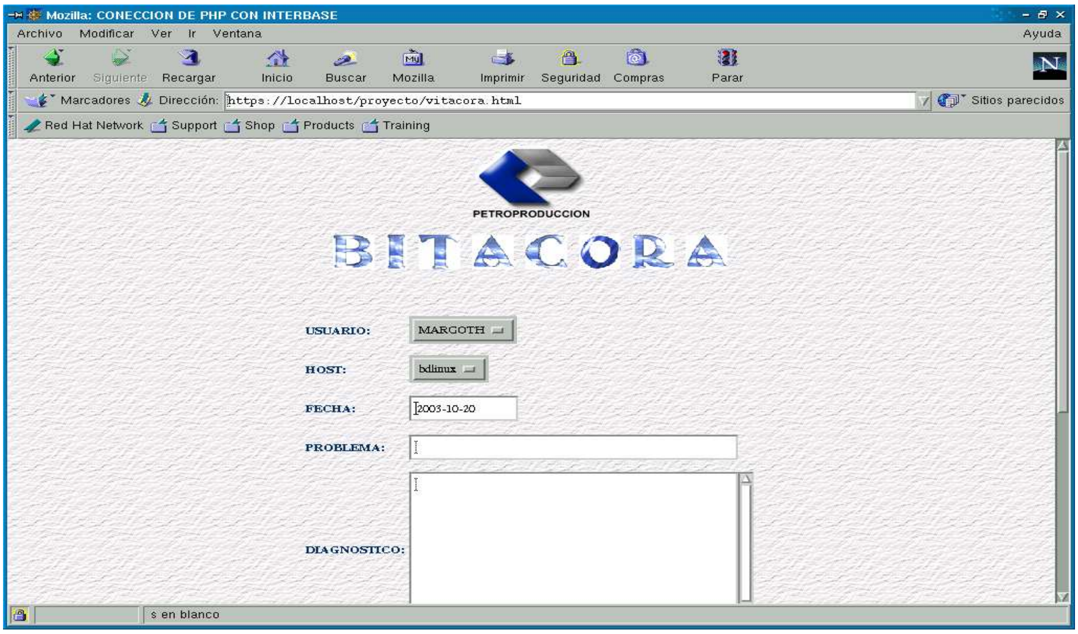


FUENTE: La autora

Si se desea ingresar otro usuario presionar el botón NUEVO USUARIO de lo contrario presionar el botón IR AL MENU para regresar al menú principal.

11. Si el usuario pertenece al nivel 3 tendrá la opción de registro de bitácora a parte de las dos anteriores. En la página bitácora.html se le presentará en un combo solo los usuarios de nivel 3 registrados en el registro de nuevos usuarios, en el cual se puede escoger y llenar los demás campos en blanco.

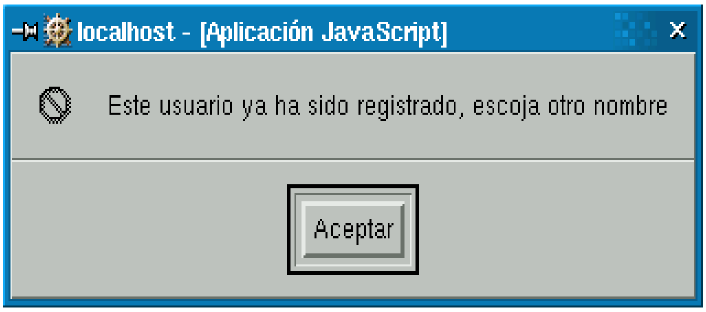
Gráfico # 72. PÁGINA BITACORA.HTML



FUENTE: La autora

En caso de que el usuario escogido ha sido registrado anteriormente se le presentará el gráfico # 73.

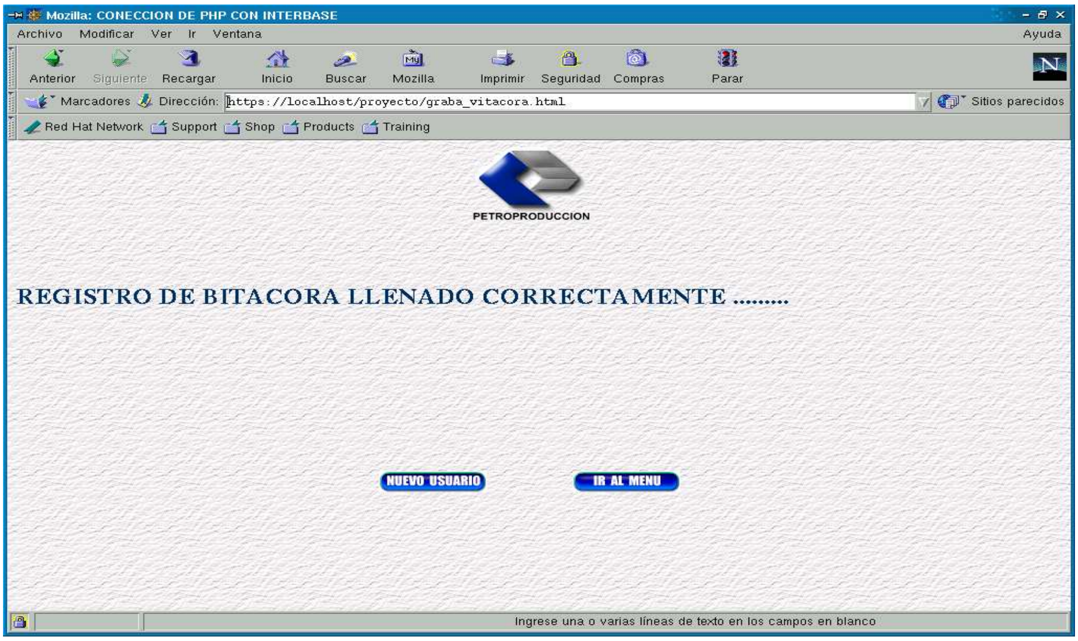
Gráfico # 73. MENSAJE DE ERROR DE REGISTRO DE BITACORA



FUENTE: La autora

Si el nombre del usuario escogido no ha sido registrado se le presentará la página del gráfico # 74.

Gráfico # 74. PÁGINA GRABA_BITACORA.HTML



FUENTE: La autora

Si se desea registrar otro usuario presionar el botón NUEVO USUARIO de lo contrario presionar el botón IR AL MENU para regresar al menú principal.

12. Si desea validar otro usuario presionar el botón VALIDAR OTRO localizado en la parte inferior de la página menu.html y se regresará a la página validar.html.

4.2.4 Conclusiones de la aplicación

- A través de la presente aplicación se comprueba el funcionamiento del servidor web SSL y el usuario tiene la posibilidad de validar el nombre y la clave de los usuarios, ingresar usuarios nuevos, realizar consultas de los mismos y elaborar el correspondiente registro de bitácora con los usuarios que poseen el tercer nivel o máximo los cuales se encuentran en la base de datos de Interbase.

- Por medio de la descripción del funcionamiento de la aplicación el usuario tiene la posibilidad de consultar cada vez que lo necesite ya que se ha detallado paso a paso la ejecución de la misma y con gráficos ilustrativos.[23]

23 https://localhost/ ; Ultimo acceso: Sábado 27 de Septiembre del 2003

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 VERIFICACION DE OBJETIVOS

1. A través del desarrollo de este proyecto se ha logrado solucionar el problema de seguridad de la información que se encuentra en el servidor web de PETROECUADOR ya que con la implementación de este tipo de seguridad la información que se transmite por la red estará encriptada y protegida obteniendo integridad, confidencialidad y autenticidad de los datos.

2. Con la culminación de este proyecto se ha llegado a cumplir satisfactoriamente con el objetivo general, el cual se enmarcó en el desarrollo de cada uno de los objetivos específicos solucionados durante el tiempo estimado y bajo el control del administrador de redes de Linux de PETROECUADOR; de tal manera que el proyecto se visualiza en el sitio adquirido por la empresa.

5.2 CONCLUSIONES

- Con el desarrollo de la presente aplicación validación de usuario y clave se comprueba el funcionamiento del servidor web SSL, el mismo que permite utilizar el puerto 443 para la transmisión segura de la información, así como visualizar el correspondiente certificado digital del servidor. Además en la actualidad el entorno de desarrollo de aplicaciones web y la tecnología de la información (TI) ha marcado el rumbo en las relaciones económicas; Internet y otras formas de comunicación e información están cambiando las maneras de trabajar, aprender, comunicarse y hacer negocios, se dice que la TI elimina fronteras ofreciendo a empresarios, proveedores y clientes en todo el mundo el contacto a través de un solo clic.

- El protocolo SSL es utilizado en el Comercio Electrónico por su gran seguridad en él envio de información, siendo una herramienta principal para la encriptación del mensaje por claves públicas / privadas para la conexión de redes, su sistema de certificado y autentificación permite un logro en el ámbito de empresas para que no exista interferencia de terceros que quieran perjudicar el éxito de sus tiendas virtuales en Internet. Con SSL, se podrá gestionar las compras en línea de forma segura, con autenticación, confidencialidad de los datos intercambiados e integridad en los mismos y como consecuencia se reduce el riesgo, se aumenta la confianza de los clientes y se gana competitividad.

- Apache es en la actualidad el principal servidor de web. Es el más rápido, eficiente y el que evoluciona a mayor velocidad. Apache por su naturaleza de software abierto, es ideal para instalar en máquinas GNU/Linux que aseguran un Sistema Operativo con unas comunicaciones excelentes. Apache ha ayudado a que el campo de GNU/Linux se amplíe de forma muy sólida en el mundo Internet, creando una Internet-box que difícilmente puede ser superada por otra plataforma en los sistemas actuales, tanto en costo como en potencia.

5.3 RECOMENDACIONES

- Se recomienda como primer paso y principal levantar el servidor Apache y el servidor se base de datos de Interbase para la ejecución del servidor web SSL con su correspondiente aplicación validación de usuario y clave, caso contrario generará errores.
- Cuando se realiza la configuración al momento de ingresar las claves debe utilizar combinaciones de letras mayúsculas, minúsculas y números de tal manera que sean difíciles de deducir. No use fechas de cumpleaños, nombres comunes o de parientes ya que son descubiertas fácilmente.

- Cuando se desarrollan páginas web para ser publicadas se debe poner a disposición de los clientes una página sobre la política de privacidad, explicando que se hace con los datos privados. Además ser claro en la especificación de los productos, añadir imágenes y proporcionar siempre información detallada acerca de precios, condiciones de garantía, forma de envío, formas para pagar, como tarjeta de crédito, contra reembolso, etc.

- Cuando se va a comprar un certificado digital a una empresa certificadora se debe seleccionar varias empresas y de allí escoger la que ofrezca mayores políticas de seguridad y comprobando los datos que aparecen en los certificados.

- Nunca se debe entregar datos confidenciales como número de tarjeta de crédito por correo electrónico si no es a través de un servidor seguro que utilice sistemas de cifrado que garantizan la confidencialidad de la comunicación.

BIBLIOGRAFIA

LIBROS:

Arq. Ulloa Francisco, Investigación 2000.

Pressman Roger S., Ingeniería de Software "Un enfoque práctico", Cuarta Edición.

PÁGINAS WEB:

BoNd, página oficial del canal #Ayuda_IRC. "Criptografía simétrica y antisimétrica" Revisado por BoNd (16/07/2001) [http://www.ayuda-irc.net/pgp.shtml#introduccion-cripto], Viernes, 13 Diciembre 2002

El portal de los Ingenieros en Informática Copyright 2000-2001-2002 - ingenieroseninformatica.org - Todos los derechos reservados. " Autentificación e integridad (firma digital) " [http://ingenieroseninformatica.org/recursos/tutoriales/firmaelectronica/cap2.php], Viernes, 10 de Enero del 2003

Copyright © 1999, Kurt Seifried, José Antonio Revilla, "[Guía de Seguridad del Administrador de Linux - GSAL] v.19991128" [http://www1.tiendalinux.com:81/documentacion/manuales/gsal/gsal-19991128-htm/http.htm] miércoles, 4 de diciembre del 2002.

Ricardo Villafaña Figueroa, Universidad de las Américas-Puebla-México "Tecnología de la Información", [http://mailweb.udlap.mx/~rvillafa/Impacto%20TI.htm], Jueves, 2 de enero de 2003.

GNU Privacy Guard, "Firma Digital" [http://linuxcol.uniandes.edu.co/~gramo/comos_gramo/UMAN.2001-1/crypt/Conceptos_Basicos.html], Lunes 20 de Enero del 2003.

Seguridad Básica en el Sistema Linux, " Seguridad y Paswords" [http://emain.port5.com/ponencias/Seguridad_Basica_en_Linux/seguridad-3.html], Lunes 23 de Diciembre del 2002.

Marc Vaquer Crusat, "Que es Apache SSL " [http://www.arrakis.es/~qenda/Articles/ArticleCONSERVER/CONECTIVALINUXSERVER.htm], Martes 7 de Enero del 2003.

Calle 22.com, "Introducción al e-business y al comercio electrónico (e-commerce):" [http://comunidades.calle22.com/comunidades/500/com500con2.asp], Jueves 12 de Diciembre del 2002.

Geocities.com, “Qué es SSL” [http://www.geocities.com/juankox/une/decimo/ecommerce/SSL.doc], Miércoles 18 de Diciembre del 2002.

Daniel Moreno Gamboa, Politécnico Grancolombiano, Facultad de Sistemas materia de énfasis 4 Bogotá 16 de abril de 2001, “SSL SECURE SOCKET LAYER” [http://sigma.poligran.edu.co/politecnico/apoyo/sistemas/dist/docs_011/ssl.doc] Jueves, 16 de enero de 2003.

José de Jesús Angel, “Criptografía Para Principiantes” Versión 1.0 http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html, Martes 5 de Agosto del 2003

Quintana Martel Marcos “Criptografía” http://nave.escomposlinux.org/productos/mozilla/1.0.1/progreso/Platform_neutral/help/glossary.html, Jueves 5 de Junio del 2003 José de Jesús Angel Angel

Miguel García Feal y José Luis Chouciño Ferreiro “Seguridad en Internet SSL” http://www.seguridad%20SSL.html, Viernes 11 de Julio del 2003

Tim J. Hudson, SSLeay F.A.Q "Ralf Engelschall ha realizado un excelente módulo que integra Apache y SSLeay" http://www.modssl.org/docs/2.8/, Jueves 7 de agosto del 2003

"Metodologías Informáticas · Arquitectura cliente / servidor" http://www.inei.gob.pe/cpi-mapa/bancopub/libfree/lib616/CAP0312.HTML, Martes 15 de Julio del 2003

Instituto de Investigaciones Biomédicas, Alberto Cols, "Software de base - Cliente Servidor" http://www2.iib.uam.es/bioinfo/curso/perl/mw/soft_de_base.es.html, Martes 22 de Julio del 2003

Oscar Javier Prieto Izquierdo, "Introducción a las aplicaciones web", http://www.zope.org, Viernes 8 de Agosto del 2003

"Paradigma de Construcción de Prototipos" http://www.paradigmadeconstrucciondeprototipos.html, Viernes 8 de Agosto del 2003

Alina Castellanos Leyva, "Características generales de Linux" http://www.linux.cu/manual/basico-html/footnode.html, Lunes 4 de Agosto del 2003

Álvaro del Castillo San Félix Desarrollador y admin de software libre Barrapunto, " El servidor de web Apache: Introducción práctica", Apache 1.x y 2.0 alpha http://barrapunto.com, Lunes 4 de Agosto del 2003

Adaptation By Mike Rochette Bytewise Inc. "How to install Mod_ssl, Php4, and Kylix modules as Apache Dso", http://www.cambuddys.com/phpssldso.html, Viernes 29 de Agosto del 2003

Adaptation By Mike Rochette Bytewise Inc. "Index", http://www.cambuddys.com/index.php, Viernes 29 de Agosto del 2003

Egda. Caisaguano Ch. Anita Margoth "Tesis Implementación de un servidor web SSL", https://localhost/, Martes 2 de Septiembre del 2003

Sitio de la Autoridad Certificadora Verising, "Comprando un certificado desde Verisign", http://europe.redhat.com/documentation/rh/7/ref-guide-es/s1-securing-buycert.php3/, Martes 9 de Septiembre del 2003