ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"IMPLEMENTACIÓN DE UN WEBSITE DE COMERCIO ELECTRÓNICO UTILIZANDO UNA INFRAESTRUCTURA DE RED SEGURA: PORTAL PARA REALIZAR PAGOS DE SERVICIOS BÁSICOS ON-LINE, CON SOPORTE PARA MEDIOS DE PAGO ELECTRÓNICOS"

TÓPICO DE GRADUACIÓN

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

Presentado por:

Verónica Alvarado Torres
Julia Espinoza León
Carlos Julio Sánchez

GUAYAQUIL – ECUADOR

2005

AGRADECIMIENTO

A todas las personas que colaboraron en el desarrollo del presente trabajo: familiares, amigos, profesores y nuestra directora de tesis Ing. Karina Astudillo.

DEDICATORIA

A Dios, por permitirme contar con el apoyo de mi familia, en todo momento.

Verónica Alvarado T.

DEDICATORIA

A Dios.

A mi familia.

Julia Espinoza L.

DEDICATORIA

A Dios.

A mis padres.

A mi hermano.

Carlos Julio Sánchez R.

TRIBUNAL DE GRADUACIÓN

Ing. Miguel Yapur A.
SUB-DECANO DE LA FIEC
PRESIDENTE

Ing. Karina Astudillo.
DIRECTORA DE TESIS

Ing. Cristina Abad R.
MIEMBRO PRINCIPAL

Ing. Fabricio Echeverría B.
MIEMBRO PRINCIPAL

Escaneado con CamScanner

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de Graduación de la ESPOL)

Verónica Alvarado Torres.

Julia Espinoza León

Carlos Julio Sánchez.

Escaneado con CamScanner

RESUMEN

En la actualidad el sistema de pagos de las planillas de agua, luz y teléfono es realizado acercándose a una de las oficinas de la institución que brinda el servicio básico o en alguna de las instituciones que permiten realizar estos pagos.

El pago on-line en el Ecuador no es una alternativa muy utilizada, en parte por la poca publicidad de los portales Web y por la dificultad del proceso de pago ocasionado por la cantidad de pasos que se deben seguir para poder realizarlo, además la mayor parte del contenido de estos sitios Web no hace referencia solo al pago de planillas de servicios básicos sino que provee de otra información que tiende a confundir al usuario.

Este documento describe los procesos de Análisis, Diseño e Implementación de un **PORTAL PARA REALIZAR PAGOS DE SERVICIOS BÁSICOS *ON-LINE*, CON SOPORTE PARA MEDIOS DE PAGO ELECTRÓNICOS,** este sitio Web estará enfocado a satisfacer las necesidades del usuario ofreciendo un sistema de cancelaciones mucho más flexible debido a las diferentes formas de pago como son: Tarjetas de Crédito y Tarjetas de Débito de cualquier institución Financiera; así mismo proveerá la facilidad de

consultas de Cuentas y fecha de vencimiento de la última planilla de pago a realizar.

El primer capítulo explica las diferentes formas en que son realizados actualmente los pagos de servicios básicos en el Ecuador. Este capítulo también presenta una justificación de nuestro proyecto como alternativa para realizar pagos de planillas de servicios básicos, así como sus objetivos, restricciones, limitaciones, alcance y brevemente se describen a quienes está orientado el sistema. Además indicaremos a qué tipo de usuarios está dirigido este sistema.

El capítulo 2 explica sobre los servicios que ofrece el portal como son consultas y pagos. También se hace un análisis de la competencia, el segmento de mercado al cual está enfocado el portal y la estrategia de marketing empleada.

El capítulo 3 describe los requerimientos funcionales y no funcionales de la aplicación, también muestra el modelo conceptual que describe el dominio del problema y ayuda a obtener una idea más clara del mismo. Además realiza un análisis técnico del hardware y software, el mismo que permitirá seleccionar una mejor herramienta sobre otra, considerando ciertas especificaciones como: rendimiento, facilidad de aprendizaje, costo.

Igualmente describimos ciertos aspectos de seguridad que deben ser considerados en el desarrollo de aplicaciones Web.

El capítulo 4 describe la arquitectura física del sistema, la definición y diagramas de los diferentes casos de uso que describen los diferentes procesos del sistema. Especificaremos también un diagrama de secuencias que darán una representación gráfica de los eventos que fluyen de los actores al sistema. Conjuntamente describimos cada una de las tablas que se encuentren en el diagrama entidad relación de la Base de datos.

El capítulo 5 describe los componentes de la aplicación, así mismo se explicará paso a paso los procesos de implementación e instalación de los recursos usados en el Sistema.

Por último, daremos las conclusiones a las que hemos llegado al finalizar el desarrollo de la Tesis y recomendaciones basadas en la experiencia obtenida en el proceso de implementación del Sistema.

ÍNDICE GENERAL

Pág.

AGRADECIMIENTO……..…….....II

DEDICATORIA……..…….....III

TRIBUNAL DE GRADUACIÓN……..VI

DECLARACIÓN EXPRESA……..……VII

RESUMEN……..……VIII

ÍNDICE GENERAL……..……..……..XI

ÍNDICE DE FIGURAS……..……...XV

ÍNDICE DE TABLAS……..……...XVII

ÍNDICE DE ABREVIATURAS……..……..….XVIII

CAPITULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN……1

1.1 Pagos de Servicios Básicos en el Ecuador ……..….3

1.2 Justificación y objetivos ……..……..……..……7

1.3 Usuarios del sistema …….…….....10

1.4 Limitaciones y restricciones ……..……..…..……11

1.5 Alcance ……..……..…..……13

CAPITULO 2

2. ANÁLISIS DE MERCADO……...……..15

2.1 Servicios ofrecidos por el portal de pagos …….....16

2.2 Análisis de la competencia……..……..……..……18

2.3 Segmento de mercado ……..……..20

2.4 Estrategia de marketing ……..…22

CAPITULO 3

3. ANÁLISIS DEL SISTEMA……...26

3.1 Requerimientos..……..……..…….....27

3.2 Modelo conceptual……..……..……34

3.3 Análisis técnico ……..……..37

3.4 Análisis de seguridad …….……85

CAPITULO 4

4. DISEÑO DEL SISTEMA……..89

4.1 Arquitectura del sistema ……..……90

4.2 Casos de uso ……..……..……95

4.3 Diagramas de secuencia ……..……..……111

4.4 Diseño de base de datos ……...……114

CAPITULO 5

5. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA……127

5.1 Proceso de implementación …….....128

5.2 Instalación de herramientas de software …….…149

5.3 Autenticación …….…....149

CONCLUSIONES ……..……160

RECOMENDACIONES..……...……164

APÉNDICE A RESULTADO DE ENCUESTAS SOBRE PAGOS DE SERVICIOS BÁSICOS…….... ..169

APÉNDICE B CONCEPTOS SOBRE SEGURIDAD INFORMÁTICA……..178

APÉNDICE C DIAGRAMAS DE SECUENCIA……..198

APÉNDICE D INSTALACIÓN DE HERRAMIENTAS DE SOFTWARE..…212

BIBLIOGRAFÍA……..328

ÍNDICE DE FIGURAS

Pág.

FIGURA 3.1 Modelo Conceptual …….……. ...36
FIGURA 3.2 Comparación de un navegador que usa Java con un navegador convencional…….....53
FIGURA 3.3 Típica disposición de un firewall doméstico……..……. ...71
FIGURA 3.4 Modos Túnel y Transporte…….. ...76
FIGURA 4.1 Capas de la Arquitectura del sistema…….....91
FIGURA 4.2 Arquitectura del Sistema…….....92
FIGURA 4.3 Diagrama Físico de la red…….. ...94
FIGURA 4.4 Símbolo utilizado para representar un caso de uso…… ...95
FIGURA 4.5 Símbolo utilizado para representar un actor……... .. 96
FIGURA 4.6 Símbolo utilizado para la frontera del sistema…… ...96
FIGURA 4.7 Caso de uso Iniciar sesión…….. ...98
FIGURA 4.8 Caso de uso Registrar usuario…….. ...99
FIGURA 4.9 Caso de uso Consultar planilla……...101
FIGURA 4.10 Caso de uso Pagar servicio básico…….. .104
FIGURA 4.11 Visualizar planilla…….. .107
FIGURA 4.12 Salir del sistema……....110
FIGURA 4.13 Símbolo utilizado para representar la línea de vida de un objeto en un diagrama de secuencia……... .112
FIGURA 4.14 Símbolo utilizado para representar el tiempo de vida de un objeto en un diagrama de secuencia……... .112
FIGURA 4.15 Símbolo utilizado para representar los mensajes entre objetos en un diagrama de secuencia…… .113
FIGURA 4.16 Diagrama entidad-relación de la base de datos…… .126
FIGURA 5.1 Asignación de direcciones IP y gateway a los equipos…….… .130
FIGURA 5.2 Paquete com.pagosaldia.servlets y sus clases…….… .138
FIGURA 5.3 Paquete com.pagosaldia.tools y sus clases……139
FIGURA 5.4 Paquete com.pagosaldia.beans y sus clases…… .140
FIGURA 5.5 Paquete com.pagosaldia.datasource y sus clases……. .141
FIGURA 5.6 Paquete com.pagosaldia.displaytag.wrapper y sus clases……141
FIGURA 5.7 Prueba de rendimiento con 100 usuarios conectados….145

FIGURA 5.8 Prueba de rendimiento con 250 usuarios conectados... .146
FIGURA 5.9 Prueba de rendimiento con 500 usuarios conectados….147
FIGURA 5.10 Prueba de rendimiento con 1000 usuarios conectados. .148
FIGURA B.1 Encriptación y desencriptación de datos……..…….183
FIGURA B.2 Criptografía de clave secreta……..184
FIGURA B.3 Criptografía de clave pública…….. .185
FIGURA B.4 Autenticación……...……186
FIGURA B.5 Protocolo de seguridad SSL…….. .189
FIGURA B.6 Protocolo de seguridad SET……..192
FIGURA B.7 Configuración típica de un Firewall……...……197
FIGURA C.1 Diagrama de Secuencia de Inicio de Sesión ……. .199
FIGURA C.2 Diagrama de Secuencia de Registro de Usuario …….200
FIGURA C.3 Diagrama de Secuencia de Consultar Planilla de Agua..201
FIGURA C.4 Diagrama de Secuencia de Consultar Planilla de Luz... .202
FIGURA C.5 Diagrama de Secuencia de Consultar Planilla de Teléfono……...203
FIGURA C.6.a Diagrama de Secuencia de Seleccionar Opción Pagar Planilla de Agua ……..……204
FIGURA C.6.b Diagrama de Secuencia de Selección de planilla de Agua a pagar ……...……205
FIGURA C.7.a Diagrama de Secuencia de Seleccionar Opción Pagar Planilla de Luz ……206
FIGURA C.7.b Diagrama de Secuencia de Selección de planilla de Luz a pagar ……...207
FIGURA C.8.a Diagrama de Secuencia de Seleccionar Opción Pagar Planilla de Teléfono ……208
FIGURA C.8.b Diagrama de Secuencia de Selección de planilla de Teléfono a pagar ……209
FIGURA C.9 Diagrama de Secuencia de Pago de Planilla……....210
FIGURA C.10 Diagrama de Secuencia de Salir del Sistema……..211
FIGURA D.1 Modelo de configuración de VPN red a red……. .289
FIGURA D.2 Configuración de equipos servidores de VPN ……..……292
FIGURA D.3 Pantalla mostrada cuando se ejecuta por primera vez ACID……326
FIGURA D.4 Pantalla para optimización y configuración de la Base de datos para ACID…….. .326
FIGURA D.5 Pantalla de Análisis de alertas……327

ÍNDICE DE TABLAS

Pág.

TABLA 1.1 Formas de pago ofrecidas por las empresas que permiten realizar pagos de servicios básicos…….. 6
TABLA 2.1 Costo asociado al proceso de pago…….. ..17
TABLA 2.2 Clasificación de los competidores……....18
TABLA 3.1 Requerimientos de hardware del servidor web……...37
TABLA 3.2 Requerimientos de hardware del servidor de correo.…...38
TABLA 3.3 Requerimientos de hardware del servidor de nombres de dominio……..39
TABLA 3.4 Requerimientos de hardware del servidor de base de datos……...40
TABLA 3.5 Requerimientos de hardware del servidor Proxy……...41
TABLA 3.6 Requerimientos de hardware de firewall interno……....42
TABLA 3.7 Requerimientos de hardware de firewall externo……....43
TABLA 3.8 Requerimientos de hardware de usuarios de la Lan interna……..44
TABLA 3.9 Dispositivos de red……...45

ABREVIATURAS

3DES	Triple DES
AES	Advanced Encryption
API	Application Programming Interface
CGI	Common Gateway Interface
CVS	Control Version System
DES	Data Encryption Estándar
DMZ	De-Militarized Zone
DNS	Domain Name Server
EJB	Enterprise JavaBeans
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GUI	Graphic User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IDE	Integrated Development Environment
IDS	Intrusion Detection System
J2EE	Java 2 Enterprise Edition
JDT	JDT, plugin para el lenguaje Java
JMX	Java Management eXtensions
JRE	Java 2 Runtime Environment
JSP	Java Server Pages
LAN	Local Area Network
MD5	Message Digest 5

MTA	Mail Transport Agent
MUA	Mail User Agent
NAT	Network Address Translation
PDE	Plug-in Development Environment
PHP	Hypertext Preprocessor
RAM	Random Access Memory
RDBMS	Relational Data Base Manager System
SET	Secure Electronic Transaction
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSI	Server Side Incluye
SSL	Secure Sockets Layer
SWT	Standard Widget Toolkit
TCP/IP	Transmission Control Protocol / Internet Protocol
UML	Unified Modeling Language
URL	Uniform Resource Locators
UTP	Unshielded Twisted Pair
VPN	Virtual Private Network
XML	eXtensible Markup Language

CAPITULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN

El comercio electrónico en el Ecuador tiene poca acogida, existen varios factores por los que se da esta situación, en la mayoría de los casos es, porque los ciudadanos no tienen confianza al momento de ingresar los datos de su tarjeta de crédito o de débito, por otro lado un gran número de personas no cuenta con el acceso a Internet o tienen poco conocimiento sobre lo que se refiere a pagos a través del Web.

Otro factor a considerar es la ley de comercio electrónico que está en vigencia desde el 17 de Abril del 2002 y su reglamento desde el 31 de Diciembre del 2002, en donde encontramos los términos bajo los cuales se rige el comercio electrónico y las sanciones aplicadas en caso de encontrar irregularidades en alguna transacción en línea. También el código penal debe ser reformado debido a las nuevas formas de delito que surgen en el

comercio electrónico, actualmente se revisa el código penal con el fin de incluir estos nuevos delitos como son: Fraude informático; Delito de daños informáticos; Falsificación electrónica; Intromisión indebida a los sistemas de información o telemáticos; Recopilación de información por medios fraudulentos; Violación al derecho a la privacidad en los términos de la ley de comercio electrónico.

La idea de crear un portal de pagos de servicios básicos surge debido a que no existe un portal exclusivo para realizar el pago de estos servicios y provea varias opciones para hacerlo, como son: tarjeta de débito o tarjeta de crédito de cualquier institución financiera. El inconveniente del servicio ofrecido por los bancos es que el cliente debe tener una cuenta en el banco que permite realizar el pago, dejando de lado a aquellas personas que tienen una tarjeta de débito y su banco emisor no ofrece el servicio de pagos.

1.1. Pagos de Servicios Básicos en el Ecuador

Existen varias alternativas para realizar el pago de las planillas de agua, luz y teléfono en el Ecuador: en las agencias de las empresas que proporcionan los servicios básicos, en los bancos y en empresas auxiliares que también permiten realizar pagos. Como empresas auxiliares se ha llamado a aquellas empresas que permiten realizar pagos y que no tienen dependencia con la empresa de servicios básicos excepto los bancos.

A continuación se describe cómo se realizan los pagos con cada una de estas alternativas.

- **Pago en la agencia que ofrece el servicio básico:**

  Las empresas que brindan los servicios básicos cuentan con varias agencias ubicadas en distintos sectores de la ciudad a la que proporcionan el servicio que permiten a sus clientes realizar los pagos de sus planillas.

  Entre las ventajas de pagar en estas agencias están las diferentes alternativas de pago que ofrecen: efectivo, tarjeta de crédito, cheque; también permiten hacer convenios de pagos permitiendo a

los usuarios pagar parcialmente sus planillas dependiendo del monto facturado.

La principal desventaja es que los clientes deben contar con el tiempo suficiente para acercarse a estas agencias a cancelar sus planillas y en la mayoría de los casos deben hacer largas colas

- **Pago en bancos:**

  Los bancos también permiten realizar el pago de planillas de servicios básicos. Existen dos opciones para pagar: acercándose a las diferentes sucursales o por Internet.

  La mayoría de los bancos cuentan con la primera opción, la misma que tiene la ventaja de que cualquier persona puede acercarse a cualquier agencia bancaria a realizar sus pagos sin necesidad de tener una cuenta en dicha entidad. La desventaja es que también requiere que el cliente dedique tiempo para acercarse hasta el banco.

  La segunda opción que es pagar a través de Internet no la ofrecen todos los bancos. La principal ventaja es que no requiere que el usuario se acerque a ninguna agencia ni hacer largas colas,

simplemente debe contar con el acceso a Internet. Como desventajas se tiene que algunos bancos requieren que la persona que va a pagar tenga una cuenta en dicho banco o que cuenten con una tarjeta de crédito. Además, en algunos bancos cobran un valor elevado por este servicio, y otros solo permiten pagar uno de los servicios básicos.

- **Pago en empresas auxiliares**

  Existen empresas que permiten a los usuarios de servicios básicos pagar sus planillas.

  Como ventajas se puede mencionar las diferentes formas de pago que ofrecen como son: efectivo, tarjeta de crédito y cheque. Entre las desventajas se tiene que las personas deben acercarse a estas agencias a realizar sus pagos y el costo adicional por este servicio es un poco elevado.

En la siguiente tabla se muestran las formas de pago ofrecidas por las empresas que permiten realizar pagos de planillas de servicios básicos antes mencionadas.

	FORMAS DE PAGO[1]				
	Efectivo	Tarjeta Crédito	Cheque	Otro *	Internet
Empresa que ofrece el servicio básico	✓	✓	✓	✓	✗
Bancos	✓	✗	✓	✗	✓
Empresas Auxiliares	✓	✗	✓	✗	✗

* Pagos parciales de planillas

Tabla 1.1.- Formas de pago ofrecidas por las empresas que permiten realizar pagos de servicios básicos.

[1] **Fuente:** Empresas que permiten realizar pagos de planillas de servicios básicos.

1.2. Justificación y objetivos

En la actualidad no existe en Ecuador un sitio en Internet que permita realizar exclusivamente el pago de cualquier servicio básico, sin importar la empresa que lo proporciona y la ciudad en la que se encuentra.

Los bancos son las únicas entidades en el Ecuador que permiten realizar pagos de servicios básicos (agua, luz, teléfono) a través de Internet, muchos de ellos requieren que el cliente tenga una cuenta en el mismo banco y en algunos casos el costo por este servicio es muy elevado comparado con el total a pagar.

Otra forma de realizar los pagos es por medio de instituciones como Servipagos que cobran un valor adicional por este servicio. Estas instituciones no ofrecen ningún servicio básico pero permiten realizar los pagos por medio de ellas. Únicamente agencias de la misma empresa que brinda el servicio no cobran un valor adicional, el inconveniente es que muchas personas no cuentan con el tiempo suficiente para acercarse a estas agencias que generalmente requieren hacer largas filas para poder realizar dichos pagos.

En muchos casos las personas tienen que cancelar más de una planilla en más de una agencia lo que requeriría una mayor cantidad de tiempo.

Justificación:

Por lo expuesto anteriormente se ha pensado en desarrollar un portal que permita a cualquier persona que cuente con una tarjeta de crédito o débito realizar sus pagos de servicios en cualquier lugar del Ecuador y sin importar la empresa que lo ofrece, manteniendo protegida toda la información requerida en el proceso de pago. Además en este portal solo encontrará información relacionada a las planillas por ejemplo: ¿cómo realizar los pagos?, anuncios de las empresas que brindan estos servicios para mantener actualizados a los clientes y las preguntas frecuentes realizadas por los usuarios.

Objetivos:

Implementar un portal que sea exclusivamente para pago y consulta de planillas de servicios básicos en el Ecuador.

Hacer del portal la mejor opción para los clientes al momento de realizar el pago de sus planillas.

Implementar una infraestructura de red que permite proteger los datos del usuario del sistema y realizar los pagos de forma segura.

Ofrecer flexibilidad en la forma de realizar los pagos, esto es tarjeta de crédito y tarjeta de débito de cualquier banco.

Aplicar los conceptos de seguridad informática aprendidos durante el desarrollo del tópico de graduación “Seguridad de Información”, detallados en el Apéndice B.

1.3. Usuarios del sistema

El sistema está orientado a todas las persona que no disponen de tiempo para acercarse a una agencia a realizar el pago de sus planillas, teniendo la mayoría que pagar más de una planilla, que tienen acceso a Internet y tienen un nivel de conocimiento básico en el uso de aplicaciones Web.

El capítulo dos hace un estudio más detallado sobre quienes serán los usuarios del sistema basados en una encuesta realizada.

1.4. Limitaciones y restricciones

Limitaciones:

Las agencias de las empresas que ofrecen servicios básicos dan la posibilidad a sus clientes para hacer pagos parciales, esta opción no es permitida en la aplicación, la planilla debe ser cancelada en su totalidad.

Los usuarios del sistema deben contar con una tarjeta de crédito o débito para poder cancelar sus planillas debido a que estas son las formas de pago que el portal ofrece.

Los usuarios que se registren deben tener una cuenta con alguna de las empresas que ofrecen servicios básicos de lo contrario no podrán visualizar ninguna planilla. Si algún usuario desea pagar una planilla de otra persona debe registrarse ingresando el número de cédula del titular de la cuenta o con el usuario y contraseña del titular previamente registrado.

El sistema permita realizar únicamente pagos de agua, luz y teléfono.

Restricciones:

Si el proceso de pago es realizado por una tercera entidad se estará sujeto a su esquema de seguridad, poniendo en peligro la confidencialidad de los datos de la tarjeta del cliente.

El número de usuarios registrados se ve afectado mientras menor sea el número de personas que hayan recibido capacitación sobre Internet. Los valores que los proveedores de Internet cobran a sus usuarios también es un factor que afecta al número de usuarios que accedan al portal.

1.5. Alcance

El portal básicamente está diseñado para que el usuario pueda hacer consultas y pagos de sus planillas pendientes. A continuación se lo describe:

1. Permite registrarse a los usuarios, de esta forma podrán hacer sus consultas y pagos de planillas, y la información que se les mostrará será personalizada de acuerdo a los datos ingresados en el formulario de registro.

2. Los usuarios pueden realizar consultas de sus planillas pendientes de agua, luz y teléfono. En caso de no tener planillas pendientes no se le mostrará ningún registro.

3. El sistema permite además visualizar las planillas. La planilla que verá es igual a la que entregan en cada domicilio, para poder hacerlo debe cancelar un valor adicional al total de la planilla.

4. También se puede realizar el pago de planillas, esta transacción tiene un costo adicional que será sumado al total de la planilla que se desea cancelar. Una vez realizado el pago se envía un mail al usuario con detalles del mismo.

5. Las consultas o pagos que se realicen son registrados en la base de datos. En la base de datos de la empresa de servicio básico se actualiza el estado de la planilla indicando que ésta ha sido cancelada.

6. Los datos del servidor Web viajan hacia la base de datos a través de una red virtual privada la misma que se encarga de encriptar los datos en el emisor y desencriptarlos en el receptor, de esta forma se protege de usuarios malintencionados.

7. El acceso de usuarios desde Internet a la DMZ es controlado por un firewall externo, el mismo que permite o niega el acceso según el servicio requerido. También ha sido instalado un firewall interno que se encarga de controlar el acceso a la base de datos por parte de los usuarios de la red interna ya que no todos están autorizados para hacerlo.

8. El servidor de correo es solo para los usuarios de la red local, los mismos que deben autenticarse por medio de un usuario y contraseña previamente creado por el administrador para poder tener acceso a él.

9. Los usuarios de la red local pueden acceder a Internet a través de un servidor Proxy, para poder hacerlo deben autenticarse a través de un usuario y una contraseña creado por el administrador.

CAPITULO 2

2. ANÁLISIS DE MERCADO

En este capítulo explicamos cuáles son los servicios que ofrece el portal; también haremos un análisis del segmento de mercado, los competidores y la estrategia de marketing empleada para promocionarlo, ya que de estos factores depende el éxito o fracaso del mismo.

También realizamos una encuesta sobre servicios básicos para conocer un poco más sobre las preferencias de los usuarios al momento de realizar sus pagos. Los resultados de la encuesta se muestran en el apéndice A.

2.1. Servicios ofrecidos por el portal de pagos

El portal ofrece dos tipos de servicios: consultas y pagos de planillas de agua, luz y teléfono. Para poder acceder a estos servicios, los usuarios deben llenar un formulario de registro en donde ingresarán sus datos personales.

Existen dos tipos de consultas:

- Consulta de planillas pendientes
- Consulta de la factura detallada

La consulta de planillas pendientes muestra un listado con todas las planillas pendientes del usuario agrupadas de acuerdo al servicio básico. Este listado presenta la siguiente información con respecto a la planilla: número de cuenta, número de factura, nombre del propietario de la cuenta, fecha máxima de pago, total adeudado y empresa que proporciona el servicio

La consulta de factura detallada permite mostrar una imagen de la planilla conteniendo toda la información, como las planillas que llegan al domicilio del propietario. Este tipo de consulta tiene un costo de diez centavos por cada factura que se desee visualizar.

El segundo servicio ofrecido por el portal es el de realizar el pago de las planillas, este servicio tiene un costo que varía dependiendo del monto a pagar.

En la siguiente tabla se detalla el costo de la transacción cuyo valor varía de acuerdo al monto total de la factura que el cliente desee cancelar:

TOTAL DE LA PLANILLA (dólares)	COSTO DE LA TRANSACCION (dólares)
menor o igual a 15	0,20
mayor a 15 y menor o igual a 20	0,30
mayor a 20 y menor o igual a 30	0,35
mayor a 30 y menor o igual a 60	0,50
mayor a 60 y menor o igual a 80	1,00
mayor a 80	1,50

Tabla 2.1.- Costo asociado al proceso de pago.

También se pueden ver anuncios actualizados con noticias de las empresas de agua, luz y teléfono, que son de interés para los clientes; pero esta opción no ha sido considerada como un servicio del portal, sino como un valor agregado para los clientes.

2.2. Análisis de la competencia

Los competidores están clasificados de la siguiente manera:

COMPETIDORES	
On-line	**Off-line**
Bancos	Bancos
	Agencia de la empresa que ofrece el servicio básico
	Empresas auxiliares (Servipagos)

Tabla 2.2.- Clasificación de los competidores.

Según la tabla anterior observamos que solo los bancos ofrecen el servicio de pagos a través de Internet; pero por esta razón no se debe dejar de considerar a las empresas que permiten el pago de planillas off-line, debido a que, según la encuesta que realizamos, la mayor parte de los clientes realizan sus pagos acercándose a las agencias de la empresa de servicios básicos y otro gran número de personas lo realizan acercándose a los bancos. El número de personas que realizan sus pagos por Internet es muy pequeño.

Se puede concluir que los principales competidores son las empresas de servicios básicos y los bancos, aunque no se debe dejar de lado a las empresas auxiliares que aunque en un menor porcentaje también es una opción para que los usuarios realicen sus pagos.

Todas estas empresas tienen la ventaja de ya estar presentes en el mercado y contar con un número de clientes, por lo tanto la promoción del portal será muy importante para llegar a tener éxito.

Como ventaja se tiene la agilidad para realizar los pagos sin tener que acercarse a ninguna agencia, la información que encontrará en el portal es exclusivamente sobre servicios básicos. Además se le hará conocer al cliente a través del correo electrónico sus facturas pendientes y la fecha máxima permitida para realizar los pagos.

2.3. Segmento de mercado

Según una encuesta realizada el 94 por ciento de los usuarios realizan pagos de servicios básicos; de ellos el 14 por ciento son mayores a 40 años, el 27 por ciento tienen entre 31 y 40 años, el 54 por ciento tienen entre 20 y 30 años, y solo un 5 por ciento tienen menos de veinte años. Con estos datos se puede concluir que la edad de la mayoría de los posibles clientes estará entre 20 y 40 años. Del 5 por ciento de personas que pagan por Internet todas están en este rango de edad.

Para determinar el segmento de mercado también se deben considerar otros factores como el acceso a Internet, el nivel de conocimiento en el uso de aplicaciones Web, si poseen tarjeta de débito o crédito, si tienen una cuenta en alguna de las empresas que ofrecen servicios básicos etc. Este último factor mencionado es importante porque para registrarse en el portal se debe ingresar el número de cédula de una persona que tenga una cuenta en alguna empresa de servicio básico, de lo contrario no tendrá acceso a información alguna.

Con lo expuesto anteriormente, se puede llegar a la conclusión de que el segmento de mercado está conformado por todas aquellas personas que tengan más de veinte años, tengan una cuenta en alguna empresa de

servicios básicos, posean una tarjeta de crédito o débito, tengan acceso a Internet y un nivel de conocimiento mínimo en el manejo de aplicaciones Web.

Todos los resultados de la encuesta realizada pueden verse en el Apéndice A.

2.4. Estrategia de marketing

Una vez definido el segmento de mercado, vamos a definir los mecanismos empleados para poder llegar al mayor número de posibles clientes.

Debemos reconocer, que un gran número de personas no confían en las transacciones hechas a través de Internet, por esta razón no solo se tiene que promocionar los servicios ofrecidos por el portal, también se debe conseguir que confíen en los mecanismos de seguridad aplicados al portal.

Actualmente la mejor forma de hacer publicidad es a través de los buscadores en Internet debido a que serán muchas las personas que visitarán el sitio y es gratuito. Debido a que el número de personas que tienen acceso a Internet es pequeño, se tiene que hacer más énfasis en medios como prensa escrita, televisión, radio, lugares más frecuentados, etc. También se puede promocionar el sitio poniendo enlaces al portal en las páginas Web más visitadas del Ecuador y en portales de las empresas que ofrecen servicios básicos.

Para promocionar el portal, se han elegido algunas estrategias que serán explicadas a continuación:

- **Seleccionar un nombre adecuado para el portal**

  Escoger un nombre no muy extenso y que haga referencia a la actividad a que se dedica la empresa, esto ayudará a las personas a recordar fácilmente la dirección y no tener que utilizar algún buscador para encontrar el portal.

- **Anuncios en sitios Web de empresas de servicios básicos**

  Poner un enlace al portal en la página de las empresas que ofrecen servicios básicos ayudará a que los clientes de dichas empresas se interesen en conocer el portal.

- **Sitios Web ecuatorianos más visitados**

  Obtener una lista de los portales ecuatorianos más visitados y también publicar un anuncio del portal y el respectivo enlace a la página principal.

- **Lugares altamente frecuentados**

  No solamente la publicidad por Internet será necesaria, también colocar anuncios en lugares que sean frecuentados por un gran número de personas como los centros comerciales, agencias donde realizan pagos habitualmente, etc.

- **Hojas volantes**

  Entrega de hojas volantes con información acerca del portal y los servicios que ofrece, también es una buena opción para promocionarlo.

- **Correo electrónico**

  Obteniendo un listado con direcciones de correo electrónico de ecuatorianos y enviarles mails con información de los servicios ofrecidos.

- **Optimización para buscadores**

  La optimización para que el portal sea uno de las primeras direcciones mostradas en los buscadores sería de gran ayuda, debido a que las personas, por lo general, solo revisan los resultados de la primera página.

- **Boletines informativos**

  Enviar boletines informativos promocionando los servicios y beneficios que puede proporcionar el portal, además enviar periódicamente información con las últimas novedades en las empresas de servicios básicos para mantener al día a los clientes.

Al inicio, la promoción del portal debe ser mayor, pues debemos convencer a todas aquellas personas que prefieren realizar sus pagos en las agencias, de que los servicios del portal son buenos y les ofrece mayores beneficios.

Siempre debemos informar a los clientes de cualquier modificación en los servicios y hacerles conocer que siempre es pensado en el beneficio de los ellos.

CAPITULO 3

3. ANÁLISIS DEL SISTEMA

En el análisis del sistema definimos los requerimientos que son contemplados en el Portal de pagos, también elaboramos un modelo conceptual del sistema en el cual se van a representar los elementos o conceptos del dominio del problema y sus relaciones con otros conceptos.

En este capítulo también realizamos un estudio técnico del hardware y software utilizado para el desarrollo e implementación del sistema, además incluiremos un análisis de seguridad en donde explicamos las herramientas utilizadas para controlar la seguridad de portal.

3.1. Requerimientos

Es importante que los requerimientos sean definidos y especificados cuidadosamente, pues ellos definen el alcance del sistema, si se omite algún requerimiento, este no será incluido en el sistema final.

A continuación se listan todos los requerimientos seguida de explicación detallada de cada uno:

- **Registrar nuevo usuario del sistema**
  1. Para poder hacer uso de los servicios que ofrece el sistema, el usuario debe registrarse.

  2. El usuario debe llenar un formulario y estos datos serán almacenados en la base de datos.

- **Validar datos ingresados por el usuario**
  1. Los datos ingresados por el usuario en el formulario de registro tales como: nombre, apellido, dirección, teléfono, cédula, correo serán validados.

2. La contraseña y el usuario deben tener más de seis caracteres.

3. Antes de que los datos del cliente nuevo sean almacenados, el nombre del usuario es verificado si no existe para evitar duplicidad de usuarios.

- **Acceder al sistema (ingresar usuario y contraseña)**

  Para poder acceder a los servicios de Pagos y Consultas de Servicios Básicos, previamente debe ingresar un usuario y su respectiva contraseña válida.

- **Validar usuario**

  Cuando se ingresa el usuario y su contraseña para acceder a los servicios del sistema, el usuario tiene que ser reconocido como un usuario valido o registrado en la Base de Datos del sistema.

- **Encriptar contraseña de usuario**

  Después de validar el usuario y cuando se registra un nuevo usuario, la contraseña será transmitida y almacenada en forma encriptada o cifrada para aumentar el nivel de seguridad con el sistema y confianza con el cliente.

- **Consulta general de datos de factura.**

  Al ser reconocido como un usuario registrado, el sistema buscará datos generales de las planillas de servicios básicos ofrecidos por empresas vinculadas al portal.

  Los datos generales están agrupados por las empresas de los servicios Básicos, así mismo, por cada cuenta que tenga el usuario en esas empresas con su fecha Máxima de Pago.

- **Consulta detallada de datos factura por servicio básico**

  Con este tipo de Consulta, el usuario, además de tener la visión global de su cuenta (Consulta General), podrá visualizar el número de factura, el saldo de la factura y también tendrá un enlace que le proveerá la visualización de la respectiva planilla.

- **Visualizar planilla**

  Todo usuario, que haga una Consulta Detallada, tendrá la facilidad de Generar su planilla, donde visualizará toda la información que observa en una planilla emitida por la empresa que ofrece el Servicio.

- **Almacenamiento de consulta**

  El sistema, en forma automática y transparente al usuario, registrará o almacenará cada consulta efectuada a través del mismo.

- **Pago de visualización de planilla**

  El usuario, al Generar la Visualización de su planilla de Servicio Básico, debe efectuar un pago en línea por su planilla.

- **Imprimir planilla**

  Al haberse Generado la visualización de su planilla, el usuario tiene la opción de poder imprimir la planilla.

- **Pago de servicio básico**

  Así mismo, el usuario tiene la facilidad de poder efectuar el pago en línea de su planilla, para esto, debe ingresar su número de tarjeta de crédito o débito con su respectiva contraseña,

- **Validación de número de tarjeta**

  Al ingresar el número de tarjeta para efectuar un Pago, este número será validado para verificar su autenticidad y estado.

- **Validación de clave de tarjeta**

  Al ingresar la clave respectiva de la tarjeta, ésta será verificada si corresponde con el número de tarjeta.

- **Validación de cupo de tarjeta**

  Para poder efectuar el pago en línea con una tarjeta, se verificará que la tarjeta tenga un cupo mínimo, equivalente al pago a efectuarse.

- **Encriptación de datos de tarjeta**

  Al efectuar toda transacción donde se efectúe un pago con tarjeta, el número de la tarjeta y su clave van a estar encriptadas o cifradas para tener un mayor nivel de seguridad en el sistema.

- **Almacenamiento de pago**

  Toda transacción, que involucre un pago en línea desde el Sistema, sé almacenará como un registro en la Base de Datos, para reflejar el pago correspondiente al usuario.

- **Notificar al cliente sobre pago realizado**

  Al terminar la secuencia para efectuar un pago en línea, desde el Sistema, se emitirá un correo de notificación, donde se informará al

usuario sobre los detalles del pago que se ha realizado con el respectivo monto.

- **Salir del sistema**

  El usuario tendrá la facilidad de poder cerrar su sesión de usuario, para poder evitar cualquier transacción posterior a su cierre.

- **Contar con una sección de preguntas frecuentes**

  El portal contará con una sección de preguntas frecuentes, donde constarán las principales preguntas que realicen los usuarios del sistema y las respectivas respuestas para facilitar el acceso a cualquiera de los servicios ofrecidos.

- **Contar con una sección de contactos**

  El sistema incluirá una sección que contenga todas las formas de contacto con las personas encargadas de la administración del portal. Las formas de contacto son: teléfono, correo electrónico.

- **Fácil de Usar**

  Las opciones para acceder a los servicios que ofrece el portal serán coherentes con la acción a realizar y estarán claramente representadas para evitar confusiones por parte del usuario.

- **Confiabilidad**

  Los datos confidenciales del usuario como: número de tarjeta, clave y contraseña de ingreso al sistema serán encriptados, de esta forma protegemos del uso indebido de la información.

- **Mecanismo de respaldo de base de datos**

  Los datos de la base serán respaldados 2 veces al día, de la siguiente manera:

  - 12 H 00
  - 24 H 00

  Los datos del servidor de aplicaciones serán respaldados una vez al día a las 24 H 00

3.2. Modelo Conceptual

El modelo conceptual es uno de las primeras actividades que deben realizarse cuando se desarrolla un sistema, en el mismo están representados todos los conceptos del dominio del problema, y como están asociados. Este modelo subraya fuertemente una concentración en los conceptos del problema, no en las entidades del software.

Los conceptos son descritos a continuación:

Consulta Básica: Datos principales de las planillas.

Costo por Transacción: Se refiere al valor que el portal cobra por el servicio de pago de una planilla.

Formulario Registro: Datos ingresados por un usuario al momento de registrarse.

Imagen Planilla: Se refiere a la información que contiene una planilla de cualquier servicio básico.

Pago Servicio Básico: Se refiere a la información asociada al pago de una planilla de cualquier servicio básico.

Sesión: Tiempo que un usuario permanece conectado al sistema, luego de haber validado su ingreso.

Tarjeta: Objeto utilizado en el proceso de pago y que equivale al dinero físico en una compra.

Usuario: Persona que ingresa al sistema para hacer alguna consulta o pago.

Los conceptos y sus asociaciones son mostrados en el siguiente diagrama:

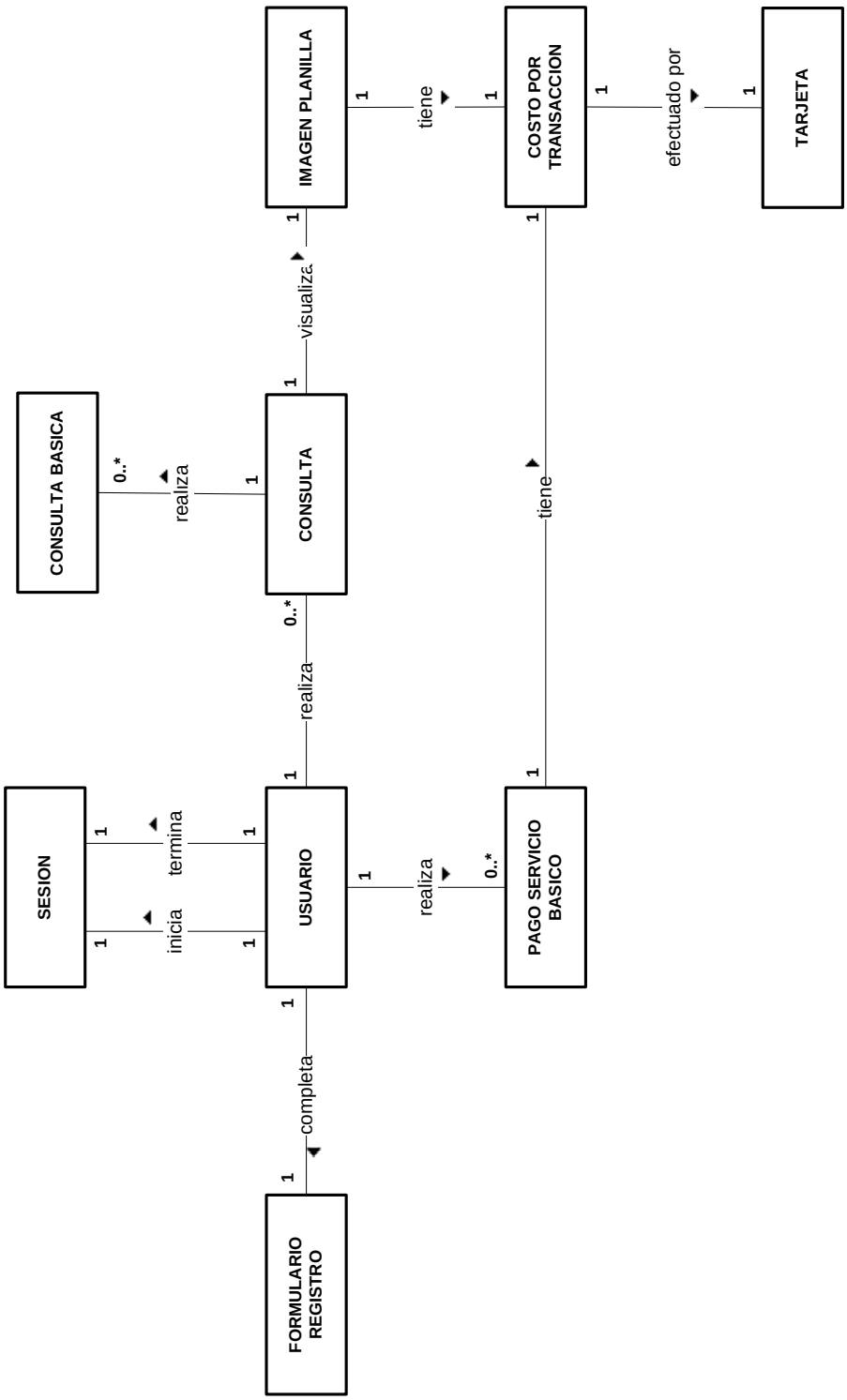


Figura 3.1. Modelo Conceptual

3.3. Análisis Técnico

En esta sección, realizamos un estudio de las herramientas utilizadas para la implementación del portal de comercio electrónico y la infraestructura de red, las mismas que han sido divididas en: Herramientas de Hardware y Herramientas de Software.

A continuación damos una explicación de las características de cada una de estas herramientas.

3.3.1. Herramientas de Hardware

Equipo para Servidor Web

Para que el desempeño de un servidor Web sea satisfactorio, debe ser instalado en un equipo que cumpla con los siguientes requerimientos mínimos:

REQUERIMIENTOS DE HARDWARE DEL SERVIDOR WEB	
CARACTERISTICA	VALOR
Velocidad del procesador	500 Mhz
Memoria RAM	128 MB
Capacidad del disco duro	3 GB
Tarjeta de red	1

Tabla 3.1. Requerimientos de hardware del servidor Web

El servidor Web del proyecto tiene un procesador Pentium IV de 1.8 GHz, 256 MB de memoria RAM y un disco duro con capacidad de almacenamiento de 30 GB. La capacidad del disco duro debe ser elegida de acuerdo al tamaño, imágenes, archivos multimedia que la aplicación Web requiera; en este caso la aplicación no utiliza muchos de estos recursos, por lo tanto, el requerimiento de espacio en disco es mínimo. También es necesaria una conexión a Internet, en este caso, se tiene acceso a través de un firewall al cual se conecta usando una tarjeta de red.

Equipo para Servidor de Correo

Los requerimientos mínimos de hardware para la puesta en marcha de un servidor de correo son los siguientes:

REQUERIMIENTOS DE HARDWARE DEL SERVIDOR DE CORREO	
CARACTERISTICA	VALOR
Velocidad del procesador	500 Mhz
Memoria RAM	128 MB
Capacidad del disco duro	8 GB
Tarjeta de red	1

Tabla 3.2. Requerimientos de hardware del servidor de correo.

Las características del servidor de correo son las siguientes: procesador Pentium II de 500MHz, 128 MB de memoria RAM, 40 GB de disco duro. El espacio en el disco duro depende de la cantidad de usuarios que tengan una cuenta en el servidor y la cuota que se le asigne a cada uno, en este caso, solo los usuarios internos de la LAN tienen cuenta, por lo tanto, el número es muy reducido. La conexión a Internet la hace a través de un firewall, para esto es necesaria otra tarjeta de red.

Equipo para Servidor de Nombres de Dominio (DNS)

Los requerimientos de hardware para un servidor de nombres de dominio son mínimos, en la siguiente lista veremos cuales son.

REQUERIMIENTOS DE HARDWARE DEL SERVIDOR DE NOMBRES DE DOMINIO	
CARACTERISTICA	VALOR
Velocidad del procesador	100 Mhz
Memoria RAM	16 MB
Capacidad del disco duro	1 GB
Tarjeta de red	1

Tabla 3.3. Requerimientos de hardware del servidor de nombres de dominio

El servidor DNS del portal tiene las siguientes características: procesador Pentium II de 400 MHz, 192 MB de memoria RAM, 30 GB de espacio en el disco duro, una tarjeta de red para conectarse con el firewall.

Equipo para Servidor de Base de Datos

Los requerimientos mínimos para la instalación de un servidor de base de datos Oracle 9i son los siguientes:

REQUERIMIENTOS DE HARDWARE DEL SERVIDOR DE BASE DE DATOS	
CARACTERISTICA	VALOR
Velocidad del procesador	400 Mhz
Memoria RAM	512 MB
Capacidad del disco duro	3 GB
Espacio swap (para sistemas linux)	1 GB
Tarjeta de red	1

Tabla 3.4. Requerimientos de hardware del servidor de base de datos

La características del servidor de base de datos que se ha implementado son: procesador Pentium IV de 2.4 GHz, 1 GB de memoria RAM, 60 GB despacio en el disco duro, en la instalación del sistema operativo linux se debe especificar el espacio swap, en este caso es de 2 GB. También

necesitamos una tarjeta de red para conectar el servidor al firewall que es el que permite o niega el acceso a los recursos.

Equipo para Servidor Proxy

Los requerimientos mínimos para la instalación de un servidor proxy son los siguientes:

REQUERIMIENTOS DE HARDWARE DEL SERVIDOR PROXY	
CARACTERISTICA	VALOR
Velocidad del procesador	1000 Mhz
Memoria RAM	128 MB
Capacidad del disco duro	3 GB
Tarjeta de red	1

Tabla 3.5. Requerimientos de hardware del servidor proxy

La características del servidor Proxy del portal son las siguientes: procesador Pentium IV de 2.5 GHz, 512 MB de memoria RAM, 30 GB de espacio en el disco duro, una tarjeta de red.

Equipo para Firewall Interno

El firewall interno tiene la función de controlar el acceso por parte de los usuarios de la Lan interna al servidor de base de datos.

Los requerimientos mínimos de hardware para implementar un firewall usando Iptables de Linux son:

REQUERIMIENTOS DE HARDWARE DE FIREWALL INTERNO	
CARACTERISTICA	VALOR
Velocidad del procesador	100 Mhz
Memoria RAM	64 MB
Capacidad del disco duro	1 GB
Tarjeta de red	2

Tabla 3.6. Requerimientos de hardware de firewall interno

El firewall interno del portal tiene las siguientes características: procesador Pentium IV de 400 MHz, 128 MB de memoria RAM, 30 GB de espacio en el disco duro y dos tarjetas de red, una para conectarse al servidor de base de datos y otra para conectarse a la Lan interna.

Equipo para Firewall Externo

El firewall externo cumple las siguientes funciones:

- Controlar el acceso por parte de los usuarios de Internet y de la Lan interna a los servidores que se encuentran en la DMZ.
- Controlar el acceso de los usuarios de la Lan interna a Internet.
- Controlar el acceso de usuarios de la DMZ a la Lan interna.
- Controlar el acceso de usuarios de Internet a la Lan interna.

Para implementar un firewall usando Iptables de linux y que funcione adecuadamente, se deben cumplir los siguientes requerimientos mínimos de hardware:

REQUERIMIENTOS DE HARDWARE DE FIREWALL EXTERNO	
CARACTERISTICA	VALOR
Velocidad del procesador	100 Mhz
Memoria RAM	64 MB
Capacidad del disco duro	1 GB
Tarjeta de red	2

Tabla 3.7. Requerimientos de hardware de firewall externo

El firewall externo del portal tiene las siguientes características: procesador Pentium IV de 400 MHz, 128 MB de memoria RAM, 30 GB de espacio en el disco duro y tres tarjetas de red, una para conectarse a Internet, otra para conectarse a la DMZ y otra para conectarse a la Lan interna.

Equipos para usuarios de red Lan interna

Las estaciones de trabajo para los usuarios de la Lan interna cumplen las siguientes características:

REQUERIMIENTOS DE HARDWARE DE USUARIOS DE LA LAN INTERNA	
CARACTERISTICA	VALOR
Velocidad del procesador	2 Ghz
Memoria RAM	512 MB
Capacidad del disco duro	80 GB
Tarjeta de red	1

Tabla 3.8. Requerimientos de hardware de usuarios de la Lan interna

Dispositivos de Red

En la siguiente tabla se encuentran los dispositivos utilizados en la implementación de la red.

	switch de 12 puertos para red DMZ
	switch de 12 puertos para la red de servidores
	switch de 24 puertos para la Lan interna
	Tarjetas de red para las estaciones de trabajo
	13 Cables UTP CAT 6
	Servidores (DNS , Web, Correo, Base de Datos, Firewall Externo, Firewall Interno)
	Estaciones de trabajo (Dos desarrolladores, un webmaster)

Tabla 3.9. Dispositivos de red

3.3.2. Herramientas de Software

Entorno Integrado de Desarrollo (IDE)

El Entorno de desarrollo, utilizado para el Portal, es Eclipse, Eclipse es una poderosa herramienta, que permite integrar diferentes aplicaciones, para construir un entorno integrado de desarrollo (IDE).

Es un proyecto de desarrollo de software Open-Source, que está dividido en tres partes: Eclipse Project, Eclipse Tools y Eclipse Technology Project.

El Eclipse Project está subdividido a su vez en tres sub-proyectos que son: la propia plataforma, JDT (Java Development Tool) y PDE (Plugin Development Enviroment).

Mediante Eclipse se puede crear diversas aplicaciones como: sitios Web, programas Java, C++ y Enterprise Java Beans.

Su principal aplicación es JDT, herramienta para crear aplicaciones en Java.

Otras aplicaciones pueden ser integradas a Eclipse en forma de plugins, que son reconocidos automáticamente por Eclipse al iniciarlo.

Como Eclipse está escrito en Java, para su funcionamiento se debe tener instalado el JRE (Java Runtime Enviroment). Eclipse detecta automáticamente la ubicación del JRE instalado.

Las funcionalidades que otorga Eclipse se localizan de dos formas diferentes: en un pequeño núcleo conocido como el Plataform Runtime o en forma de plugins. Existe un conjunto de plugins que ya vienen con la plataforma. Entre los plugins que vienen con la plataforma encontramos:

Ant, Compare , Core , CVS , Debug , Help , JDT, Jface, Releng , Scripting, Search , SWT ,Text, UI, Update , Team , WebDAV .

Estos plugins pueden ser descargados de la siguiente dirección: http://www.eclipse-plugins.info/eclipse/plugins.jsp.

La plataforma Eclipse esta construida en base a plugins. Este mecanismo permite desarrollar, integrar y correr nuevos plugins.

Un plugin es la mínima unidad de la plataforma que puede ser desarrollada separadamente. Se pueden encontrar herramientas pequeñas desarrolladas en un sólo plug-in o herramientas mucho más complejas que se componen de un conjunto de plugins que se comunican entre sí.

Entre los principales beneficios de Eclipse se tienen:

- Es una herramienta Open-Source.
- Soporta la construcción de una variedad de herramientas para el desarrollo de aplicaciones.
- Soporta el desarrollo de aplicaciones basadas en GUI y no-GUI.
- Soporta herramientas que manipulan diferentes tipos de archivos, como: Java, C, C++, EJB, HTML, GIF, etc.
- Corre en una gran cantidad de sistemas operativos incluyendo Windows y Linux.

- Provee a los desarrolladores, herramientas (ej.- PDE) que facilitan la creación de plugins.
- Mediante JDT facilita la creación de aplicaciones programadas en Java.

Entre las desventajas del uso de Eclipse están:

- Si bien Eclipse es multiplataforma, los plugins no tienen por qué serlo.
- Existen plugins que sólo corren en una plataforma, o que aún no han sido desarrollado para más de una.
- Al ser una herramienta Open-Source, se desarrollan plugins que no tienen todas las funcionalidades que tienen en otras herramientas comerciales, como es IBM Websphere.

Lenguaje de programación

El lenguaje principal usado en el desarrollo de las páginas Web es JAVA, quien brinda una gran cantidad de características versus otros lenguajes de programación, entre ellas están:

- **Simple:** Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.
- **Orientado a Objetos:** Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.
- **Distribuido:** Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como: http y ftp, esto permite a los

programadores acceder a la información a través de la red, con tanta facilidad como a los ficheros locales.

La verdad es que Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas.

- **Robusto:** Java realiza verificaciones en busca de problemas, tanto en tiempo de compilación, como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error.

  Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria, resultado de punteros que señalan a zonas

equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java.

- **Portable:** Más allá de la portabilidad básica, por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas, de forma que las ventanas puedan ser implantadas en entornos Unix, PC o Mac.
- **Multihilos:** Al ser multihilos (multihilvanado, en mala traducción), Java permite muchas actividades simultáneas en un programa. Los threads o hilos (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los threads construidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++.
- **Dinámico:** Java se beneficia de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el

tiempo de ejecución. Las librerías nuevas o actualizadas no paralizan las aplicaciones actuales (siempre que mantengan el API anterior).

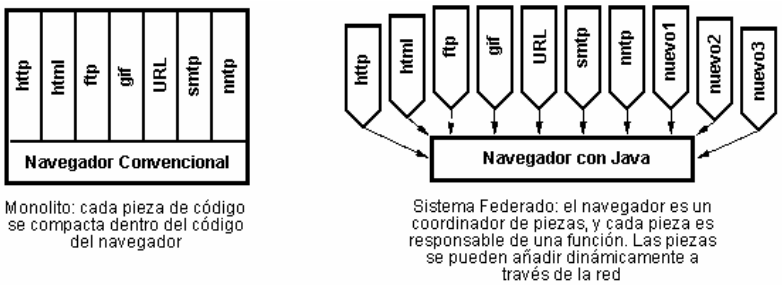


Figura 3.2. Comparación de un navegador que usa Java con un navegador convencional

Tecnología para desarrollo de páginas dinámicas

Para el desarrollo de las páginas Web dinámicas se eligió, como mejor opción el uso de JSP y Servlets, ya que su lenguaje nativo es Java con muchas librerías y código libre para desarrollo. Se dará una breve explicación, de las características más importantes, por las cuales se ha elegido esta tecnología para el desarrollo de las páginas.

¿Qué es un Java Server Page (JSP)?. Es una interfaz de programación de aplicaciones de servidores Web. En una

página jsp se entremezclan bloques de HTML estáticos y HTML dinámico, generados con Java, que se ejecutan en el servidor. Una página jsp puede procesar formularios Web, acceder a bases de datos y redireccionar a otras páginas.

Las páginas jsp son transformadas a un servlet y después compiladas.

El contenedor JSP, proporciona un motor que interpreta y procesa una página JSP como un servlet (en tomcat, dicho motor se llama jasper). Al estar basadas en los servlets, las distintas peticiones a una misma página jsp son atendidas por una única instancia del servlet.

Entre las principales diferencias entre un JSP y un Servlet están:

- En JSP, el código de presentación está separado de la lógica del programa, mientras que en un servlet, el código de presentación se compila dentro de la clase.
- En una página JSP, el código de presentación puede ser actualizado por un diseñador Web que no conozca Java.

- Los servlets se encuentran ya compilados, mientras que las páginas JSP se compilan bajo petición, lo que hace que la ejecución del servlet sea algo más rápida (en la primera petición).

Las principales características de un JSP son:

- Permiten separar la parte dinámica de la estática en una página Web
- Las páginas jsp, se almacenan en el servidor en archivos con extensión .jsp.
- El código JSP es java y se encierra entre: <% y %>, por ejemplo:

  <H1>Hora: ***<%****= new java.util.Date()* ***%>****</H1>*
- La sintaxis, también se puede expresar en formato XML:

  <jsp:xxx> ... </jsp:xxx>
- En una página jsp, hay varios objetos implícitos (predefinidos):

  request, response, out, session, application, config, pageContext, page y exception

- Cada página JSP, es compilada automáticamente hacia un servlet por el motor JSP la primera vez que se accede a esa página.
- Desde una página JSP se puede llamar a un componente Java Bean, donde se puede implementar la lógica de negocio.

Contenedor de Páginas JSP (Tomcat)

Como se mencionó anteriormente, Tomcat va a ayudar a interpretar las páginas dinámicas JSP y los Servlets.

Tomcat es desarrollado en un ambiente abierto y participativo, bajo Apache Software License. Se ha hecho, que el contenedor Tomcat interactúe con el Servidor Apache, para que el Servidor Web haga las peticiones a Tomcat y todos los Clientes se comuniquen solo con Apache.

El módulo que da la facilidad para la comunicación entre Apache y Tomcat se lo conoce como mod_jk1 y mod_jk2, el usado en el desarrollo del Portal es el mod_jk2. Una de las

características principales y mejoradas de mod_jk2 con respecto a mod_jk1 es que provee de multihilos.

Las principales características de Tomcat se mencionan a continuación:

- Características de Administración basadas en JMX.
- Administración de aplicaciones Web basadas en JSP y Struts.
- Compilador Jasper para páginas JSP.
- Buen manejo de memoria y rendimiento.
- Amplio soporte de integración con herramientas desarrollas de Apache Software.
- Tareas personalizadas para interactuar con las aplicaciones administrativas, directamente desde el script build.xml.

Servidor Web

Una de las herramientas más importantes para el desarrollo de un Portal de comercio electrónico, es tener un Servidor

Web, el mismo que facilita el mantenimiento y personalización de las páginas.

Como Servidor de páginas se ha elegido Apache, el cual tiene como una de las principales características su funcionamiento en plataformas virtuales muy utilizadas. Al principio, Apache se lo utilizaba para ser el primer Servidor Web basado en Unix, pero esto ya no es una realidad, ya que funciona en la mayoría de Sistemas Operativos, ya sea de escritorio o Servidores como Amiga OS 3.x y OS/2.

Apache presenta muchas otras características, entre ellas: un elaborado índice de directorios; un directorio de alias, negociación de contenidos, informe de errores HTTP configurable, ejecución SetUID de programas CGI, gestión de recursos para procesos hijos, integración de imágenes del lado del servidor, reescritura de las URL, comprobación de la ortografía de las URL y manuales online. El resto de características importantes de Apache son:

- **Soporte del ultimo protocolo HTTP 1.1:** Apache es uno de los primeros Servidores Web en integrar el protocolo HTTP 1.1 y al mismo tiempo sigue siendo

compatible con HTTP 1.0. Apache esta preparado para todas las novedades del nuevo protocolo. Por ejemplo, antes de HTTP 1.1, un navegador tenía que esperar respuesta del Servidor Web antes de poder emitir otra petición. Con el surgimiento de HTTP 1.1, esto ha dejado de ser así. Un navegador Web puede enviar solicitudes en paralelo, las cuales ahorran ancho de bando dejando de transmitir las cabeceras HTTP en cada solicitud.

- **Sencillo, con la configuración basada en un poderoso archivo:** el servidor Apache no posee una interfaz de usuario grafica para su administración. Se trata de un sencillo archivo de configuración, llamado httpd.conf, que se puede utilizar para configurar Apache. Únicamente necesita su editor de texto favorito. Sin embargo, es lo suficientemente flexible para permitirle repartir la configuración de su host virtual, en múltiples archivos, para no sobrecargar un único archivo httpd.conf, con toda la gestión de las múltiples configuraciones de servidores virtuales.
- **Soporte para CGI (Common Gateway Interface):** Apache soporta CGI, utilizando los módulos (*mod_cgi*

y mod_cgid). Es compatible con CGI y aporta características extendidas, como: personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores Web.

- **Soporte de FastCGI:** no todo el mundo escribe sus CGI en Perl, ¿como pueden hacer sus aplicaciones CGI más rápidas?, Apache también tiene una solución para esto. Utilice el modulo *mod _fcgi* para implementar un entorno FastCGI dentro de Apache y haga que sus aplicaciones FastCGI arranquen rápidamente.
- **Soporte de host virtuales:** Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales.
- **Soporte de autentificación HTTP:** Apache soporta autentificación básica, basada en la Web. También está preparado para autentificación basada en la digestión de mensajes, que es algo que los navegadores Web populares ya han implementado. Apache puede implementar autentificación básica, utilizando tanto archivos estándar de contraseña

como: los DBM, llamadas a SQL o llamadas a programas externos de autentificación.

- **Perl integrado:** Perl se ha convertido en el estándar para la programación de scripts CGI. Apache es seguramente uno de los factores que hacen de Perl un lenguaje de programación CGI tan popular. Apache se encuentra más cerca de Perl que nunca. Puede bajar un script CGI basado en Perl a la memoria utilizando su modulo *mod_perl* y reutilizarlo tantas veces como necesite. Este proceso, elimina las desventajas del arranque, que se encuentran asociadas a menudo con los lenguajes de interpretación como Perl.
- **Soporte de scripts PHP:** este lenguaje de script ha comenzado a ser muy utilizado y Apache ofrece un amplio soporte de PHP utilizando el modulo *mod php*.
- **Soporte de servlets de Java:** los servlets de Java y las Java Server Pages (JSP), se están convirtiendo en algo muy común en los sitios Web dinámicos. Puede ejecutar servlets de Java utilizando el premiado entorno Tomcat con Apache.

- **Servidor Proxy integrado:** Puede convertir Apache en un servidor Proxy cache. Sin embargo, la implementación actual del modulo opcional de Proxy, no soporta HTTP Proxy o el ultimo protocolo HTTP 1.1. Se esta planeando actualizar este modulo muy pronto.
- **Estado del servidor y adaptación de registros:** Apache le da una gran cantidad de flexibilidad en el registro y la monitorización del estado del servidor. El estado del servidor puede monitorizarse mediante un navegador Web. Además, puede adaptar sus archivos de registro a su gusto.
- **Soporte de Server Side Includes (SSI):** Apache ofrece un conjunto de Server Side Incluyes, que añaden una gran cantidad de flexibilidad para el desarrollador del sitio Web.
- **Soporte de Secured Sockect Layer (SSL):** puede crear fácilmente un sitio Web SSL utilizando OpenSSL y el modulo *mod_ssl* de Apache.

Servidor de base de datos

El manejador de Base de Datos relacional, que ayudará a manejar la gran cantidad de información del portal será Oracle 9i, el cual, va a brindar un gran desempeño en consultas, así también, en seguridad de datos críticos y generales de los clientes a quienes se dará servicio de pagos en el Portal.

Oracle, ofrece este RDBMS como un producto incorporado a la línea de producción. Incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios. Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo.

Se detallará unas pocas características del sistema Oracle como manejador de Información:

- Oracle soporta dos tipos de almacenamiento, por caracter (RAW) o por bloques (Files System), generalmente es recomendable que los sean colocados en Raw Device.

Raw Device: es un dispositivo de caracteres disponibles en algunos sistemas operativos, el cual es asignado directamente a Oracle.

Oracle corre más rápidamente con Raw Device que con Files System, por varias razones:

- El I/O (Input/Output) es realizado directamente en el disco por Oracle, independientemente del sistema operativo.
- El buffer caché del sistema operativo es dejado a un lado.
- Los buffers del sistema operativo y de Oracle son independientes entre sí.

- Con la intención de evitar la contención de los discos, se debe considerar la instalación de Oracle en dispositivos separados, especialmente si se tienen varios discos, y esencialmente, si se posee más de una controladora de disco. La planeación debe realizarse teniendo en cuenta los siguientes criterios:

  - Los Files System y sus dispositivos asignados.

- El swapping y paginamiento en Oracle, deberán estar en los dispositivos más rápidos.
- Los tablespace para tablas e índices en dispositivos separados.
- Los Log Files en un dispositivo separado al del tablespace de RDBMS Oracle.

- El uso de memoria en el RDBMS Oracle tiene como propósito lo siguiente:
  - Almacenar los códigos de los programas para empezar a ejecutarse.
  - Almacenar los datos necesarios durante la ejecución de un programa.
  - Almacenar información sobre cómo es la transferencia entre procesos y periféricos.

Servidor de nombres de dominio (DNS)

El Servidor que va a proveer la resolución de nombres en la red local es el Bind, el mismo da una variedad de ventajas debido a características tales como:

- BIND soporta Transferencias de zona incremental**,** (Incremental Zone Transfers, IXFR), donde un servidor de nombres sólo descargará las porciones actualizadas de una zona modificada en un servidor de nombres maestro.
- BIND puede proporcionar servicios de nombres en ambientes IP versión 6 (IPv6), a través del uso de registros de zona A6.
- Resolución de nombres a direcciones IP.
- Resolución inversa (de direcciones IP a nombres).
- Listas de control de acceso.
- Servidores secundarios.
- Transferencia segura de zonas, entre servidores primarios y secundarios (y puertos).
- Localización de servicios (registros SRV - RFC2052, del inglés, Request For Comments).
- Respuestas parametrizadas en función del origen de la petición (vistas).
- Uso de la herramienta rndc.
- Logs a medida.

Servidor de correo

Existe una gran variedad de programas de correo electrónico, que proveen al usuario de una aplicación para la creación y envío de mail. Estos programas son los llamados Agentes de Usuario o MUA (Mail User Agent), y su propósito es el aislar al usuario de los Agentes de Transporte o MTA (Mail Transport Agent), que son los encargados de transferir los mails a su correcto destino.

Sendmail es el agente de transporte de correo más común de Internet (en los sistemas UNIX). Aunque actúa principalmente como MTA, también puede ser utilizado como MUA (aunque no posee interfaz de usuario). Las misiones básicas de sendmail son las siguientes:

- Recogida de mails provenientes de un Mail User Agent (MUA), como pueden ser elm, Eudora o pine; o provenientes de un Mail Transport Agent (MTA) como puede ser el propio sendmail.
- Elección de la estrategia de reparto de los mails, basándose en la información de la dirección del destinatario contenida en la cabecera:

- Si el mail es local en el sistema, enviará el mail al programa de reparto local de mails.
- Si el mail no es local, sendmail utilizará el DNS del sistema para determinar el host al que debe ser enviado el mail. Para transferir el mensaje, iniciará una sesión SMTP con el MTA de dicho host.
- Si no es posible mandar el mail a su destino (porque la maquina receptora esta desconectada, o va muy lenta), sendmail almacenará los mails en una cola de correo, y volverá a intentar el envío del mail un tiempo después. Si el mail no puede ser enviado tras un tiempo razonable, el mail será devuelto a su autor con un mensaje de error. Sendmail debe garantizar que cada mensaje llegue correctamente a su destino, o si hay error este debe ser notificado (ningún mail debe perderse completamente).

- Reformatear el mail antes de pasarlo a la siguiente máquina, según unas reglas de reescritura. Según el

tipo de conexión que poseamos con una determinada máquina, o según, el agente de transporte al que vaya dirigido el mail, se necesitará cambiar los formatos de las direcciones del remitente y del destinatario, algunas líneas de la cabecera del mail, o incluso puede que se necesite añadir alguna línea a la cabecera. Sendmail debe realizar todas estas tareas para conseguir la máxima compatibilidad entre usuarios distintos.

- Otra función muy importante de sendmail es permitir el uso de "alias" entre los usuarios del sistema; lo que permitirá (entre otras funciones) crear y mantener listas de correo entre grupos.
- Ejecución como agente de usuario (MUA). Aunque no posee interfaz de usuario, sendmail también permite el envío directo de mails a través de su ejecutable.

Todas estas características y muchas otras que posee el sendmail deben ser configuradas y varían de unos sistemas a otros. Para configurarlas se utiliza el fichero de configuración de sendmail (sendmail.mc). La revisión y modificación de este fichero es bastante complicada y necesita de una serie de

conocimientos previos. Sendmail utiliza el archivo sendmail.cf para su ejecución, que es creado a partir del archivo de configuración, mencionado anteriormente.

Firewall

Muchas son hoy en día, las personas que se conectan, de una manera u otra, a Internet. Desde empresas que operan en la red hasta personas en sus casas, que pasan un rato divertido navegando por sus páginas preferidas.

Pero pocas de estas personas entienden realmente, las consecuencias, que tiene el abrir sus sistemas informáticos a Internet, unas consecuencias, que no sólo son de carácter benigno o incluso beneficioso. El bien que obtenemos de Internet tiene un precio: Internet no es un lugar seguro.

Es común entre los navegantes más o menos habituales de Internet, que nunca han tenido, o mejor dicho, creen que nunca han tenido un problema de seguridad en sus sistemas, el pensar que no es probable que lleguen jamás a recibir uno de estos ataques, por el simple hecho de no poseer nada de

interés, de no ser nadie importante. Esto es, claramente, falso. Cualquiera, puede ser presa de un ataque en la Red, cualquiera, por insignificante que se pueda pensar que uno es.

Por tanto, una vez visto que el peligro existe, es la hora de hablar de qué es un ***firewall***.

Un firewall es, por lo general, un software (puede ser también un equipo hardware dedicado), a través del cual nos conectamos a una red como Internet, y que sirve como filtro sobre el tráfico que por él pasa, en ambas direcciones, y que en un momento dado, puede rechazar cierto tráfico en alguna de las direcciones. Eso quiere decir que, mediante un firewall, se puede detectar el tráfico no deseado hacia los sistemas, y en general, los posibles ataques de que seamos objeto. De esta manera podremos aislar los equipos del exterior, permitiendo el uso de Internet de manera absolutamente normal, pero minimizando en lo posible la probabilidad de padecer las consecuencias de un ataque.

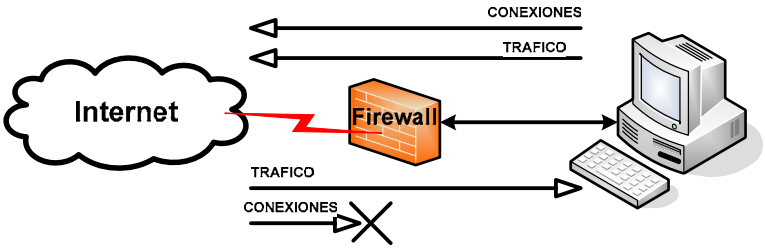


Figura 3.3: Típica disposición de un firewall doméstico

Para el uso del Firewall, vamos a usar un paquete que viene en los Sistemas Operativos Red Hat Linux actuales, que es Iptables, a continuación se dan unas características del paquete Iptables.

Iptables, es la herramienta que permite configurar las reglas del sistema de filtrado de paquetes del kernel de Linux. Con esta herramienta, se podrá crear un firewall adaptado a las necesidades de cada red.

Su funcionamiento es simple: a iptables se le proporcionan unas reglas, especificando a cada una de ellas determinadas características que debe cumplir un paquete. Además, se especifica para esa regla una acción o target. Las reglas tienen un orden, y cuando se recibe o se envía un paquete, las reglas se recorren en orden hasta que las condiciones que

pide una de ellas se cumplen en el paquete y la regla se activa, realizando sobre el paquete, la acción que le haya sido especificada.

Estas acciones, se plasman en los que se denominan targets, que indican lo que se debe hacer con el paquete. Los más usados son bastante explícitos: ACCEPT, DROP y REJECT, pero también hay otros, que nos permiten funcionalidades añadidas y algunas veces interesantes: LOG, MIRROR.

En cuanto a los paquetes, el total del sistema de filtrado de paquetes del kernel se divide en tres tablas, cada una con varias cadenas a las que puede pertenecer un paquete, de la siguiente manera:

- **filter**: Tabla por defecto, para los paquetes que se refieran a nuestra máquina.
    - **INPUT**: Paquetes recibidos para nuestro sistema.
    - **FORWARD**: Paquetes enrutados a través de nuestro sistema.

- **OUTPUT**: Paquetes generados en nuestro sistema y que son enviados.

- **nat**: Tabla referida a los paquetes enrutados en un sistema con Masquerading
  - **PREROUTING:** Para alterar los paquetes según entren.
  - **OUTPUT:** Para alterar paquetes generados localmente antes de enrutar.
  - **POSTROUTING:** Para alterar los paquetes cuando están a punto para salir.

- **mangle**: Alteraciones más especiales de paquetes
  - **PREROUTING:** Para alterar los paquetes entrantes antes de enrutar.
  - **OUTPUT:** Para alterar los paquetes generados localmente antes de enrutar.

Dado, que el soporte para el firewall, está integrado en el kernel de Red Hat Linux (Netfilter), para poder usar iptables hay que asegurarse de que el núcleo admite el uso de iptables y que se añade a la configuración del núcleo, todos

aquellos targets que se vaya a necesitar (aunque siempre es bueno tener los más posibles).

Red virtual privada (VPN)

Otra herramienta usada para en el desarrollo del Portal, es la creación de una VPN, el protocolo usado para nuestra Red Virtual es IPSec, la cual viene embebida en el paquete freesWan.

IPSec, es una extensión al protocolo IP, que proporciona seguridad a IP y a los protocolos de capas superiores. Fue desarrollado para el nuevo estándar IPv6 y después fue portado a IPv4.

IPSec, emplea dos protocolos diferentes - AH y ESP - para asegurar la autenticación, integridad y confidencialidad de la comunicación. Puede proteger el datagrama IP completo o sólo los protocolos de capas superiores. Estos modos se denominan, respectivamente, modo túnel y modo transporte.

En modo túnel, el datagrama IP, se encapsula completamente dentro de un nuevo datagrama IP, que emplea el protocolo IPSec. En modo transporte, IPSec sólo maneja la carga del datagrama IP, insertándose la cabecera IPSec, entre la cabecera IP y la cabecera del protocolo de capas superiores.

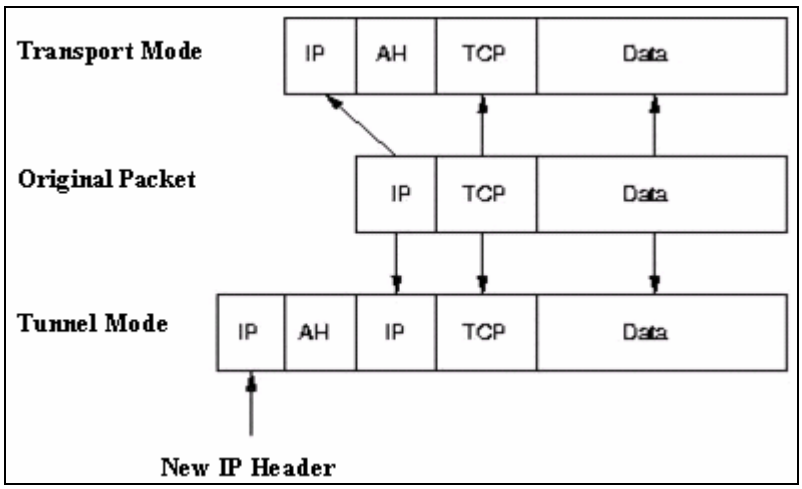


Figura 3.4: Modos Túnel y Transporte

Para proteger la integridad de los datagramas IP, los protocolos IPSec, emplean códigos de autenticación de mensaje, basados en resúmenes (HMAC - Hash Message Authentication Codes). Para el cálculo de estos HMAC los protocolos HMAC, emplean algoritmos de resumen como MD5 y SHA para calcular un resumen basado en una clave

secreta y en los contenidos del datagrama IP. El HMAC, se incluye en la cabecera del protocolo IPSec y el receptor del paquete puede comprobar el HMAC si tiene acceso a la clave secreta.

Para proteger la confidencialidad de lo datagramas IP, los protocolos IPSec, emplean algoritmos estándar de cifrado simétrico. El estándar IPSec, exige la implementación de NULL y DES. En la actualidad, se suelen emplear algoritmos más fuertes: 3DES, AES y Blowfish.

Para protegerse, contra ataques por denegación de servicio, los protocolos IPSec, emplean ventanas deslizantes. Cada paquete recibe un número de secuencia y sólo se acepta su recepción si el número de paquete se encuentra dentro de la ventana o es posterior. Los paquetes anteriores, son descartados inmediatamente. Esta es una medida de protección eficaz, contra ataques por repetición de mensajes, en los que el atacante, almacena los paquetes originales y los reproduce posteriormente.

Para que los participantes de una comunicación, puedan encapsular y desencapsular los paquetes IPSec, se necesitan mecanismos para almacenar las claves secretas, algoritmos y direcciones IP involucradas en la comunicación. Todos estos parámetros se almacenan en asociaciones de seguridad (SA - Security Associations). Las asociaciones de seguridad, a su vez, se almacenan en bases de datos de asociaciones de seguridad (SAD - Security Asocciation Databases).

Cada asociación de seguridad define los siguientes parámetros:

- Dirección IP origen y destino de la cabecera IPSec resultante. Estas son las direcciones IP de los participantes de la comunicación IPSec, que protegen los paquetes.
- Protocolo IPSec (AH o ESP). A veces, se permite compresión (IPCOMP).
- El algoritmo y clave secreta empleados por el protocolo IPSec.

- Índice de parámetro de seguridad (SPI - Security Parameter Index). Es un número de 32 bits que identifica la asociación de seguridad.

Algunas implementaciones de la base de datos de asociaciones de seguridad, permiten almacenar más parámetros:

- Modo IPSec (túnel o transporte)
- Tamaño de la ventana deslizante, para protegerse de ataques por repetición.
- Tiempo de vida de una asociación de seguridad.

En una asociación de seguridad, se definen las direcciones IP de origen y destino de la comunicación. Por ello, mediante una única SA, sólo se puede proteger un sentido del tráfico en una comunicación IPSec full duplex. Para proteger ambos sentidos de la comunicación, IPSec necesita de dos asociaciones de seguridad unidireccionales.

Las asociaciones de seguridad, sólo especifican, cómo se supone que IPSec protegerá el tráfico. Para definir qué tráfico proteger, y cuándo hacerlo, se necesita información adicional. Esta información se almacena en la política de seguridad (SP - Security Policy), que a su vez, se almacena en la base de datos de políticas de seguridad (SPD - Security Policy Database).

Una política de seguridad, suele especificar los siguientes parámetros:

- Direcciones de origen y destino de los paquetes por proteger. En modo transportes, estas serán las mismas direcciones que en la SA. En modo túnel, pueden ser distintas.
- Protocolos y puertos a proteger. Algunas implementaciones, no permiten la definición de protocolos específicos a proteger. En este caso, se protege todo el tráfico entre las direcciones IP indicadas.
- La asociación de seguridad a emplear para proteger los paquetes.

Servidor Proxy

Controlar el acceso y acelerar la navegación a los sitios visitados en Internet, es una necesidad real para el óptimo uso del ancho de banda.

Para lograrlo, se necesita de un Proxy Server, uno de los más usados es Squid, el cual, tiene ciertas ventajas y características:

- Controles de acceso extensivos, lo que permite controlar el acceso de los clientes de su organización a Internet, ya sea usando: restricciones por usuario (login y password del cliente), restricciones por dirección IP de la estación cliente o restricciones de navegación por horarios.
- Caché Transparente, lo cual permitirá, que los usuarios naveguen por Internet, sin saber que la navegaciones controlada desde un punto central de la red.
- Restricción en la descarga de archivos por parte de los clientes.

- Restricción en la navegación de sitios Web específicos, como: páginas que atenten contra la moral, sitios de Chat, etc.
- Squid, es software de libre distribución, esto no sólo implica que sea gratuito, sino que además al estar su código fuente libremente accesible y modificable, hace posible la modificación del programa, en caso de querer mejorar sus prestaciones o características.
- Squid, tiene un rendimiento superior a las demás implementaciones de proxy-cachés implementadas por software.
- Squid, soporta el protocolo ICP, para integrar las cachés en grupos colaborativos, muchas de las demás implementaciones no soportan este punto. ICP permite además de formar jerarquía, hacer que el fallo de una caché, sea superado por la colaboración de las otras.
- Squid, soporta además otros mecanismos de comunicación entre cachés más eficientes, como las Caché-Digest, por el cual las cachés intercambian cada cierto tiempo (una hora), un mapa de bits de la

tabla de digests MD5 del hash, que contiene todas las URLs de cada caché.

- Squid se ha popularizado tanto, que ya hay muchas aportaciones para tratamiento de estadísticas, administración por Web y muchas otras; además su grupo de desarrolladores, es tan grande que se incorporan continuamente nuevas funcionalidades de acuerdo con las últimas tendencias en tecnología de Redes y sistemas de información (Ej.: multicasting)
- Squid, es utilizado en las grandes agrupaciones de cachés mundiales, como la NLANR.
- Squid, compila muy fácilmente en casi cualquier plataforma Unix y su instalación es bastante simple.

Detector de Intrusos (IDS)

Un IDS o Sistema de Detección de Intrusos, es una herramienta de seguridad, que intenta detectar o monitorizar los eventos ocurridos en un determinado sistema informático o red informática, en busca de intentos de comprometer la seguridad de dicho sistema.

Entre las principales características de los IDS tenemos:

- Los IDS, buscan patrones previamente definidos, que impliquen cualquier tipo de actividad sospechosa o maliciosa sobre nuestra red o host.
- Los IDS, aportan a nuestra seguridad una capacidad de prevención y de alerta anticipada ante cualquier actividad sospechosa. No están diseñados para detener un ataque, aunque sí pueden generar ciertos tipos de respuesta ante éstos.
- Los IDS, aumentan la seguridad de nuestro sistema, vigilan el tráfico de nuestra red, examinan los paquetes analizándolos en busca de datos sospechosos y detectan las primeras fases de cualquier ataque, como pueden ser: el análisis de nuestra red, barrido de puertos, etc.

Tipos de IDS

Según sus características hay 3 tipos de IDS.

- **HIDS (HostIDS)**: Protege contra un único Servidor, PC o host. Monitorizan gran cantidad de eventos, analizando actividades con una gran precisión,

determinando de esta manera, qué procesos y usuarios se involucran en una determinada acción. Recaban información del sistema, como: ficheros, logs, recursos, etc, para su posterior análisis en busca de posibles incidencias. Todo ello en modo local, dentro del propio sistema. Fueron los primeros IDS desarrollados por la industria de la seguridad informática.

- **NIDS (Net IDS):** Protege un sistema basado en red. Actúan sobre una red, capturando y analizando paquetes de red, es decir, son sniffers del tráfico de red. Luego analizan los paquetes capturados, buscando patrones que supongan algún tipo de ataque.

  Bien ubicados, pueden analizar grandes redes y su impacto en el tráfico suele ser pequeño. Actúan mediante la utilización de un dispositivo de red, configurado en modo promiscuo (analizan,"ven" todos los paquetes que circulan por un segmento de red, aunque estos no vayan dirigidos a un determinado

equipo). Analizan el tráfico de red, normalmente, en tiempo real. No sólo trabajan a nivel TCP/IP, también lo pueden hacer a nivel de aplicación. A este tipo de IDS pertenece snort.

- **DNIDS:** Este tipo de IDS, más que proteger, monitoriza la actividad entre varias redes. Tiene una visión global.

3.4. Análisis de Seguridad

Antes de explicar los mecanismos de seguridad utilizados en la implementación del portal, se detallarán algunos conceptos que deben ser tomados en cuenta al implementar cualquier red.

3.4.1. Conceptos sobre seguridad Informática

En el apéndice B, se describen algunos conceptos importantes sobre seguridad informática que han sido aplicados en el proyecto.

3.4.2. Análisis de Seguridad del sistema

Como mecanismo de protección se ha decidido usar dos firewalls, uno externo y otro interno, para controlar el acceso a los recursos del sistema. El firewall externo, controla que los usuarios solo accedan a los servicios ofrecidos por los servidores que se encuentran en la DMZ. El firewall interno controla el acceso a la base de datos; ningún usuario, ya sea de Internet o de la red interna, pueden tener acceso a la base de datos.

Los datos que van del servidor Web a la base de datos y viceversa, viajan a través de un canal seguro, provisto por una red virtual privada configurada entre el firewall externo y el firewall interno, con esto, se previene que los usuarios internos, tengan acceso a la información que viaja hacia y desde la base de datos, o al menos, el grado de dificultad para descifrarla sea mayor.

El servidor Web, tiene un certificado, este certificado contiene la clave pública que requiere un cliente para desencriptar la información, además contiene datos sobre la empresa y la firma digital de la autoridad, que certifica la validez de los mismos.

Se ha utilizado el protocolo de seguridad SSL, para proteger la información que viaja desde el servidor Web hasta el cliente.

En el apéndice B, se describe cómo funcionan los protocolos de seguridad; según lo explicado, el protocolo SET es el más seguro para sistemas que requieran pagos con tarjeta de crédito, pero debido a la complejidad para implementarlo se ha decidido utilizar el protocolo SSL, que es más conocido y ampliamente utilizado, pese a sus limitaciones.

El acceso a Internet, por parte de los usuarios de la red interna, es controlado por medio de un Proxy, el mismo requiere que el usuario se autentique para saber si tiene o no permiso para hacerlo.

En el servidor de base de datos, se ha colocado un sistema de detección de intrusos (IDS), que se encarga de registrar cualquier evento inusual en este servidor.

Los equipos de la red interna también están protegidos con un antivirus.

CAPITULO 4

4. DISEÑO DEL SISTEMA

En este capítulo, hablamos de la arquitectura del sistema. También describimos los casos de uso, mostramos un diagrama para cada uno de ellos y los diagramas de secuencia. Estos diagramas han sido construidos usando la notación de UML.

También describimos la estructura de la base de datos y mostramos gráficamente cada una de las tablas y sus relaciones

4.1. Arquitectura del sistema

La arquitectura del sistema está basada en un modelo de n capas, las mismas que han sido divididas de la siguiente manera:

- Capa de presentación (páginas html)
- Capa del medio (páginas JSP)
- Capa de negocios (Java Beans)
- Capa de datos (Base de datos)

Utilizar una arquitectura de n capas tiene algunas ventajas como:

- Cada capa es independiente de las demás, si se modifica una capa, las otras no se verán afectadas.
- Los cambios en la interfaz de usuario se los hace solo en el servidor de aplicaciones, y no requiere de modificaciones en el equipo del usuario final.
- El código puede ser reutilizado.
- Al tener una capa de datos independiente, permite balancear la carga.

Como plataforma de desarrollo nos hemos basado en el estándar J2EE. En el siguiente diagrama se muestran las capas y las herramientas de desarrollo utilizadas en cada una:

Figura 4.1. Capas de la Arquitectura del sistema.

En el siguiente diagrama mostramos la arquitectura del sistema:

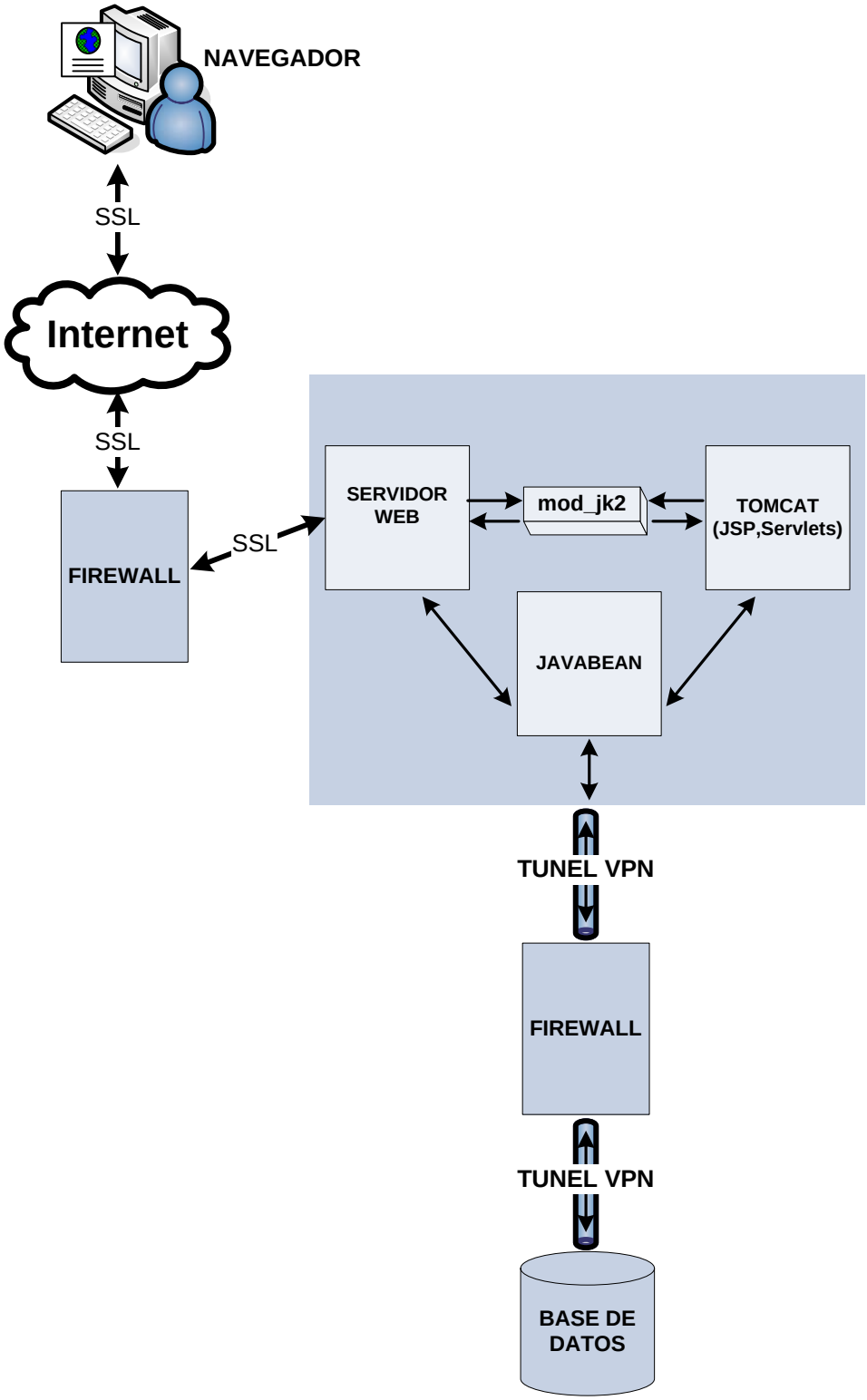


Figura 4.2. Arquitectura del Sistema.

El sistema funciona de la siguiente manera:

1. El usuario, a través de un navegador hace una petición al servidor web.
2. El servidor web, dependiendo de la página solicitada (html o jsp), se comunica con Tomcat, que es el contenedor de páginas jsp a través de la librería mod_jk2.
3. Tomcat se sirve de JavaBeans para conectarse a la base de datos, antes de atender el requerimiento del usuario.
4. Los datos son entregados a Tomcat por medio de los JavaBeans.
5. Tomcat responde al servidor web con una página html o jsp, según sea el caso.
6. Finalmente el servidor web envía la respuesta al usuario final, esta respuesta puede ser una página html o jsp.

El diagrama físico de la red es presentado en la siguiente figura:

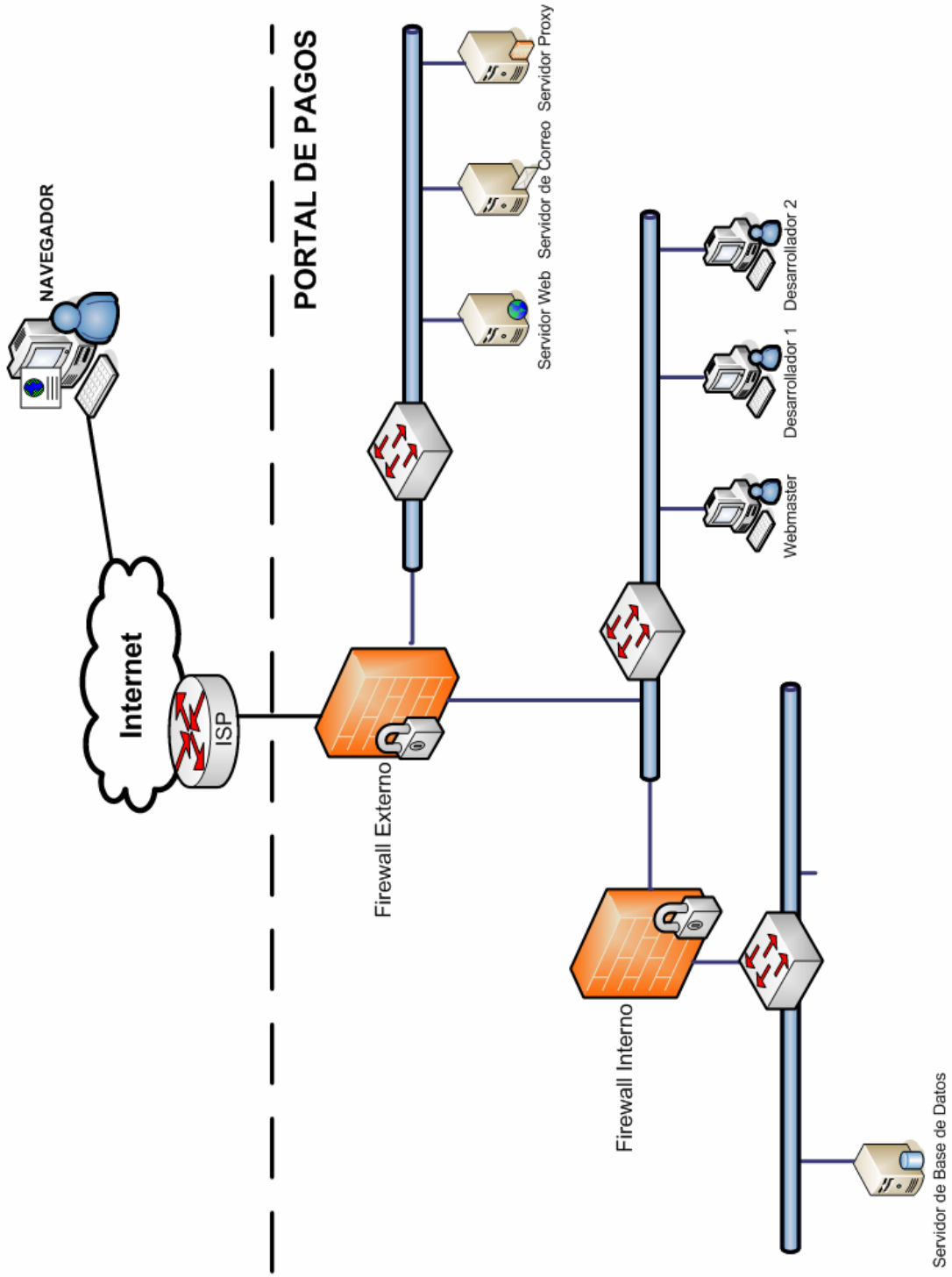


Figura 4.3. Diagrama Físico de la red.

4.2. Casos de uso

Los casos de uso nos ayudarán a entender mejor el comportamiento del sistema, describen de forma clara la funcionalidad del mismo, independiente de la implementación. Los casos de uso definen las relaciones entre el sistema y el entorno.

Los símbolos utilizados para realizar los diagramas de casos de uso son:

- Caso de Uso

  Está representado por una elipse, y debe contener el nombre del caso de uso.

Figura 4.4. Símbolo utilizado para representar un caso de uso

- Actor

  Es quien interactúa con el sistema. Se los representa con un muñeco.

Figura 4.5. Símbolo utilizado para representar un actor

- Frontera del Sistema

  Permite identificar los elementos externos e internos del sistema; además, permite identificar las responsabilidades. El ambiente externo está conformado, únicamente, por los actores. Está representada por un cuadro.

Figura 4.6. Símbolo utilizado para la frontera del sistema

A continuación hacemos una descripción de cada uno de los casos de uso.

Caso de uso: **Iniciar sesión**

Actores: Usuario

Propósito: Ingreso del usuario al sistema

Resumen: El usuario ingresa los datos de inicio de sesión: usuario y contraseña, el sistema verifica que estos datos sean válidos y permite el ingreso.

Curso normal de eventos

Acción de los actores	Respuesta del sistema
1. El usuario ingresa los datos de inicio de sesión: usuario y contraseña	**2.** El sistema encripta la contraseña y realiza la autenticación.
	3. El usuario ingresa al sistema

Cursos alternos de eventos:

Sección 2: Se introduce usuario o contraseña incorrecta, se envía un mensaje de error.

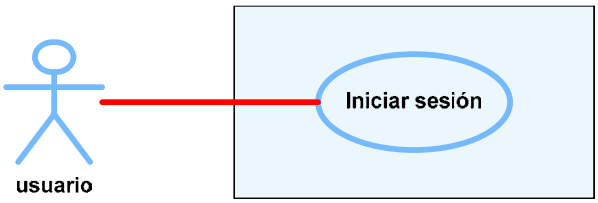


Figura 4.7. Caso de uso Iniciar sesión

Caso de uso: **Registrar usuario**

Actores: Usuario

Propósito: Registro de un nuevo usuario para poder acceder al sistema.

Resumen: El usuario presiona el botón para registrarse, el sistema le muestra el formulario de registro, después de validar los datos de registro el usuario queda registrado.

Curso normal de eventos

Acción de los actores	Respuesta del sistema
1. El usuario presiona el botón para registrarse.	**2.** El sistema le muestra el formulario de registro para usuarios nuevos.

3. El usuario ingresa los datos requeridos para registrarse y poder acceder al sistema de pagos.

4. El sistema valida y graba la información ingresada por el usuario.

5. El sistema muestra mensaje de usuario registrado.

Cursos alternos de eventos:

Sección 4: Se introduce en forma inválida algún ítem del formulario de registro, o usuario ya existe en el sistema. Se envía un mensaje error.

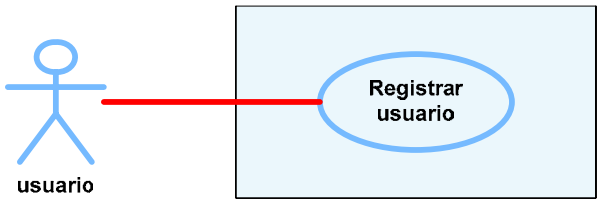


Figura 4.8. Caso de uso Registrar usuario

Caso de uso: **Consultar planilla**

Actores: Usuario

Propósito: Consulta de los datos de una planilla

Resumen: Después de ingresar al sistema, el usuario selecciona, de la barra de navegación vertical, la opción *consultas,* del submenú que se despliega debe seleccionar entre las opciones: agua, luz, teléfono. Luego de seleccionar una de estas opciones, el sistema muestra los resultados.

Curso normal de eventos

Acción de los actores	Respuesta del sistema
1. Selecciona del menú vertical la opción *consultas*	**2.** Despliega un submenú con las opciones: agua, luz, teléfono
3. Selecciona una de las opciones: agua, luz, teléfono	**4.** Despliega información de las planillas según la opción elegida: agua, luz, teléfono

Cursos alternos de eventos:

Sección 4: No hay planillas asociadas al usuario, no muestra información.

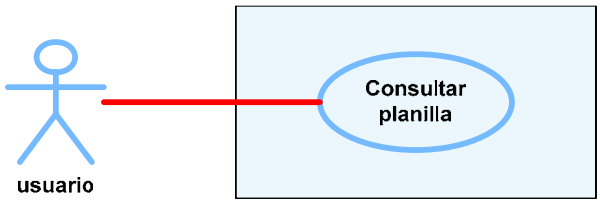


Figura 4.9. Caso de uso Consultar planilla

Caso de uso:	**Pagar servicio básico**
Actores:	Usuario, Emisor de tarjetas, Banco El usuario es quien inicia el sistema
Propósito:	Pagar una planilla de alguno de los servicios básicos: agua, luz, teléfono
Resumen:	Después de ingresar al sistema, el usuario selecciona de la barra de navegación vertical, la opción *pagos,* del submenú que se despliega, debe seleccionar entre las opciones: agua, luz, teléfono. Luego de seleccionar una de estas opciones, el sistema muestra los datos más relevantes de las planillas que el usuario tiene pendiente. El usuario selecciona la opción pagar planilla y el sistema muestra un formulario de ingreso de datos de la tarjeta con la que va a realizar el pago. El usuario ingresa los datos de la tarjeta y el sistema

envía los datos a la entidad que corresponda (emisor de la tarjeta o banco), para que valide los datos; ésta se encarga de debitar el total del pago de la cuenta del usuario y envía una notificación al sistema de éxito o fracaso del pago. El sistema informa al usuario del éxito del pago presentándole un mensaje y enviándole un correo.

Curso normal de eventos

	Acción de los actores		Respuesta del sistema
1.	El usuario selecciona del menú vertical la opción *pagos*	**2.**	El sistema despliega un submenú con las opciones: agua, luz, teléfono
3	El usuario selecciona una de las opciones: agua, luz, teléfono	**4.**	El sistema despliega información principal de las planillas según la opción elegida: agua, luz, teléfono
5.	El usuario selecciona la opción pagar planilla	**6.**	El sistema muestra un mensaje que indica el total que debe cancelar de la planilla. También muestra el costo adicional de la transacción y pregunta si

	acepta ese costo
7. El usuario acepta el costo adicional de la transacción	**8.** El sistema muestra un formulario para que ingrese datos de la tarjeta con la que desea cancelar
9. El usuario escoge la forma de pago a. Tarjeta de débito (*ver página 114*) b. Tarjeta de crédito (*ver página 115*)	
	10. El sistema registra el pago
	11. El sistema envía un correo electrónico al usuario, informándole sobre el pago que acaba de realizar.

Cursos alternos de eventos:

Sección 4: No hay planillas asociadas al usuario, no muestra información.

Sección 5: El usuario no selecciona la opción cancelar planilla. El sistema debe esperar a que el usuario inicie algún evento.

Sección 7: El usuario no acepta el costo de la transacción y cancela el pago. El sistema debe esperar a que el usuario inicie algún evento.

Sección 9: El usuario no selecciona ninguna forma de pago y cancela, el sistema retorna a la ventana de pago.

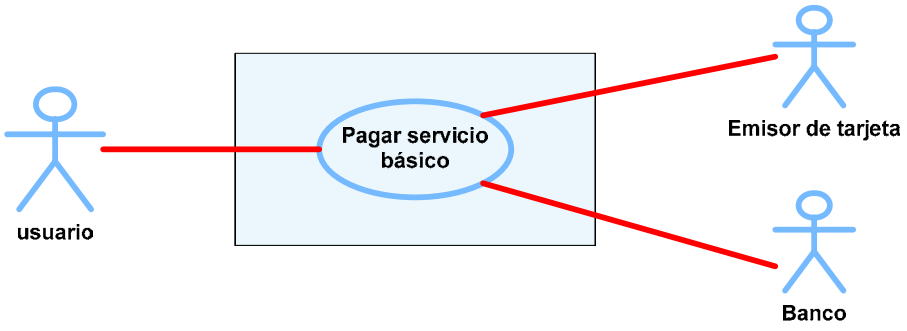


Figura 4.10. Caso de uso Pagar servicio básico

Caso de uso: Visualizar planilla

Actores: Usuario, Emisor de tarjetas, Banco

El usuario es quien inicia el sistema

Propósito: Imprimir una planilla de alguno de los servicios básicos: agua, luz, teléfono

Resumen: Este caso de uso se inicia después de seleccionar, de la barra de navegación vertical, la opción *consulta* y del

submenú una de las siguientes opciones: agua, luz, teléfono. Luego el usuario selecciona la opción visualizar planilla y el sistema pregunta si acepta el costo de esa transacción, posteriormente se muestra la imagen de la planilla con todos sus datos.

Curso normal de eventos

Acción de los actores	Respuesta del sistema
1. Selecciona del menú vertical la opción *consultas.*	**2.** Despliega un submenú con las opciones: agua, luz, teléfono
3. Selecciona una de las opciones: agua, luz, teléfono.	**4.** Despliega información de las planillas, según la opción elegida: agua, luz, teléfono.
5. Selecciona la opción visualizar planilla.	**6.** Pregunta si acepta pagar el costo de la transacción.
7. El usuario acepta el costo de la transacción.	**8.** El sistema muestra un formulario, para que ingrese datos de la tarjeta con la que desea cancelar.
9. El usuario escoge la forma de pago: a. Tarjeta de débito (*ver*	

página 107)

b. Tarjeta de crédito (*ver página 108*)

10. El sistema registra el pago

11. El sistema envía un correo electrónico al usuario, informándole sobre el pago que acaba de realizar

Cursos alternos de eventos:

Sección 4: No hay planillas asociadas al usuario, no muestra información.

Sección 5: El usuario no selecciona la opción visualizar planilla, el sistema espera a que el usuario seleccione alguna opción.

Sección 7: El usuario no acepta el costo de la transacción y cancela el pago. El sistema debe esperar a que el usuario inicie algún evento.

Sección 9: El usuario no selecciona ninguna forma de pago y cancela, el sistema retorna a la ventana de pago.

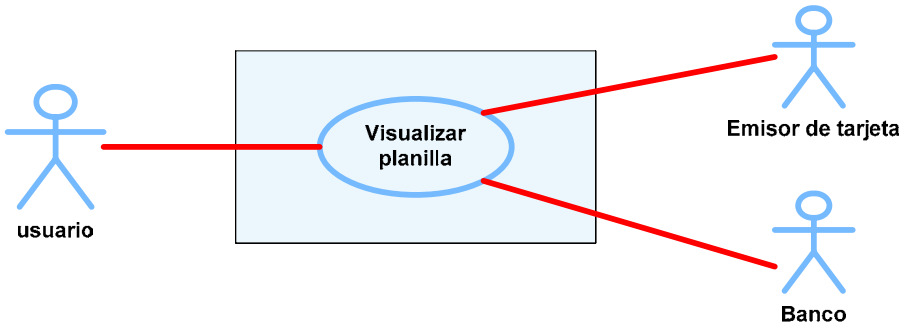


Figura 4.11. Visualizar planilla

Sección 9a: Pagar con tarjeta de débito

Curso normal de eventos

Acción de los actores	Respuesta del sistema
1. El usuario selecciona la opción de pagar con tarjeta de débito. Acepta el pago.	**2.** El sistema envía los datos de la tarjeta al banco emisor.
3. El banco verifica que los datos sean válidos, y debita el total de la cuenta del usuario.	
4. El banco envía un mensaje al sistema, indicando que el pago se realizó con éxito.	**5.** El sistema presenta un mensaje indicando que el pago fue realizado correctamente.

Cursos alternos de eventos:

Sección 1: El usuario no acepta el costo de la transacción. El sistema

debe esperar a que el usuario inicie algún evento.

Sección 3: Los datos de la tarjeta no son válidos, envía un mensaje informando que no se puede realizar el débito. Termina la transacción de pago

Sección 5: Se presenta un mensaje al usuario, indicando que el pago no se realizó.

Sección 9b: Pagar con tarjeta de crédito

Curso normal de eventos

Acción de los actores	Respuesta del sistema
1. El usuario selecciona la opción de pagar con tarjeta de crédito. Acepta el pago.	**2.** El sistema envía los datos de la tarjeta al emisor de la tarjeta
3. El emisor de la tarjeta verifica que los datos sean válidos y aprueba el crédito.	
4. El emisor envía un mensaje al sistema, indicando que el crédito fue aprobado.	**5.** El sistema presenta un mensaje indicando que el pago fue realizado correctamente.

Cursos alternos de eventos:

Sección 1: El usuario no acepta el costo de la transacción. El sistema

debe esperar a que el usuario inicie algún evento.

Sección 3: Los datos de la tarjeta no son válidos, envía un mensaje informando que no se aprueba el crédito. Termina la transacción de pago.

Sección 5: Se presenta un mensaje al usuario, indicando que el pago no se realizó.

Caso de uso: **Salir del sistema**

Actores: Usuario

Propósito: Salir del sistema

Resumen: El usuario presiona el botón salir, el sistema elimina los datos del usuario almacenados en memoria y carga la página principal.

Curso normal de eventos

Acción de los actores	Respuesta del sistema
1. El usuario presiona el botón salir del sistema	**2.** El sistema elimina los datos del usuario cargados en memoria. **3.** El sistema carga la página principal.

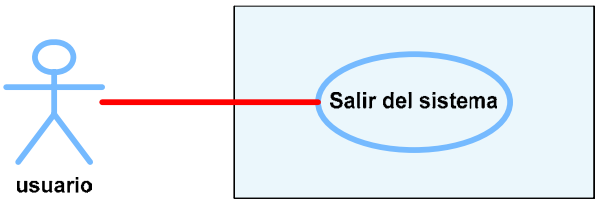


Figura 4.12. Salir del sistema

4.3. Diagramas de secuencia

Los diagramas de secuencia permiten modelar las interacciones entre objetos y los eventos que fluyen entre ellos, ordenados según su generación en el tiempo.

Los diagramas de secuencia, describen un escenario de un caso de uso de forma detallada; permiten representar los eventos generados por los actores, los eventos generados por el sistema y el orden en que estos eventos son generados.

Para construir los diagramas de secuencia se utiliza la siguiente simbología:

- Línea de vida de un objeto

  Representa la línea de vida del objeto durante la interacción. En la parte superior de la línea está el nombre del objeto, escrito dentro de un rectángulo.

Figura 4.13. Símbolo utilizado para representar la línea de vida de un objeto en un diagrama de secuencia.

- Activación

  Representa el tiempo en que un objeto existe. Es representado con un rectángulo sobre la línea de vida. El tiempo fluye desde arriba hacia abajo.

Figura 4.14. Símbolo utilizado para representar el tiempo de vida de un objeto en un diagrama de secuencia.

- Mensaje

  Para representar el envío de mensajes de un objeto a otro, se utiliza una línea sólida horizontal con una flecha apuntando al objeto que recibe el mensaje. Los mensajes son dibujados desde arriba hacia abajo, en el orden en que se van generando, llevan el nombre del mensaje y en algunos casos se incluyen los parámetros que se envía en el mensaje.

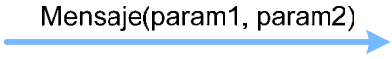


Figura 4.15. Símbolo utilizado para representar los mensajes entre objetos en un diagrama de secuencia.

Los diagramas de secuencia del sistema se los puede ver en el Apéndice C.

4.4. Diseño de la Base de Datos

La Base de nombre WEBSITE, que contiene la estructura del Portal de Pagos, tiene las siguientes tablas:

- **BW_BANCO**
- **BW_CIUDAD**
- **BW_CLIENTE**
- **BW_CONSULTA**
- **BW_CONTACTO**
- **BW_EMPRESA**
- **BW_PAGO**
- **BW_PROVINCIA**
- **BW_REGION**
- **BW_SERVBAS**
- **BW_SERVICIO**
- **BW_TARJETA**
- **BW_TIPCONS**
- **BW_USUARIO**

La tabla **BW_BANCO** almacenará los diferentes bancos, con los cuales nosotros ofrecemos el servicio de pagos en línea.

La tabla **BW_BANCO** tiene los siguientes campos:

- **BA_IDBANC:** Este campo es de tipo NUMERIC, el cual almacenará un ID para los bancos que usemos en nuestro portal.
- **BA_DESCRI:** Este campo es de tipo VARCHAR2, el cual almacenará el nombre del Banco.
- **BA_ABRBANC:** Este campo es de tipo VARCHAR2, el cual almacenará un código interno por el cual será conocido un banco específico.

La tabla **BW_CIUDAD** almacenará las diferentes ciudades del Ecuador, para las diferentes empresas que ofrecen el servicio de agua, luz o teléfono.

La tabla **BW_CIUDAD** tiene los siguientes campos:

- **CI_IDCIUD:** Este campo es de tipo NUMERIC, el cual almacenará un código para la ciudad respectiva.
- **CI_DESCRI:** Este campo es de tipo VARCHAR2, el cual almacenará el nombre de la ciudad.

- **CI_IDPROV:** Este campo es de tipo NUMERIC, el cual almacenará el código de la provincia al cual pertenece la ciudad.

La tabla **BW_CLIENTE** almacenará los datos principales de nuestros clientes que se hayan registrado y hayan obtenido un login y un password.

La tabla **BW_CLIENTE** tiene los siguientes campos:

- **CL_CEDRUC:** Este campo es de tipo VARCHAR2, el cual almacenará la cédula o RUC de nuestro cliente.
- **CL_APELLIDO:** Este campo es de tipo VARCHAR2, y almacenará el apellido de nuestro cliente.
- **CL_NOMBRE:** Este campo es de tipo VARCHAR2, y almacenará el nombre de nuestro cliente.
- **CL_DIRECCIÓN:** Este campo es de tipo VARCHAR2, y almacenará la dirección del cliente.
- **CL_EMAIL:** Este campo es de tipo VARCHAR2, y almacenará el correo electrónico del cliente.
- **CL_TELEFONO:** Este campo es de tipo VARCHAR2, y almacenará el teléfono del cliente.
- **CL_CODEMP:** Es tipo VARCHAR2, y almacena las empresas de servicios básicos en las que el cliente tiene una cuenta.

La tabla **BW_CONSULTA** almacenará todas las consultas, que nuestro usuario (cliente registrado) realice, acerca todo lo relacionado a pagos de sus servicios.

La tabla **BW_CONSULTA** tiene los siguientes campos:

- **CN_IDCON:** Este campo es de tipo NUMBER, y almacenará un código único para cada consulta que realice nuestros usuarios.
- **CN_FECHA:** Este campo es de tipo DATE, y almacenará la fecha en que se realizó la consulta.
- **CN_IDEMP:** Este campo es de tipo NUMBER, y almacenará la compañía a la cual nuestro usuario hizo su consulta.
- **CN_LOGIN:** Este campo es de tipo VARCHAR2, y almacenará el login del usuario con el cual realizó la consulta.
- **CN_TIPCONS:** Este campo es de tipo NUMBER, y almacenará un código de un tipo de las consultas que ofrecemos.

La tabla **BW_CONTACTO** almacenará los contactos de las diferentes empresas de los Servicio Básicos, definiendo como contacto a algún ejecutivo o personal de referencia para esa empresa.

La tabla **BW_CONTACTO** tiene los siguientes campos:

- **CO_APELLIDO:** Este campo es de tipo VARCHAR2, y almacenará el apellido del personal de contacto.
- **CO_EMAIL:** Este campo es de tipo VARCHAR2, y almacenará el email de nuestro contacto.
- **CO_IDCIUD:** Este campo es de tipo NUMBER, y almacenará el código de ciudad de nuestro contacto. Este código será el mismo que el código de ciudad de la empresa del servicio básico.
- **CO_IDCONTAC:** Este campo es de tipo NUMBER, y almacenará el código o id de nuestro contacto.
- **CO_IDEMP:** Este campo es de tipo NUMBER, y almacenará el código de la compañía.
- **CO_NOMBRE:** Este campo es de tipo VARCHAR2, y almacenará el nombre de nuestro contacto.
- **CO_TELEFONO:** Este campo es de tipo VARCHAR2, y almacenará el teléfono de nuestro contacto para la empresa.

La tabla **BW_EMPRESA** almacenará las empresas con las cuales nosotros podemos ofrecer el servicio de pagos en línea con nuestro Portal de servicios básicos.

La tabla **BW_EMPRESA** tiene los siguientes campos:

- **EM_DIRECCION:** Este campo es de tipo VARCHAR2, y almacenará la dirección física de la localización de la matriz de la empresa de servicio básico.
- **EM_IDCIUD:** Este campo es de tipo NUMBER, y almacenará el código de ciudad donde se encuentra la empresa.
- **EM_IDEMP:** Este campo es de tipo NUMBER, y almacenará el código o id de la empresa.
- **EM_NOMBRE:** Este campo es de tipo VARCHAR2, y almacenará el nombre de la empresa.
- **EM_RUC:** Este campo es de tipo VARCHAR2, y almacenará el número de RUC de la empresa.
- **EM_SERVBAS:** Este campo es de tipo NUMBER, y almacenará el código del servicio básico que ofrece la empresa.
- **EM_TABLA:** Este campo es de tipo VARCHAR2, y almacenará el nombre de la tabla con el cual estará referenciado en nuestra base de datos esa empresa.
- **EM_TELEFONO:** Este campo es de tipo VARCHAR2, y almacenará el teléfono de la empresa.

La tabla **BW_PAGO** almacenará todos los pagos que sean efectuados por nuestros clientes, registrados a través de nuestro Portal de pagos de Servicios Básicos.

La tabla **BW_PAGO** tiene los siguientes campos:

- **PG_ABRSERV:** Este campo es de tipo VARCHAR2, y almacenará un código interno de nuestro tipo de servicio, ya sea de agua, luz o teléfono.
- **PG_ABRTARJ:** Este campo es de tipo VARCHAR2, y almacenará el código interno, que pertenece a la tarjeta con la cual se efectuó el pago. Las tarjetas pueden ser Diners, MasterCard, Visa etc. y sus abreviaciones puede ser algún número o letras que los identifique.
- **PG_CEDRUC:** Este campo es de tipo VARCHAR2, y almacenará la cédula o RUC del usuario que efectuó el pago.
- **PG_CUENTA:** Este campo es de tipo VARCHAR2, y almacenará el número de cuenta del servicio básico, del pago efectuado por nuestro cliente registrado.
- **PG_FACTURA:** Este campo es de tipo VARCHAR2, y almacenará el número de factura del servicio básico del pago efectuado por nuestro usuario.

- **PG_FECHA:** Este campo es de tipo DATE, y almacenará la fecha en que se realizó el pago.
- **PG_IDCON:** Este campo es de tipo NUMBER, y almacenará el id de la consulta, si es que se realizó un pago proveniente de una consulta de planilla o alguna otra consulta pagada.
- **PG_IDEMP:** Este campo es de tipo NUMBER, y almacenará el código de la empresa a la cual se efectuó el pago.
- **PG_IDPAGO:** Este campo es de tipo NUMBER, y almacenará un código para el pago efectuado.
- **PG_IDSERVBAS:** Este campo es de tipo NUMBER, y almacenará el código del servicio básico.
- **PG_LOGIN:** Este campo es de tipo VARCHAR2, y almacenará el login de nuestro cliente registrado, el cual efectuó previamente un pago.
- **PG_TOTAL:** Este campo es de tipo NUMBER, y almacenará el total de lo que canceló nuestro usuario en ese pago.

La tabla **BW_PROVINCIA** almacenará las provincias de nuestro país: Ecuador.

La tabla **BW_PROVINCIA** tiene los siguientes campos:

- **PV_DESCRI:** Este campo es de tipo VARCHAR2, y almacenará la descripción o nombre de la provincia.
- **PV_IDPROV:** Este campo es de tipo NUMBER, y almacenará el código de la provincia.
- **PV_IDREG:** Este campo es de tipo NUMBER, y almacenará el código de la región al cual pertenece la provincia.

La tabla **BW_REGION** almacenará las diferentes regiones de nuestro país Ecuador.

La tabla **BW_REGION** tiene los siguientes campos:

- **RG_DESCRI:** Este campo es de tipo VARCHAR2, y almacenará el nombre o descripción de la región.
- **RG_IDREG:** Este campo es de tipo NUMBER, y almacenará el código de la región.

La tabla **BW_SERVBAS** almacenará los servicios básicos con los que nuestro Portal puede efectuar los servicios de pagos o consultas.

La tabla **BW_SERVBAS** tiene los siguientes campos:

- **SB_DESCRI:** Este campo es de tipo VARCHAR2, y almacenará el nombre o descripción del servicio básico.
- **SB_IDSERVBAS:** Este campo es de tipo NUMBER, y almacenará el código de nuestro servicio básico.

La tabla **BW_SERVICIO** almacenará los servicios que ofrecemos en nuestro Portal de pagos.

La tabla **BW_SERVICIO** tiene los siguientes campos:

- **SV_ABRSERV:** Este campo es de tipo VARCHAR2, y almacenará un código interno, para reconocer el servicio que ofrecemos en nuestro portal.
- **SV_DESCRI:** Este campo es de tipo VARCHAR2, y almacenará un nombre o descripción del servicio que ofrecemos.
- **SV_IDSERV:** Este campo es de tipo NUMBER, y almacenará el código del servicio ofrecido en nuestro portal.

La tabla **BW_TARJETA** almacenará las tarjetas con las que nuestros usuarios pueden efectuar pagos o consultas de sus cuentas y facturas.

La tabla **BW_TARJETA** tiene los siguientes campos:

- **TJ_ABRTARJ:** Este campo es de tipo VARCHAR2, y almacenará un código interno de la tarjeta.
- **TJ_DESCRI:** Este campo es de tipo VARCHAR2, y almacenará el nombre o descripción de la tarjeta.
- **TJ_IDTARJ:** Este campo es de tipo NUMBER, y almacenará el código de la tarjeta de forma encripada.
- **TJ_TIPO:** Este campo es de tipo VARCHAR2, y almacenará el tipo de la tarjeta ya sea CREDITO o DEBITO o algún otro.

La tabla **BW_TIPCONS** almacenará los diversos tipos de consulta que podemos ofrecer en nuestro Portal de pagos de servicio básicos.

La tabla **BW_TIPCONS** tiene los siguientes campos:

- **TC_DESCRI:** Este campo es de tipo VARCHAR2, y almacenará el nombre o una descripción de una consulta.
- **TC_TIPCONS:** Este campo es de tipo VARCHAR2, y almacenará el id o código de la consulta.

La tabla **BW_USUARIO** almacenará los clientes que se registren en nuestro Portal para poder efectuar los pagos en línea.

La tabla **BW_USUARIO** tiene los siguientes campos:

- **US_CEDRUC:** Este campo es de tipo VARCHAR2, y almacenará la cédula del usuario.
- **US_LOGIN:** Este campo es de tipo VARCHAR2. Almacenará el login para poder ingresar y usar los servicios de consultas y pagos de las planillas de servicios básicos.
- **US_PASSWD:** Este campo es de tipo VARCHAR2, y almacenará la contraseña secreta del usuario para poder usar los servicios del Portal. Este campo es almacenado en la base de forma encriptada.
- **US_PREGUNTA:** Este campo es de tipo VARCHAR2, y almacenará una pregunta que le ayudará a recuperar su contraseña en caso de que no la acuerde.
- **US_RESPUESTA:** Este campo es de tipo VARCHAR2, y almacenará la respuesta a la pregunta que debe coincidir con la respuesta que el usuario ingrese cuando se olvide su contraseña. Este campo es almacenado en la base de forma encriptada.

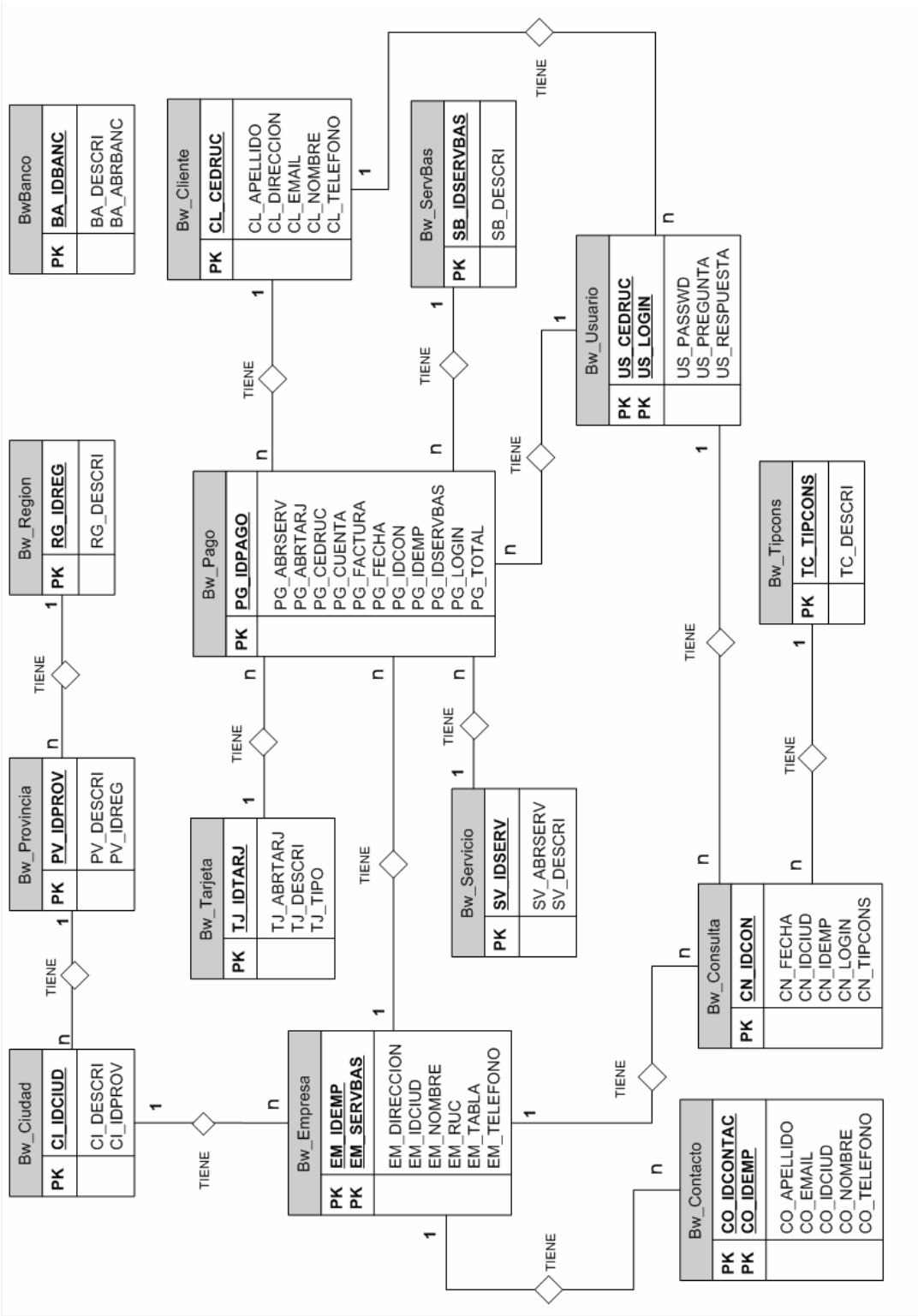


Figura 4.16. Diagrama entidad-relación de la base de datos.

CAPITULO 5

5. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

En este capítulo, explicamos cada uno de los pasos seguidos en la implementación del sistema a nivel de hardware y software. Además mencionamos las dificultades presentadas en este proceso.

Hacemos una descripción de los pasos seguidos en la instalación de las herramientas de software, en cada uno de los servidores utilizados.

Explicamos también, el mecanismo de autenticación usado en nuestro servidor web, por medio de certificados digitales; además describimos los pasos que deben seguirse para adquirir un certificado digital y cómo instalarlo en un servidor Web Apache, ejecutado en un servidor Linux.

5.1. Proceso de Implementación

Esta sección la hemos dividido en dos partes:

- Descripción general, que contiene los pasos seguidos para implementar el portal de pagos.
- Descripción de Componentes, que describe brevemente las clases utilizadas.

5.1.1. Descripción General

Empezamos explicando la implementación física del sistema, basado en la arquitectura de red explicada anteriormente, en el capítulo cuatro sección 1.

Utilizamos 2 switches de 12 puertos; uno para la red que contiene los servidores expuestos a Internet como son: servidor Web, servidor Proxy, servidor DNS, servidor de Correo y otro para la red de servidores que actualmente está formada solo por el servidor de base de datos. También tenemos un switch de 24 puertos, utilizado para la red interna, en donde tenemos las máquinas para el personal que conforma la empresa.

Hemos decidido utilizar switches en lugar de hubs debido a las características que éstos tienen, principalmente:

- Ancho de banda dedicado, todos los puertos tienen el mismo ancho de banda.
- Separa dominios de colisión.

Tuvimos la necesidad de comunicar las diferentes redes, para poder hacerlo, necesitamos de un enrutador que se encargue de pasar la información de una red a otra, esta característica la conseguimos por medio del firewall, el mismo que será utilizado como enrutador, además de proveer seguridad a la red.

Otra configuración que se debió hacer fue asignar las direcciones IP y un gateway a cada equipo, los equipos conectados a un mismo switch deben pertenecer a la misma red. Para conectar un host al switch utilizamos un cable directo UTP categoría 6.

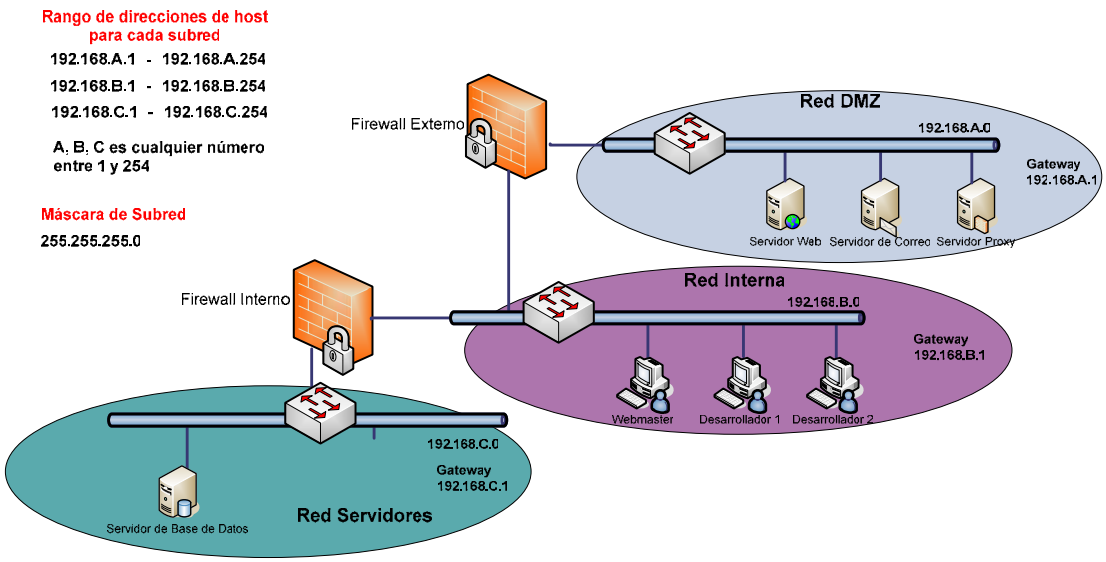


Figura 5.1. Asignación de direcciones IP y gateway a los equipos

Con el firewall externo controlamos el acceso indebido de usuarios de Internet a cualquiera de las redes. Con el firewall interno controlamos el acceso a Internet por parte de los usuarios de la red interna; también este firewall nos permite controlar el acceso al servidor de base de datos.

Los equipos que contienen los firewalls tienen instalado RedHat Linux 9 con los recursos mínimos de instalación, esta versión del sistema operativo tiene incluida la herramienta Iptables, que permite crear reglas de filtrado de paquete para aceptar o denegar el acceso a los recursos. Con esta herramienta podemos

crear un firewall adaptado a nuestras necesidades. También existen equipos de hardware que actúan como firewalls, hemos decidido utilizar Iptables debido a su libre distribución.

El servidor Web que utilizamos es el Apache versión 2, corriendo sobre el sistema operativo Red Hat Linux 9, en el equipo que se encuentra instalado el servidor Web, también se encuentra instalado el contenedor de páginas jsp Tomcat versión 4.1.30 y la máquina virtual de Java j2sdk versión 1.4.2, que es requerida por Tomcat para su funcionamiento.

En el servidor de bases de datos tenemos instalado Oracle 9i, también ejecutándose sobre el sistema operativo Linux. En este servidor, instalamos un sistema de detección de intrusos, que registre el acceso indebido a sus recursos, el IDS utilizado es Snort versión 2.3.

El servidor de nombres de dominio que utilizamos es bind versión 9.2.3, este DNS es para Linux.

Tenemos un servidor de correo, que permite el envío y recepción de correos a los usuarios internos. Como cliente de correo se

puede utilizar cualquiera que trabaje con los protocolos pop e imap. El servidor de correo instalado es sendmail versión 8.12.11, y utilizamos la librería cyrus-sasl que viene incluida en Linux Red Hat 9 para la autenticación de los usuarios.

El software utilizado para la creación de la red virtual privada es freeswan-module y freeswan-userlan. Estos paquetes están instalados en los firewalls, según la configuración permiten encriptar el tráfico desde y hacia el servidor de base de datos.

El Servidor Proxy utilizado para controlar el acceso a Internet por parte de los usuarios de la red interna es Squid versión 2.5.

Todos los servidores tienen instalado el sistema operativo Linux Red Hat 9. Las estaciones de trabajo para los desarrolladores tienen el sistema operativo Windows 2000.

Los requerimientos de hardware, de todos los equipos que forman la red, fueron descritos en el capítulo 3 sección 3.

5.1.2. Descripción de Componentes

Nombre: ConsultaAgua.java

Paquete: com.pagosaldia.servlets

Descripción: Consulta en la base de las empresas de servicios básicos de agua, todas las planillas que pertenecen al usuario conectado actualmente. El resultado de la consulta es reenviada a la página Consulta.jsp, que se encarga de mostrarlos de forma ordenada.

Nombre: ConsultaLuz.java

Paquete: com.pagosaldia.servlets

Descripción: Consulta en la base de las empresas de servicios básicos de luz, todas las planillas que pertenecen al usuario conectado actualmente. El resultado de la consulta es reenviada a la página Consulta.jsp, que se encarga de mostrarlos de forma ordenada.

Nombre: ConsultaTelefono.java

Paquete: com.pagosaldia.servlets

Descripción: Consulta en la base de las empresas de servicios básicos de teléfono, todas las planillas que pertenecen al usuario conectado actualmente. El resultado de la consulta es reenviada a la página Consulta.jsp, que se encarga de mostrarlos de forma ordenada.

Nombre: ConsultaGeneral.java

Paquete: com.pagosaldia.servlets

Descripción: Consulta en la base de todas las empresas de servicios básicos (agua, luz y teléfono), todas las planillas que pertenecen al usuario conectado actualmente. El resultado de la consulta es reenviada a la página Consulta.jsp, que se encarga de mostrarlos de forma ordenada.

Nombre: Ingreso.java

Paquete: com.pagosaldia.servlets

Descripción: Verifica que los datos ingresados por el usuario para autenticarse correspondan a los datos en la base y así poder acceder al sistema. Si no existe el usuario o la contraseña es incorrecta, se presenta un mensaje indicando el error y se mantiene en la página principal. Si los datos son correctos hace una consulta general y muestra estos datos de forma ordenada en la página de servicios.htm

Nombre: Pago.java

Paquete: com.pagosaldia.servlets

Descripción: Valida que los datos de la tarjeta (crédito o débito) sean correctos, además verifica que el monto del crédito sea mayor o igual al monto a pagar y que el estado sea activo.

También se encarga de registrar el pago en la base de datos y enviar un mail confirmando el pago realizado; para el envío de mail utiliza la clase EnviarMail.java.

Nombre: PagoAgua.java

Paquete: com.pagosaldia.servlets

Descripción: Consulta, en la base de las empresas de servicios básicos de agua, todas las planillas que deben ser canceladas. Los datos más importantes de la planilla para realizar el pago son reenviados a la página PagoAgua.jsp, que se encarga de mostrarlos de forma ordenada.

Nombre: PagoLuz.java

Paquete: com.pagosaldia.servlets

Descripción: Consulta en la base de las empresas de servicios básicos de luz, todas las planillas que deben ser canceladas. Los datos más importantes de la planilla para realizar el pago son reenviados a la página PagoLuz.jsp, que se encarga de mostrarlos de forma ordenada

Nombre: PagoTelefono.java

Paquete: com.pagosaldia.servlets

Descripción: Consulta en la base de las empresas de servicios básicos de teléfono, todas las planillas que deben ser canceladas. Los datos más importantes de la planilla para realizar el pago son reenviados a la página PagoTelefono.jsp, que se encarga de mostrarlos de forma ordenada

Nombre: Registrar.java

Paquete: com.pagosaldia.servlets

Descripción: Verifica que los datos ingresados por el usuario, para registrarse, no estén en la base de datos y que tenga una cuenta en alguna de las empresas que ofrecen servicios básicos. Luego de hacer esta validación almacena los datos del nuevo usuario en la base, y envía un mensaje de confirmación de registro al usuario, este mensaje es mostrado en la página registrado.htm

Figura 5.2. Paquete com.pagosaldia.servlets y sus clases

Nombre: EnviarMail.java

Paquete: com.pagosaldia.tools

Descripción: Se encarga de conectarse al servidor de correo, también implementa el método que permite enviar un correo. Esta clase es utilizada para enviar un correo después de realizar un pago.

Nombre: GFunctions.java

Paquete: com.pagosaldia.tools

Descripción: Implementa los métodos para encriptar y desencriptar algún texto, también el método para el reenvío de información de un servlet a una página.

Nombre: Tarjeta.java

Paquete: com.pagosaldia.tools

Descripción: Implementa los métodos que me permiten validar los datos de autenticación de la tarjeta (usuario y clave), valida el estado de la tarjeta y el cupo disponible para el crédito.

Figura 5.3. Paquete com.pagosaldia.tools y sus clases

Nombre: InfoUsuario

Paquete: com.pagosaldia.beans

Descripción: Bean usado para el almacenamiento en memoria, de los datos principales de las facturas pendientes de nuestro usuario.

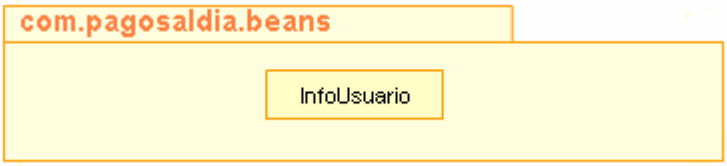


Figura 5.4. Paquete com.pagosaldia.beans y sus clases

Nombre: Connector

Paquete: com.pagosaldia.datasource

Descripción: Contiene todos los métodos generales para ejecutar consultas, inserciones, actualización de registros y Stored Procedures, utilizando los beans de las tablas de la Base de datos,

Nombre: Consultas

Paquete: com.pagosaldia.datasource

Descripción: Contiene métodos para la ejecución de consultas, inserciones y actualización de registros pero con filtros específicos o condiciones usando lenguaje SQL, sin el uso de los beans de las tablas. Esta clase utiliza la clase Connector.java

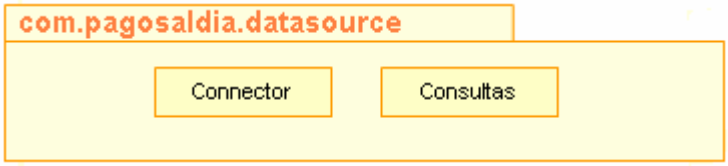


Figura 5.5. Paquete com.pagosaldia.datasource y sus clases

Nombre: Wrapper.java

Paquete: com.pagosaldia.displaytag.wrapper

Descripción: Esta clase nos ayuda con el uso de las tablas dinámicas generadas por la librería displaytag. La clase Wrapper proporciona a nuestras tablas un mejor comportamiento y estética.

Figura 5.6. Paquete com.pagosaldia.displaytag.wrapper y sus clases

5.1.3. Pruebas

Hemos realizado varias pruebas, las mismas que las hemos clasificado de la siguiente manera:

- Pruebas de Diseño
- Pruebas de Funcionalidad
- Pruebas de Rendimiento

Las pruebas de diseño consistieron en verificar que todos los enlaces del sitio estén funcionando correctamente, también revisamos que el texto contenido en las páginas esté correctamente redactado y sin faltas ortográficas.

Con las pruebas de funcionalidad verificamos que:

- Los campos de los formularios sean validados en el cliente, antes de ser enviados al servidor Web.
- La validación de usuario y contraseña esté funcionando correctamente.
- El proceso de consulta se realice de forma correcta, esto es, verificar si usuario tiene planillas pendientes de pagar, presentar la información de forma ordenada y agrupada de acuerdo al servicio básico, presentar mensajes de alerta y/o de error.

- El proceso de verificación de la tarjeta esté funcionando correctamente. Este proceso consiste en validar el usuario, clave, cupo y estado de tarjeta, si algún dato no es correcto presentar un mensaje que notifique al usuario lo que está pasando.
- Luego de realizar el pago se envíe un mail al cliente, informándole que el pago se realizó satisfactoriamente.
- Una vez que una planilla ha sido cancelada, su estado haya sido modificado, para que al hacer una nueva consulta esta planilla no sea mostrada.

Las pruebas de rendimiento consistieron en medir el tiempo que tarda el sistema en responder a cualquier transacción realizada por un cliente. Estas pruebas las realizamos con la herramienta OpenSTA, que es de libre distribución. La transacción usada es la de acceso al sistema, OpenSTA es el encargado de generar la carga (número de usuarios conectados). La plataforma utilizada para realizar estas pruebas tiene las siguientes características:

- En el Servidor Web se encuentra instalado: Sistema operativo Windows XP, Apache y Tomcat.

- Servidor de Base de Datos tiene instaladas las siguientes herramientas: Sistema operativo Linux Red Hat 9 y Oracle 9i.
- Cualquier petición a la base de datos, debe atravesar un Firewall Externo y un Firewall Interno, los mismos que tienen instalado lo siguiente: Sistema operativo Linux Red Hat 9 e Iptables

En los siguientes gráficos, podemos observar el resultado de las pruebas realizadas con OpenSTA. Estas pruebas han sido realizadas con las siguientes opciones de carga:

- 100 usuarios
- 250 usuarios
- 500 usuarios
- 1000 usuarios.

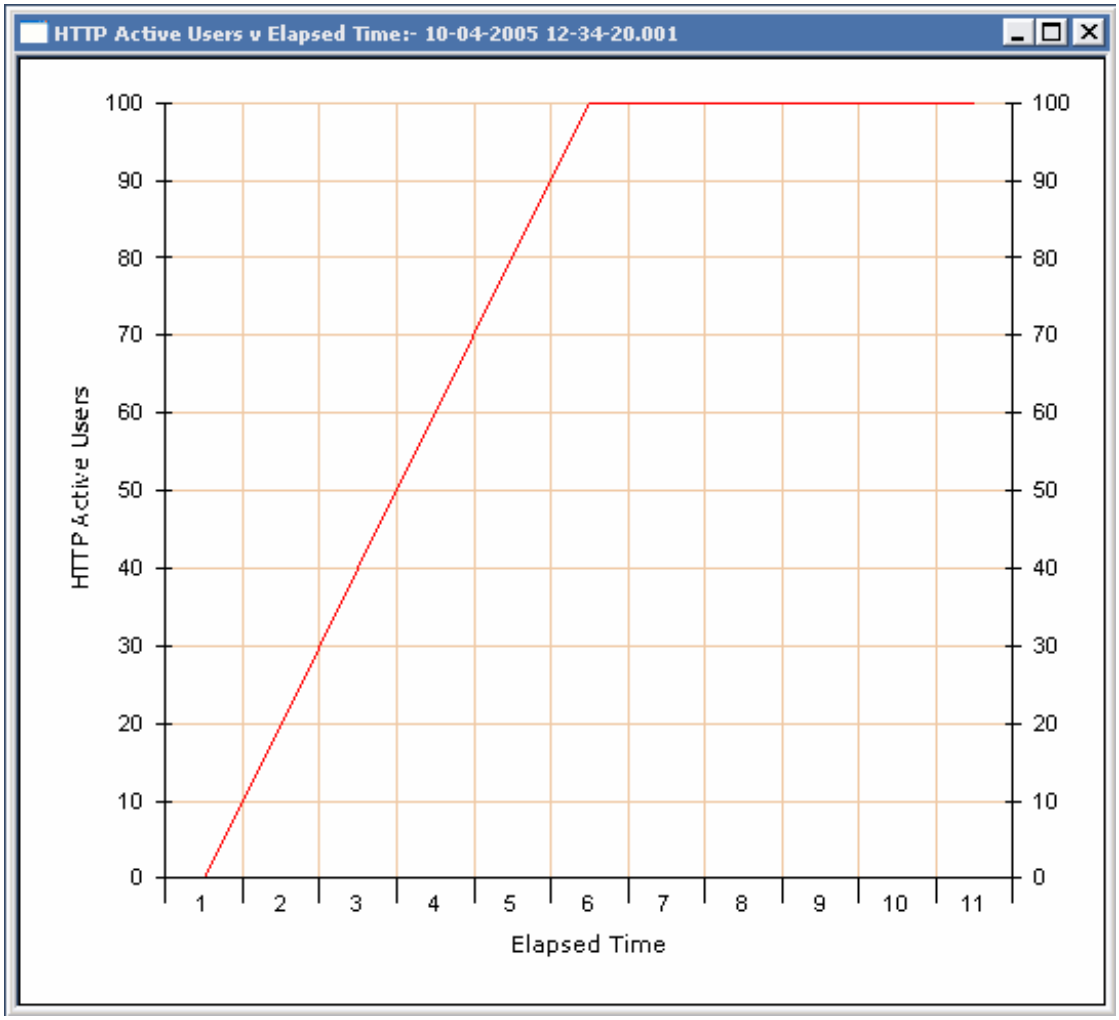


Figura 5.7 Prueba de rendimiento con 100 usuarios conectados

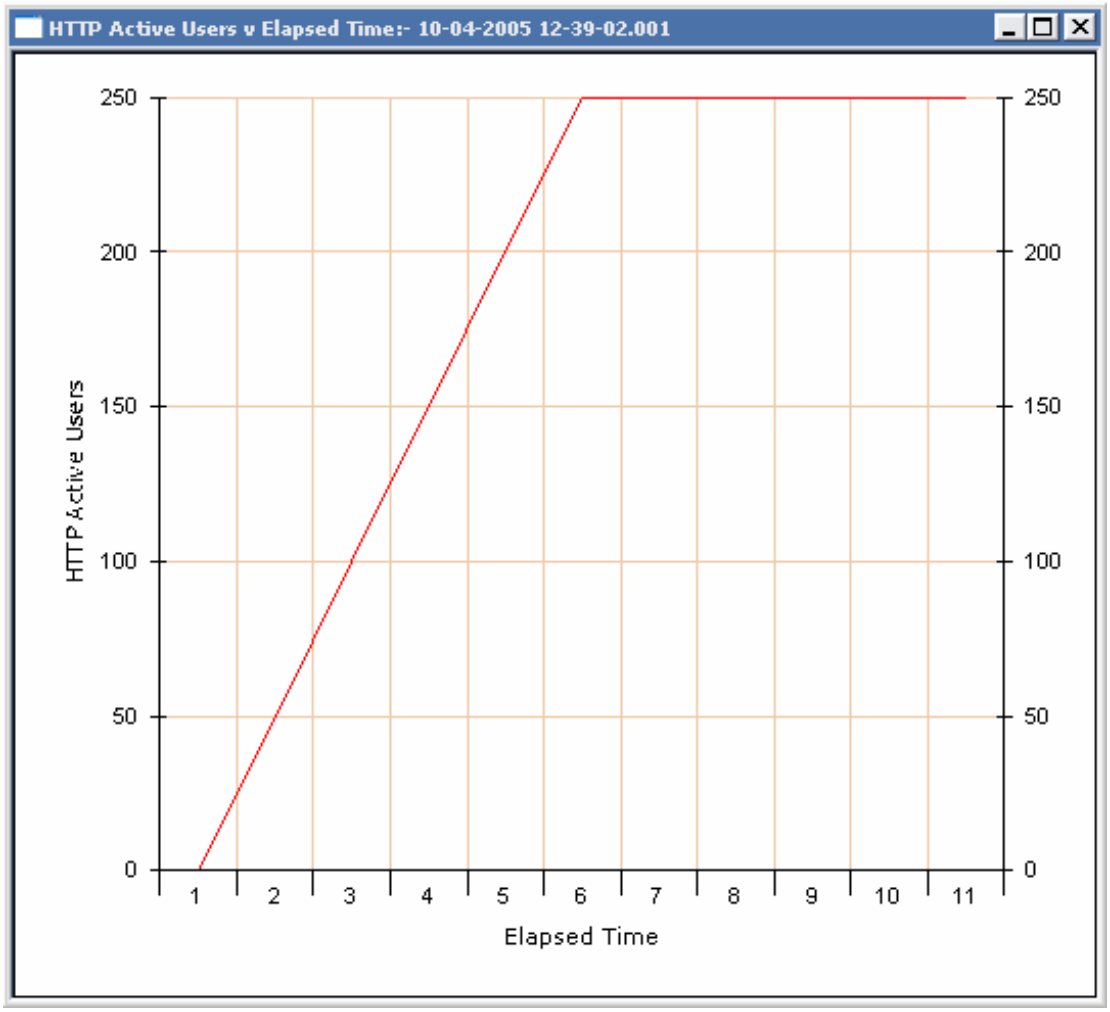


Figura 5.8 Prueba de rendimiento con 250 usuarios conectados

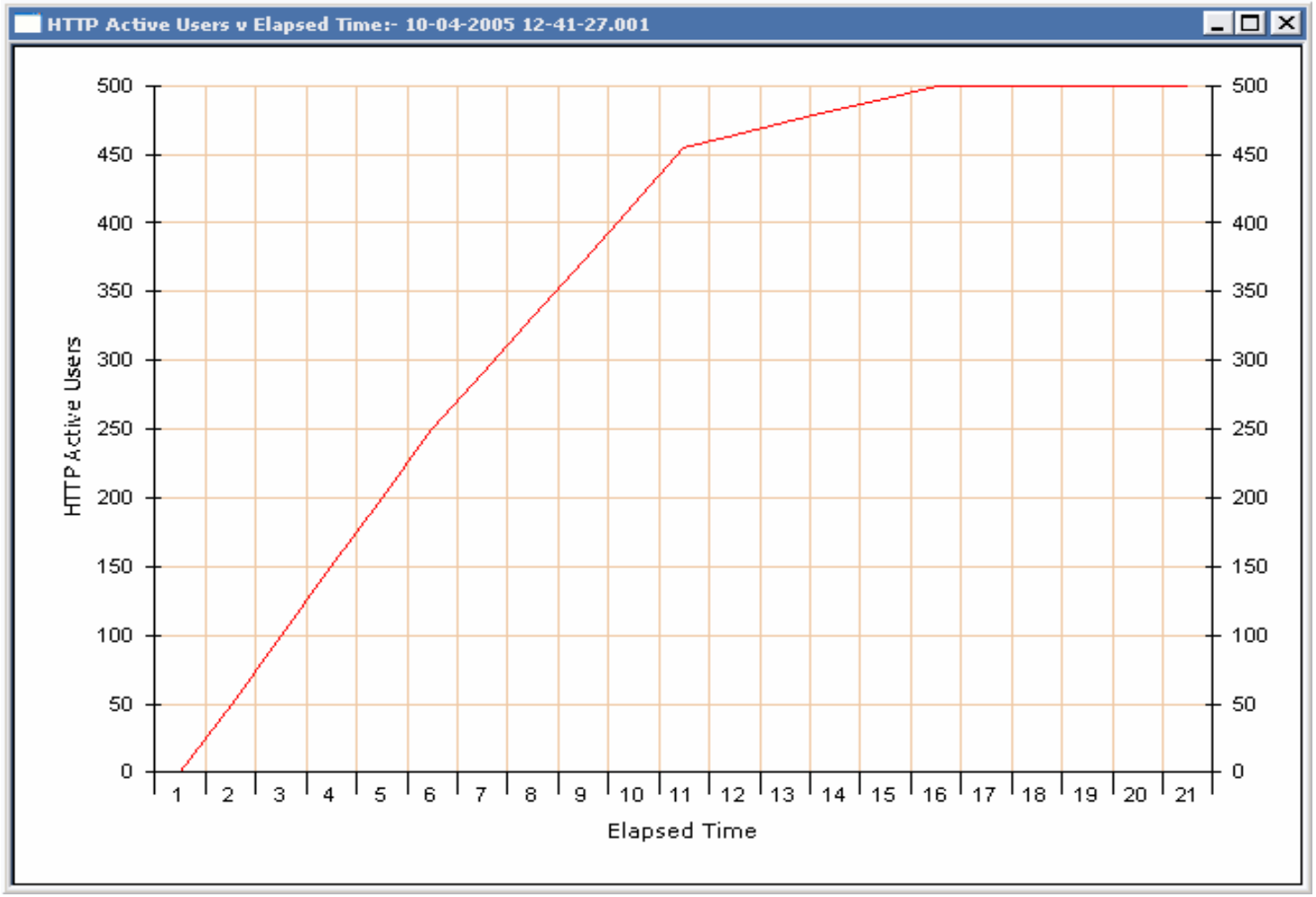


Figura 5.9 Prueba de rendimiento con 500 usuarios conectados

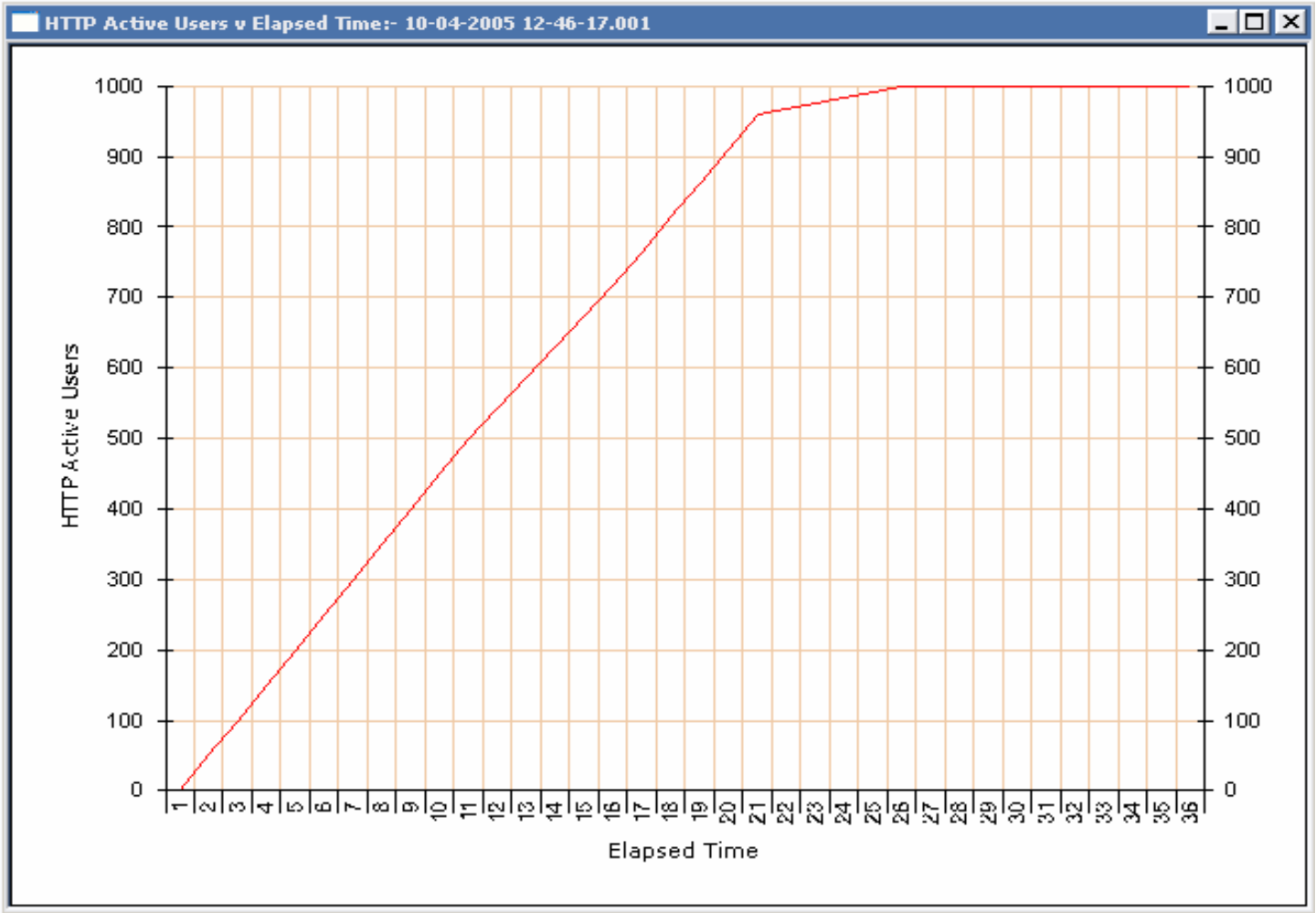


Figura 5.10 Prueba de rendimiento con 1000 usuarios conectados

5.2. Instalación de herramientas de Software

En el Apéndice D, se encuentran los manuales de instalación de todas las herramientas de software, de los diferentes servidores utilizados para la implementación del portal.

5.3. Autenticación

En esta sección, explicamos cómo obtener un certificado digital de una de las autoridades de certificación y cómo instalarlo. En nuestro caso, lo instalamos en el servidor Web. También definimos algunos conceptos importantes sobre certificados digitales.

5.3.1. Conceptos Generales

Como ya vimos en el capítulo 3 sección 4.1, un certificado digital es un documento electrónico que contiene los datos de identificación del propietario, su clave pública y la firma digital de una autoridad certificadora. A continuación definimos qué es una firma digital y una autoridad de certificación.

Firma Digital

La firma digital es utilizada para firmar documentos digitales, con esta firma se garantiza la autenticidad del mensaje y el no repudio. Ahora explicamos cómo se genera la firma digital de un documento:

Emisor:

Aplica una función hash[1] de resumen al documento que va a enviar.

Encripta el mensaje con su clave privada

Envía el mensaje cifrado

Receptor:

Genera su propio resumen, usando la clave pública del emisor.

Compara su resumen con el recibido. Si los resúmenes coinciden, el mensaje es auténtico.

Las firmas digitales tienen los siguientes propósitos:

[1] Función Hash: Esta función transforma un mensaje de cualquier longitud en un mensaje de longitud fija

- **Validar el contenido** de un mensaje electrónico, y se puede utilizar posteriormente, para comprobar que un emisor envió, de hecho, ese mensaje.
- **Probar que no se ha falsificado un mensaje durante su envío**. Las firmas digitales respaldan la autenticidad del correo electrónico, transacciones de contabilidad, órdenes de empresa, documentos para grupos de trabajo y otros mensajes y archivos que se trasladan entre sistemas, usuarios u organizaciones.

La validez de la firma digital depende de la seguridad de la clave privada.

Autoridad de Certificación (CA)

Esta entidad se encarga de firmar digitalmente con su clave privada los certificados digitales, es decir, es quien asegura que los datos que contiene el certificado digital son válidos. Por esta razón, el esquema de seguridad que emplee la CA para proteger la clave privada, debe ser robusto.

Las autoridades de certificación realizan las siguientes tareas:

- Emisión de los certificados de usuarios registrados y validados por la Autoridad de Registro (RA)[2].
- Revocación de los certificados que ya no sean válidos (CRL - lista de certificados revocados). Un certificado puede ser revocado porque los datos han dejado de ser válidos, la clave privada ha sido comprometida o el certificado ha dejado de tener validez, dentro del contexto para el que había sido emitido.
- Renovación de certificados.
- Publicar certificados en el directorio repositorio de certificados.

En las siguientes secciones explicamos cómo se hace una solicitud de un certificado digital a una autoridad certificadora y la instalación del mismo.

5.3.2. Requerimiento de un certificado digital

Vamos a generar la solicitud de un certificado digital, para poder hacerlo, hemos instalado previamente el servidor web apache con un módulo para que soporte SSL.

[2] Autoridad de Registro: Entidad delegada por la autoridad de certificación para emitir certificados digitales.

Primero creamos los directorios para el certificado (crt), para la clave privada (key) y para el requerimiento de un certificado (csr) para alguna entidad certificadora (Verisign).

```
# mkdir /usr/local/apache/conf/ssl.crt
# mkdir /usr/local/apache/conf/ssl.key
# mkdir /usr/local/apache/conf/ssl.csr
```

O todo en una sola línea:

```
# mkdir /usr/local/apache/conf/ssl.crt
/usr/local/apache/conf/ssl.key   /usr/local/apache/conf/ssl.csr
```

Generamos nuestra clave:

```
#/usr/bin/openssl genrsa -des3 1024 >
/usr/local/apache/conf/ssl.key/server.key
```

Cuando se crea una clave pública, se debe ingresar una frase clave, que debe ser tecleada y recordada cuando se ejecute el apache con módulo SSL

Luego cambiamos los permisos del archivo server.key, que contiene la clave generada:

```
# chmod go-rwx /usr/local/apache/conf/ssl.key/server.key
```

Ahora generamos nuestro requerimiento de certificado (csr)

```
# /usr/bin/openssl req -new -key
/usr/local/apache/conf/ssl.key/server.key -out
/usr/local/apache/conf/ssl.csr/server.csr
```

Una vez ingresado el comando anterior, mostrará por pantalla el siguiente mensaje:

```
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
```

Debe ingresar la frase clave que eligió cuando generó su clave. El sistema mostrará algunas instrucciones que requerirán una serie de respuestas. Dichas respuestas serán incorporadas a la petición del certificado.

Al final pedirá un password y una compañía, adicional a esos dos pasos, solo presionar la tecla enter. La pantalla, con respuestas de ejemplo, será similar a ésta:

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]:US

State or Province Name (full name) [Berkshire]:North Carolina

Locality Name (eg, city) [Newbury]:Raleigh

Organization Name (eg, company) [My Company Ltd]:Test Company

Organizational Unit Name (eg, section) []:Testing

Common Name (your name or server's hostname) []:test.example.com

Email Address []:admin@example.com

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

5.3.3. Creación de un certificado digital

En esta sección, explicamos cómo generar un certificado digital firmado por nosotros mismos. Este certificado lo creamos para generar nuestras pruebas. Si queremos poner un certificado válido en un servidor, debemos requerirlo a una autoridad certificadora.

Para crear el certificado autofirmado no se necesita el archivo server.csr. El archivo server.csr, solo se lo usa con la entidad que entrega certificados, por ejemplo: Verisign (www.verisign.com). Ingrese los siguientes comandos, todo en una sola línea:

```
# /usr/bin/openssl req -new -key /etc/httpd/conf/ssl.key/server.key -x509
-days 365 -out /etc/httpd/conf/ssl.crt/server.crt
```

Luego de ingresar el comando anterior, aparecerá el siguiente mensaje:

```
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
```

Después de introducir la contraseña o frase clave, se pedirá más información. La salida del ordenador y el conjunto de peticiones

será parecido al siguiente (se necesita dar la información correcta de la organización y de la máquina):

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]:US

State or Province Name (full name) [Berkshire]:North Carolina

Locality Name (eg, city) [Newbury]:Raleigh

Organization Name (eg, company) [My Company Ltd]:My Company, Inc.

Organizational Unit Name (eg, section) []:Documentation

Common Name (your name or server's hostname) []:myhost.example.com

Email Address []:myemail@example.com

Después de proporcionar la información correcta, un certificado autofirmado será creado y colocado en /usr/local/apache/conf/ssl.crt/server.crt.

Ahora, editamos el archivo **/usr/local/apache/conf/ssl.conf** y modificamos las siguientes líneas:

Línea 29, si es que se desea que escuche Apache Seguro en otro puerto:

Listen 443

Línea 89

ServerName www.algo.com:443

En el archivo **httpd.conf,** se debe configurar para que el servidor Web utilice este certificado, esto es explicado en la sección de instalación del servidor Web.

Después de haber generado el certificado es necesario reiniciar el servidor seguro, para hacerlo ingresamos los siguientes comandos:

```
#/usr/local/apache/bin/apachectl stop
#/usr/local/apache/bin/apachectl startssl
```

Luego probamos en un browser (mozilla, Internet Explorer, etc.) de la siguiente manera:

https://localhost:443

CONCLUSIONES

Una vez terminado el desarrollo y la implementación del sistema podemos concluir lo siguiente:

1. Todo el sistema ha sido implementado con software open source, el mismo que brinda iguales características que el software propietario y que pueden ser configuradas de acuerdo a los requerimientos de cada sistema. Además, por ser código abierto, puede ser revisado o modificado, esto puede ser una ventaja o desventaja. Podemos verlo como desventaja ya que como cualquier persona puede tener acceso al código pueden utilizarlo para realizar algún tipo de ataque, la ventaja es que puede ser revisado antes de usarlo para detectar algún tipo de irregularidad.

2. El uso de herramientas de libre distribución no fue una limitación al momento de desarrollar la aplicación, por lo que podemos concluir que

se pueden desarrollar aplicaciones robustas sin tener que pagar licencias por este tipo de herramientas.

3. Se creó una VPN entre el servidor web y el servidor de base de datos para proteger los datos del acceso de cualquier intruso, incluso de los usuarios de la red interna. La desventaja que se presenta al implementar la VPN es que el acceso a los datos será más lento debido a que deben ser encriptados al momento de enviarlos y desencriptarlos al llegar al servidor.

4. Hemos utilizado el protocolo SSL para proteger los datos que van desde el servidor web al navegador del cliente y viceversa, este protocolo es el más utilizado en el comercio electrónico pese a que el protocolo SET ofrece mayor seguridad durante todo el proceso de pago, esto es, proteger los datos de todos los participantes en un proceso de pago usando tarjeta de crédito.

5. El uso de una arquitectura de n capas permite distribuir la carga entre los diferentes servidores utilizados.

6. El uso de Java como lenguaje de programación permite la portabilidad de la aplicación en caso de que se necesite migrar a otra plataforma.

7. Los pagos por Internet en el Ecuador no son muy aceptados debido a que la mayoría de personas no están dispuestas a ingresar la clave de sus tarjetas, algunos no confían porque pueden hacer mal uso de su tarjeta o porque desconocen los sistemas de seguridad aplicados en una transacción de pago.

8. Es necesario que se realicen modificaciones en el código penal para incluir las nuevas formas de delito informático, de esta forma se puede conseguir que tanto consumidores como personas que venden productos o servicios por Internet se sientan respaldados por las leyes.

9. El protocolo SET es más seguro para transacciones de pago a través de Internet realizadas con tarjeta de crédito, pero debido a la complejidad para implementarlo es poco utilizado.

10. SSL es el protocolo más utilizado en la mayoría de los sistemas de pago por Internet debido a que es mucho más fácil de implementarlo. Algunos navegadores como Internet Explorer y Nestcape soportan este protocolo. Este es el protocolo que hemos utilizado en la implementación de nuestro portal.

11. La VPN ocasiona que la velocidad de la transmisión de los datos disminuya debido a que debe encriptar en el emisor y desencriptar en el receptor la información, por esta razón se debe hacer un análisis antes de implementarla, para esto se debe evaluar si es más importante la velocidad de transmisión o la privacidad de los datos.

RECOMENDACIONES

Basados en la experiencia obtenida en el desarrollo del proyecto podemos dar las siguientes recomendaciones:

1. Realizar el diseño físico de red, esto permitirá conocer mejor la ubicación de los equipos y como deben interactuar entre ellos.

2. Los cables de red deben cumplir con los estándares de cableado estructurado y además deben ser correctamente probados para evitar problemas en el futuro como: comunicación entre los equipos, interferencia electromagnética, atenuación, etc.

3. Los cables deben ser de muy buena calidad para evitar rupturas en los hilos de cobre ya que estos no son fáciles de diagnosticar.

4. Definir el número de redes que utilizaremos y asignar de manera eficiente el rango de direcciones, considerando el futuro crecimiento de cada una.

5. Configurar correctamente todos los parámetros de red en cada equipo para evitar problemas de comunicación entre ellos. Un error que se da frecuentemente es la incorrecta asignación del gateway.

6. Los equipos que van conectados a más de una red deben ser configurados cuidadosamente asignando los parámetros de red correctos a cada interfaz para evitar problemas de comunicación.

7. Una vez implementado el diseño de red, probar la comunicación entre equipos de una misma red.

8. Permitir que los equipos de diferentes redes se puedan comunicar, para esto se puede utilizar un firewall que puede ser configurado para que realice las funciones de un ruteador.

9. Verificar que la información que se va a publicar esté correctamente redactada y no tenga faltas ortográficas.

10. Revisar que todos los enlaces estén habilitados.

11. Validar los datos ingresados en los formularios en el lado del cliente para reducir la carga en los servidores.

12. El diseño de las páginas debe ser lo más parecido a la realidad para que el usuario pueda navegar por el sitio intuitivamente.

13. Verificar que la opción de preguntas y respuestas funcionen correctamente, esto es que las preguntas lleguen a la persona encargada de revisarlas.

14. Verificar que el cliente reciba la confirmación de pago luego de haberlo revisado.

15. Obtener las últimas versiones de los paquetes de software que se van a utilizar, verificar que no estén en fase de prueba y mantenerlo siempre con las últimas actualizaciones.

16. Restringir el acceso al servidor de la base de datos a usuarios de la red interna no autorizados.

17. Es recomendable tener redundancia en servidores críticos como por ejemplo la base de datos y el servidor Web. También se deben respaldar con frecuencia los datos de estos servidores.

18. Los datos que viajan desde el servidor Web a la base de datos deben ser protegidos para que los usuarios de la red interna no tengan acceso a ellos. Esto se lo puede conseguir usando una VPN.

19. Instalar un certificado digital en el servidor Web, con esto se consigue que los datos entre el servidor Web y el navegador del cliente sean autenticados y cifrados, esto es importante debido a la información que fluye entre ellos.

20. Controlar el acceso a los recursos de la red por parte de los usuarios internos y externos con la ayuda de un firewall.

21. Configurar todos los paquetes de acuerdo a nuestras necesidades, la mayoría de paquetes tienen opciones por defecto que pueden afectar la seguridad de nuestro sistema.

22. El uso de herramientas de libre distribución es una buena opción para el desarrollo, existe mucha información sobre estas herramientas en

Internet para poder instalarlas y configurarlas. Es recomendable el uso de manuales en inglés ya que contienen información más actualizada.

23. Se recomienda que el servidor de correo tenga desactivada la opción de reenvío para evitar que nuestro servidor sea usado para distribución de spam. También se debe tener actualizada la lista de servidores que envían correo masivamente (spam) y configurar en el servidor para que no reciba correo de estos servidores ya que el spam puede ocasionar un ataque de denegación de servicio.

24. Revisar periódicamente los archivos logs generados por el IDS ya que se crean diariamente y pueden llenar el disco duro ocasionando una caída del sistema.

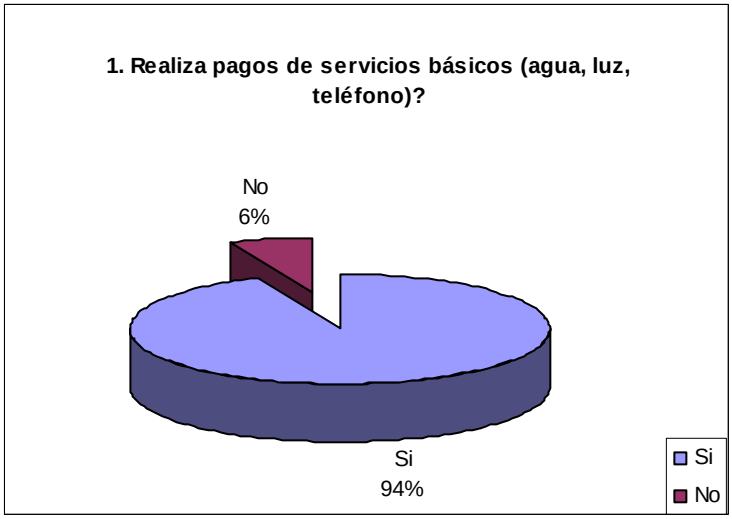
25. El IDS debe ser colocado en los equipos que corren alto riesgo de ser atacados por intrusos y cuya función sea crítica en la red.

APÉNDICE A

RESULTADOS DE ENCUESTAS SOBRE PAGOS DE SERVICIOS BÁSICOS

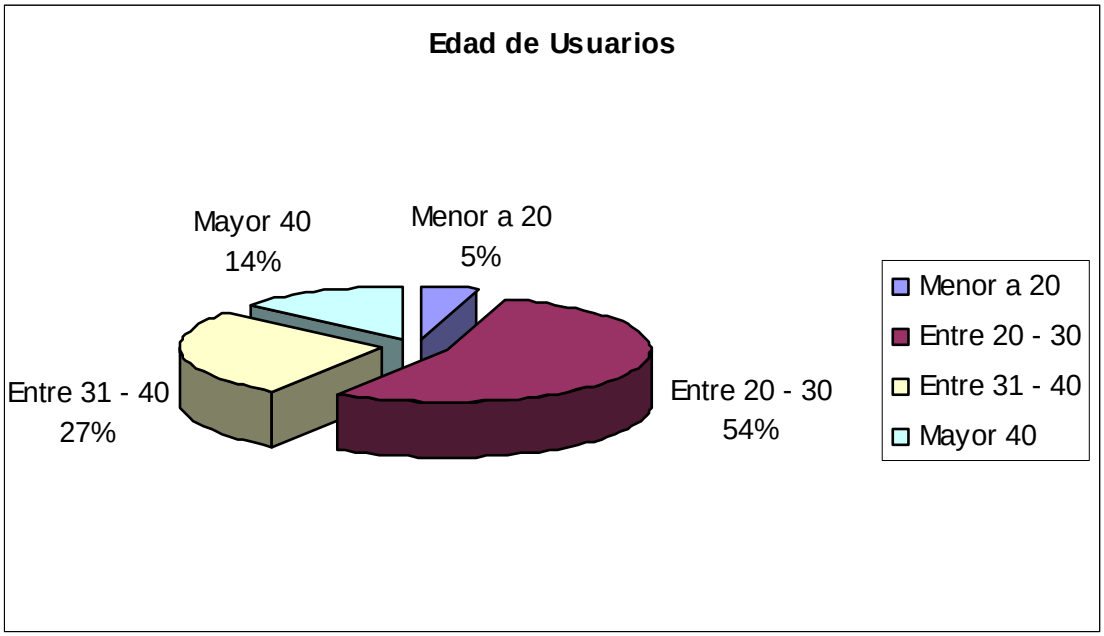
Total de personas encuestadas: 47

1. Realiza pagos de servicios básicos (agua, luz, teléfono)?

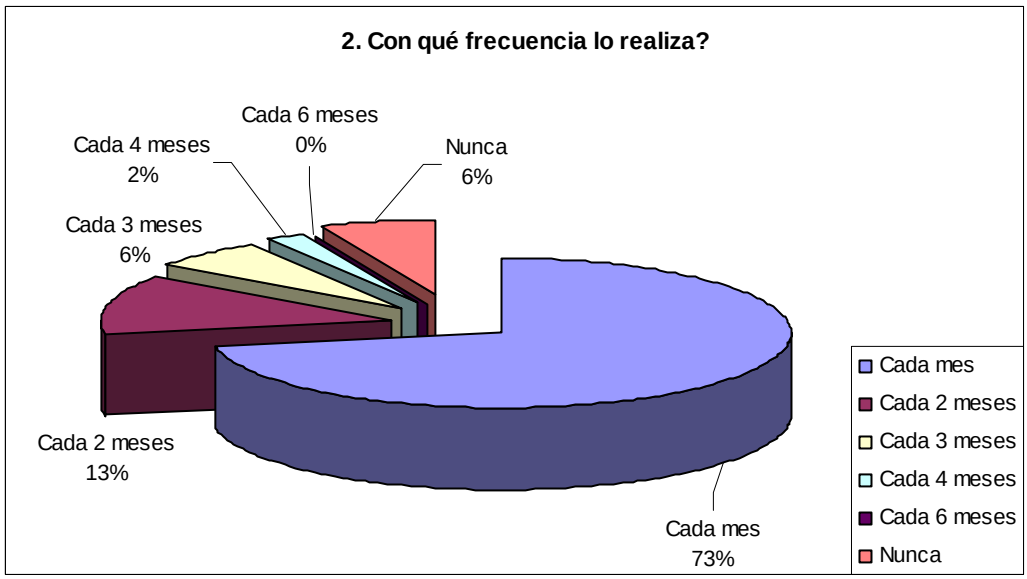


Rango de edades del 94 por ciento de personas que contestaron sí

Menor a 20	Entre 20 - 30	Entre 31 - 40	Mayor 40
2	24	12	6

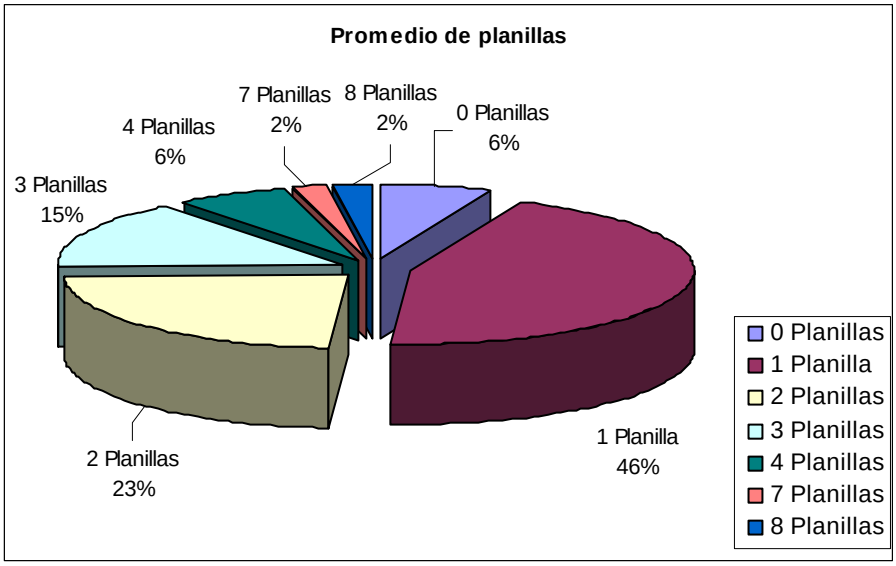


2. ¿Con qué frecuencia lo realiza?

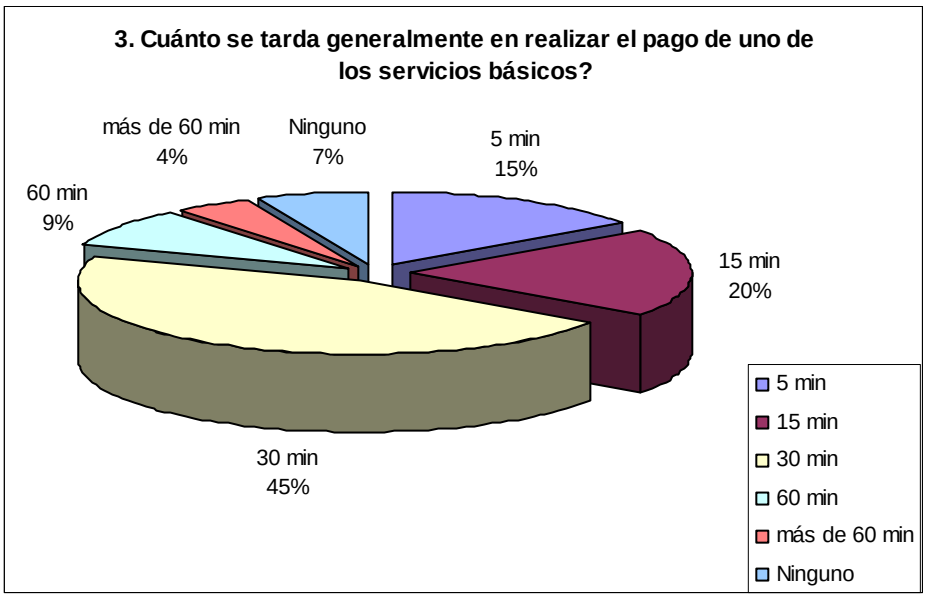


Promedio de planillas que cancela cada persona

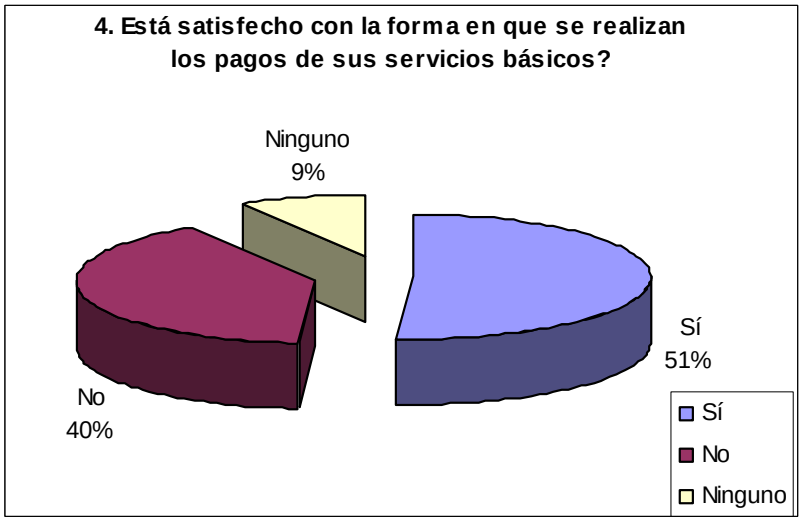
Número de Planillas						
0 Planillas	**1 Planilla**	**2 Planillas**	**3 Planillas**	**4 Planillas**	**7 Planillas**	**8 Planillas**
3	21	11	7	3	1	1



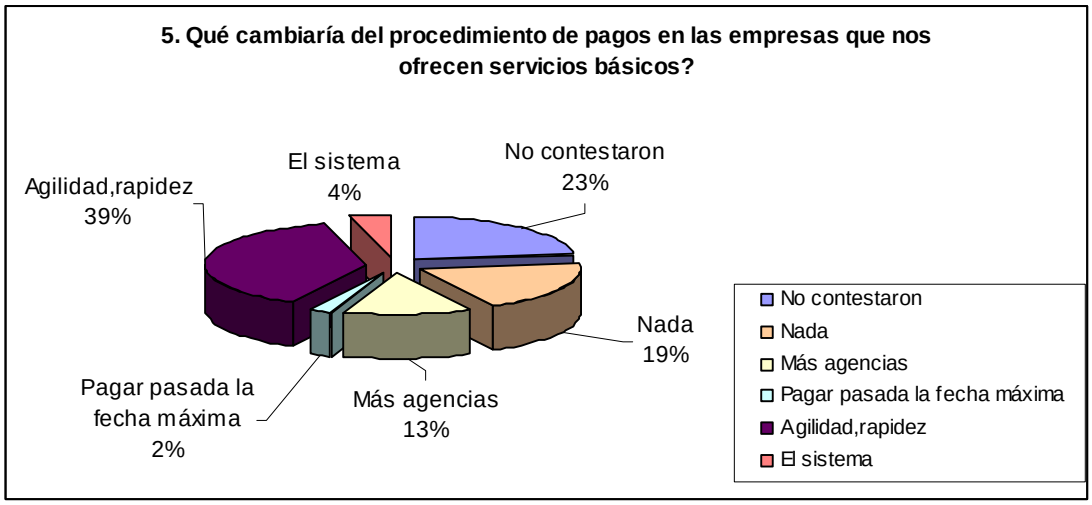
3. ¿Cuánto se tarda generalmente en realizar el pago de uno de los servicios básicos?



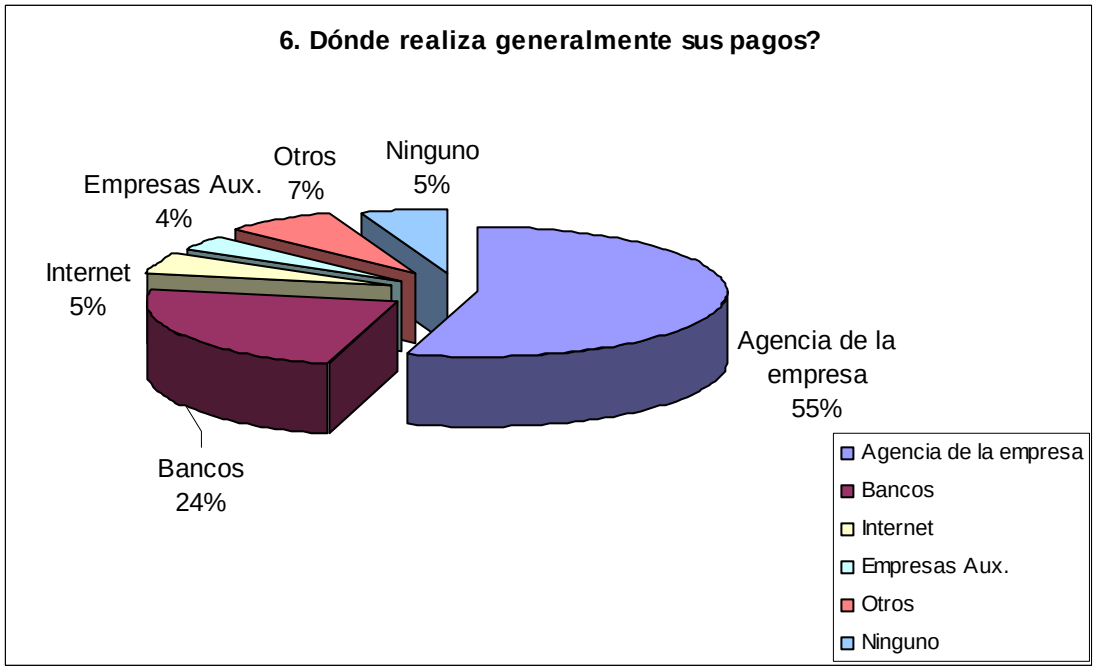
4. Está satisfecho con la forma en que se realizan los pagos de sus servicios básicos?



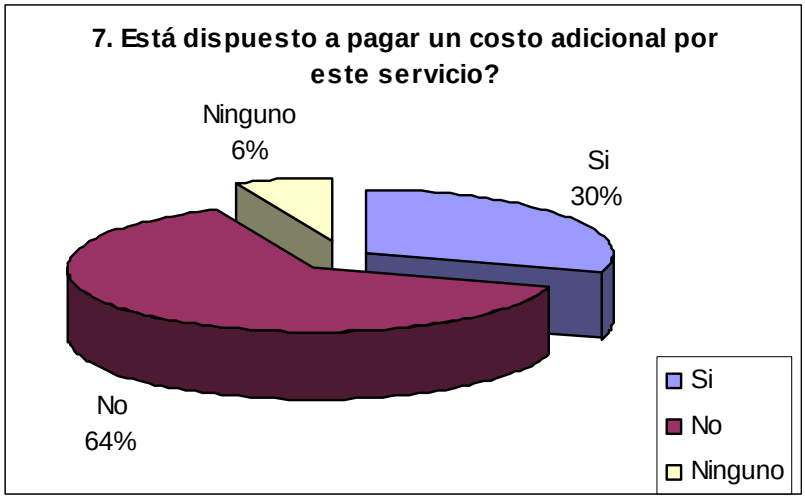
5. ¿Qué cambiaría del procedimiento de pagos en las empresas que nos ofrecen servicios básicos?



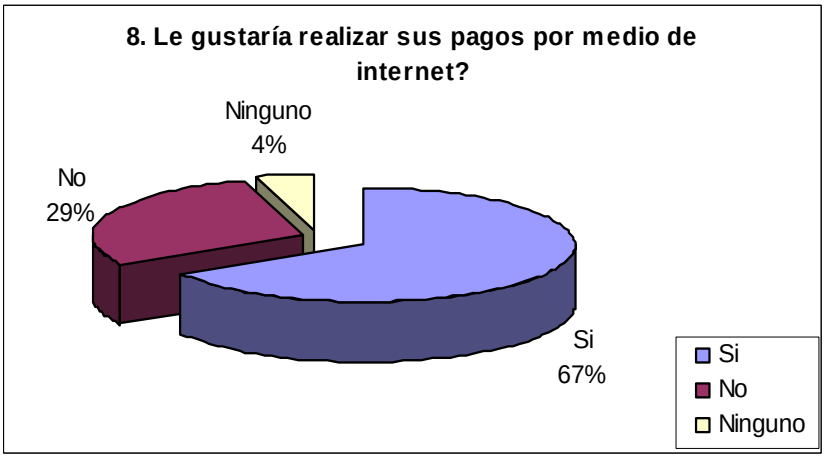
6. ¿Dónde realiza generalmente sus pagos?



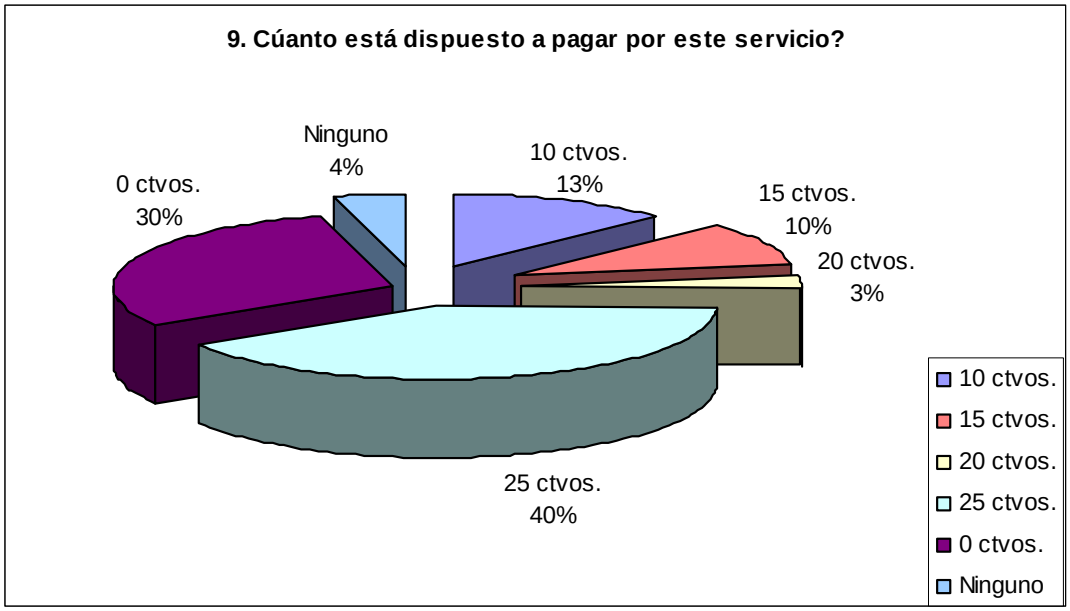
7. ¿Está dispuesto a pagar un costo adicional por este servicio?



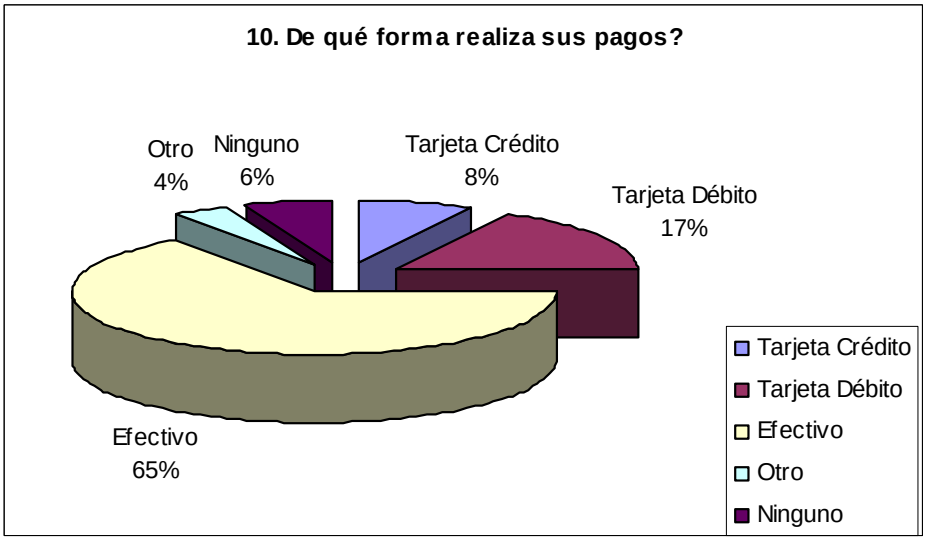
8. ¿Le gustaría realizar sus pagos por medio de Internet?



9. ¿Cuánto está dispuesto a pagar por este servicio?



10. ¿De qué forma realiza sus pagos?



11. Sugerencias de ¿cómo le gustaría que sean sus pagos por Internet?

La mayoría de personas no contestaron esta pregunta, pero de las que lo hicieron contestaron que sea rápido, no tenga información que no esté relacionada con el pago, fácil de navegar.

Gran parte de las personas que contestaron esta pregunta dijeron que el pago debe ser seguro y confiable. Fueron pocas las personas que contestaron que no confían en los pagos por Internet.

APÉNDICE B

CONCEPTOS SOBRE SEGURIDAD INFORMÁTICA

Seguridad de Información:

Al hablar de seguridad de información nos referimos a la protección de los datos contra el acceso a ellos por parte de personas que no tienen autorización (confidencialidad), que no sean modificados o alterados en la misma máquina donde se originaron o durante el proceso de transmisión (integridad) y que puedan ser accedidos en cualquier momento (disponibilidad).

Ataques de seguridad

Existen muchas formas de comprometer la seguridad de información, más adelante se describe algunos tipos de ataques.

Primero, se identificarán los tipos de intrusos que existen:

Internos: Son personas que pertenecen a la organización y tienen mayor facilidad para acceder a los recursos de la red.

Externos: Personas que no tienen relación con la organización, pero que intentan adivinar alguna contraseña o vulnerabilidad para poder acceder a la red.

Ahora se describirán algunos de los tipos de ataque que existen:

- *Eavesdropping*

  Este tipo de ataque se lo realiza con la ayuda de programas, que monitorean todos los paquetes y a su vez viajan desde y hacia la máquina, en donde están instalados. Este tipo de ataque es utilizado principalmente para conocer contraseñas o números de tarjetas de crédito no encriptadas.

- *Snooping*

  EL objetivo de este ataque es el mismo que el anterior, pero la forma de realizarlo es diferente. Con este tipo de ataque, intentamos tener acceso a documentos de cualquier tipo, que contengan información que comprometan la seguridad de alguna entidad y en algunos casos, tratan de copiar esa información a la máquina de quien realiza el ataque.

- *Tampering o Data Diddling*

  Este tipo de ataque consiste en modificar o eliminar archivos, o alterar el funcionamiento normal de algún software. El objetivo puede ser realizar algún fraude o dejar fuera de servicio a algún servidor o incluso toda una red.

- *Spoofing*

  Este ataque se lo realiza accediendo a un sistema haciéndose pasar por otra persona, esto se lo hace, con un usuario y una contraseña obtenida por métodos ilícitos. Este tipo de ataque, permite realizar acciones como enviar correo, acceder a sistemas en nombre de otra persona no siempre con buenas intenciones.

- *Jamming o Flooding*

  Con este ataque se consigue la denegación del servicio, esto lo consiguen inundando toda la red con paquetes que no tienen sentido o consumiendo todos los recursos del sistema.

- *Caballos De Troya*

  Consiste en alterar algún programa, insertando líneas de código que realicen acciones diferentes, como: formatear el disco duro, eliminar o modificar archivos, etc.

- *Ingeniera Social*

  Consiste en obtener información, como: el usuario y la contraseña de algún amigo(a) o de alguna persona conocida; también, se lo puede conseguir llamando a una persona y preguntarle, diciendo que es el administrador de la red.

- *Difusión de Virus*

  Los virus pueden ingresar al sistema sin la intervención de un intruso, se lo puede hacer con ayuda de un dispositivo de almacenamiento, como: disquete, cd, etc. o por medio del correo electrónico. Este tipo de ataque es el más común en la mayoría de empresas.

Criptografía

Es la técnica para transformar texto claro (no encriptado) en texto cifrado (encriptado) o criptograma, que será entendido solo por las persona que están autorizados para hacerlo; para poder descifrarlo es necesaria una clave.

Encriptación

Consiste en transformar un texto inteligible, en otro que sea imposible de entender. El método utilizado para encriptar se denomina algoritmo de encriptación.

Desencriptación

Es lo contrario de encriptación, consiste en transformar un texto difícil de entender, a un texto claro.

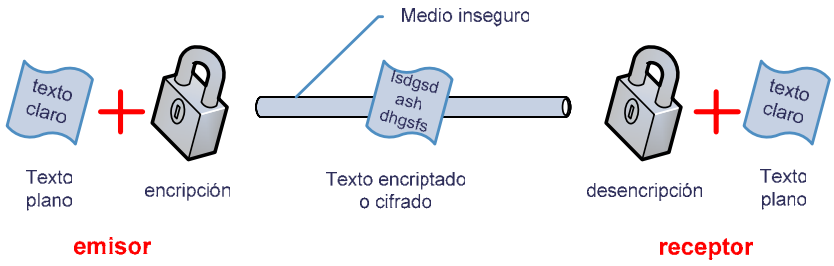


Figura B.1. Encriptación y desencriptación de datos

Los sistemas criptográficos se clasifican en: simétricos o de clave secreta, y asimétricos o de clave pública.

Sistemas criptográficos simétricos:

La criptografía simétrica o de clave secreta, utiliza una sola clave tanto para encriptar como para desencriptar los datos, por lo tanto el emisor y el receptor deben conocer la misma clave. Se debe tener cuidado al momento de elegir el mecanismo para difundir la clave, ya que si alguien más conoce la clave, el sistema deja de ser seguro.

Como ventaja de los algoritmos simétricos tenemos que son más sencillos; por lo tanto el proceso de encriptación y desencriptación es más rápido.

Entre los principales algoritmos simétricos, tenemos: DES, IDEA y RC5. Las técnicas más comunes, en la criptografía de clave secreta son: block Cipher, stream Cipher, Message Authentication code

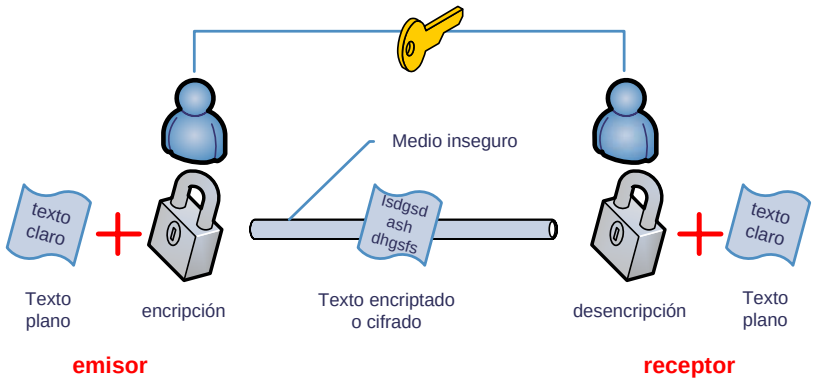


Figura B.2. Criptografía de clave secreta

Sistemas criptográficos asimétricos:

Los sistemas asimétricos, utilizan dos claves: una pública y una privada. Cuando se quiere enviar un mensaje, el emisor utiliza la clave privada para encriptarlo, este mensaje puede ser

desencriptado por el receptor solo con la clave pública. Esto ocurre también de forma inversa, un mensaje encriptado con la clave pública puede ser desencriptado solo con la clave privada.

La clave privada es conocida únicamente por el emisor del mensaje y la clave pública, es difundida por Internet y puede conocerla cualquier persona.

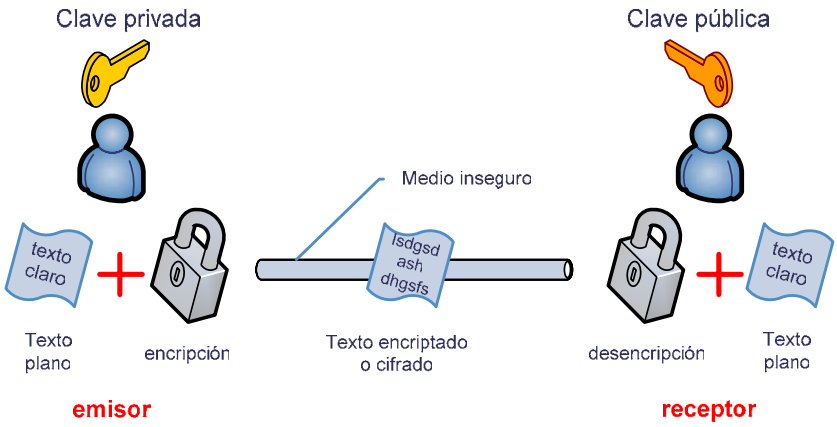


Figura B.3. Criptografía de clave pública

Estas claves son generadas por algoritmos matemáticos complejos, que se encargan de generar el par de claves, haciendo que la una tenga relación con la otra. Estos sistemas tienen como desventaja que son más lentos.

Algoritmos que utilizan criptografía de clave pública, son: Diffie-Hellman, RSA, CCE (Criptografía con curvas elípticas)

Por lo general, se utilizan sistemas de clave pública para enviar de forma segura una clave simétrica y los sistemas de clave secreta se usan para el envío de datos.

Autenticación y Autorización

Autenticación

Es el proceso mediante el cual se comprueba que alguien, es quien dice ser. Una forma de autenticación puede ser por medio de un usuario y una contraseña, el usuario debe ingresar la contraseña, el sistema debe comprobar que la contraseña realmente pertenece a ese usuario.

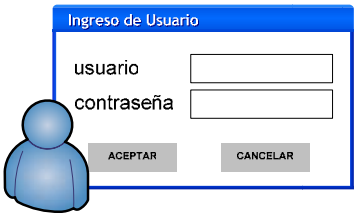


Figura B.4. Autenticación

Autorización

Una vez realizada la autenticación, se debe verificar los permisos de acceso a los recursos del sistema para ese usuario. La autorización, es el permiso para acceder a los recursos de un sistema.

Certificados Digitales

Un certificado digital es un documento electrónico, que contiene: los datos de identificación del propietario, su clave pública y la firma digital de una autoridad certificadora. Cuando una persona solicita un certificado digital se genera un par de claves, la pública, que está incluida en el certificado y la privada, que únicamente la conoce el propietario, quien debe cuidar que nadie más tenga acceso a ella. Además, el certificado tiene una firma digital de una entidad, que respalda la validez de los datos del propietario.

Los certificados digitales se basan en el estándar X.509, que es quien define la sintaxis de los mismos. Según este estándar, los certificados digitales deben contener los siguientes campos:

- Versión
- Número de serie
- Firma
- Emisor

- Validez
- Propietario o sujeto
- Clave pública del propietario
- Identificadores únicos de emisor y propietario
- Campos de extensión

Protocolos de seguridad: SSL, TSL, SET

Los protocolos de seguridad establecen las reglas para que se de, una comunicación segura entre un emisor y un receptor, a través de un medio inseguro. Como ejemplo de estos protocolos se tiene: SSL, TSL y SET.

SSL (Secure Socket Layer)

Fue desarrollado en 1994, por Netscape Communications Corporation, la versión tres. Es soportado por los navegadores principales, como son: Internet Explorer y Netscape.

Figura B.5. Protocolo de seguridad SSL

Lo siguientes pasos se dan en el establecimiento de un canal seguro de comunicación con SSL:

Solicitud de SSL

1. El cliente inicia una sesión segura con el servidor web, generalmente se lo hace escribiendo **https://**

Handshake SSL:

2. El cliente le pide al servidor que se identifique (con su certificado digital) y le informa también, sobre los algoritmos que tiene para encriptar los datos.
3. El servidor se presenta (con el certificado digital), le envía su identificador digital encriptado, su clave pública y el algoritmo de

encriptación (selecciona el mejor algoritmo soportado por ambos: servidor y cliente)

4. El cliente desencripta el identificador digital con la clave pública recibida, verifica que el certificado haya sido emitido por una autoridad certificadora segura. Luego verifica la validez del certificado, revisando fechas, URL, etc. Por último, genera una clave secreta usando la clave pública del servidor, la encripta y la envía al servidor.
5. El servidor y el cliente conocen la clave secreta. Para asegurar que la información no ha cambiado, ambos se envían la clave secreta, si coinciden, el handshake concluye.

Intercambio de datos:

6. Ahora, ambos pueden intercambiar información encriptándola y desencriptándola con la clave secreta.

Terminación de SSL:

7. Cuando el cliente abandona el servidor se le informa que termina la sesión segura.

Los algoritmos que utiliza SSL para encriptar los datos son: DES, TDES, RC2, RC4, MD5, SHA-1, DH y RSA.

Tiene las siguientes desventajas:

- Protege las transacciones solo entre el servidor web y el cliente, por esta razón, no es una buena opción para transacciones de pago con tarjeta de crédito que requieren como mínimo tres partes: el vendedor, el comprador y el emisor de tarjetas,
- No asegura al comprador de que un vendedor deshonesto pueda utilizar ilícitamente su tarjeta.
- El vendedor, corre el riesgo de que el número de tarjeta sea inválido o que no haya sido aprobada.

A pesar de estas desventajas, es ampliamente usado ya que es más fácil de implementar que otros protocolos.

TLS (Transport Layer Security):

Este protocolo está basado en la versión tres de SSL, pero tiene algunas mejoras, fue estandarizado por el IETF y no pertenece a una empresa privada.

SET (Secure Electronic Transaction):

El estándar SET fue desarrollado en 1995 por Visa y MasterCard, con la colaboración de otras compañías líderes en el mercado de las tecnologías de la información, como: Microsoft, IBM, Netscape, SAIC, GTE, RSA, Terisa Systems, VeriSign y otras.

SET, es una especificación diseñada con el propósito de asegurar y autenticar la identidad de los participantes en las compras, abonadas con tarjetas de crédito/débito en cualquier tipo de red en línea.

Figura B.6. Protocolo de seguridad SET

Una transacción utilizando el protocolo de seguridad SET, se la realiza de manera similar a una compra convencional, utilizando una tarjeta de crédito requiere de los siguientes pasos:

1. El cliente decide comprar (pagar) un artículo. El protocolo SET se inicia cuando el comprador pulsa el botón de Pagar.
2. El servidor del comerciante envía una descripción del pedido, que inicia a la aplicación monedero del cliente.
3. El cliente comprueba el pedido y transmite una orden de pago de vuelta al comerciante. La aplicación monedero, crea dos mensajes que envía al comerciante. El primero, la información del pedido, contiene los datos del pedido, mientras que el segundo, contiene las instrucciones de pago del cliente (número de tarjeta de crédito, banco emisor, etc.) para el banco adquiriente (banco del comerciante). En este momento, el software monedero del cliente genera un firma dual, que permite juntar en un solo mensaje la información del pedido y las instrucciones de pago, de manera, que el comerciante puede acceder a la información del pedido, pero no a las instrucciones de pago, mientras que el banco puede acceder a las instrucciones de pago, pero no a la información del pedido. Este mecanismo reduce el riesgo de fraude y abuso,

ya que ni el comerciante llega a conocer el número de tarjeta de crédito empleado por el comprador, ni el banco se entera de los hábitos de compra de su cliente.

4. El comerciante envía la petición de pago a su banco. El software SET en el servidor del comerciante crea una petición de autorización, que envía a la pasarela de pagos, incluyendo el importe a ser autorizado, el identificador de la transacción y otra información relevante acerca de la misma, todo ello convenientemente cifrado y firmado. Entonces, se envía al banco adquiriente la petición de autorización, junto con las instrucciones de pago (que el comerciante no puede examinar, ya que van cifradas con la clave pública del adquiriente).
5. El banco adquiriente valida al cliente y al comerciante, y obtiene una autorización del banco emisor del cliente. El banco del comerciante descifra y verifica la petición de autorización. Si el proceso tiene éxito, obtiene a continuación las instrucciones de pago del cliente, que verifica a su vez, para asegurarse de la identidad del titular de la tarjeta y de la integridad de los datos. Se comprueban los identificadores de la transacción en curso (el enviado por el comerciante y el codificado en las instrucciones de pago) y, si todo es correcto, se formatea y envía una petición de autorización al banco

emisor del cliente, a través de la red de medios de pago convencional.

6. El emisor autoriza el pago. El banco emisor, verifica todos los datos de la petición y si todo está en orden y el titular de la tarjeta posee crédito, autoriza la transacción.
7. El adquiriente envía al comerciante un testigo de transferencia de fondos. En cuanto el banco del comerciante recibe una respuesta de autorización del banco emisor, genera y firma digitalmente un mensaje de respuesta de autorización, que envía a la pasarela de pagos, convenientemente cifrada, la cual se la hace llegar al comerciante.
8. El comerciante envía un recibo al monedero del cliente. Cuando el comerciante recibe la respuesta de autorización de su banco, verifica las firmas digitales y la información, para asegurarse de que todo está en orden. El software del servidor, almacena la autorización y el testigo de transferencia de fondos. A continuación, completa el procesamiento del pedido del titular de la tarjeta, enviando la mercancía o suministrando los servicios pagados.
9. Más adelante, el comerciante usa el testigo de transferencia de fondos para cobrar el importe de la transacción. Después de haber completado el procesamiento del pedido del titular

de la tarjeta, el software del comerciante genera una petición de transferencia a su banco, confirmando la realización con éxito de la venta. Como consecuencia, se produce el abono en la cuenta del comerciante.

10. A su debido tiempo, el dinero se descuenta de la cuenta del cliente (cargo).

SET, esta diseñada para asegurar las transacciones de pago a través de Internet, realizadas con tarjeta de crédito. A diferencia, de SSL, SET proporciona seguridad a todos los participantes en un proceso de pago usando tarjeta de crédito.

Firewall

Es un sistema que permite, controlar el acceso a una red por parte de usuarios que pertenecen a una red no segura. Generalmente, un firewall es colocado entre la red interna y la red externa como se muestra en la siguiente figura:

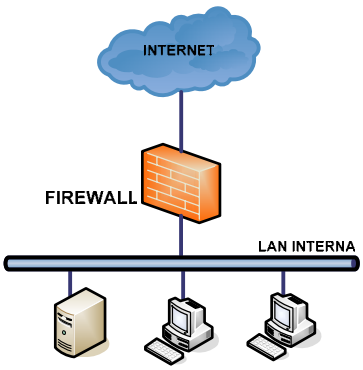


Figura B.7. Configuración típica de un Firewall

APÉNDICE C

DIAGRAMAS DE SECUENCIA

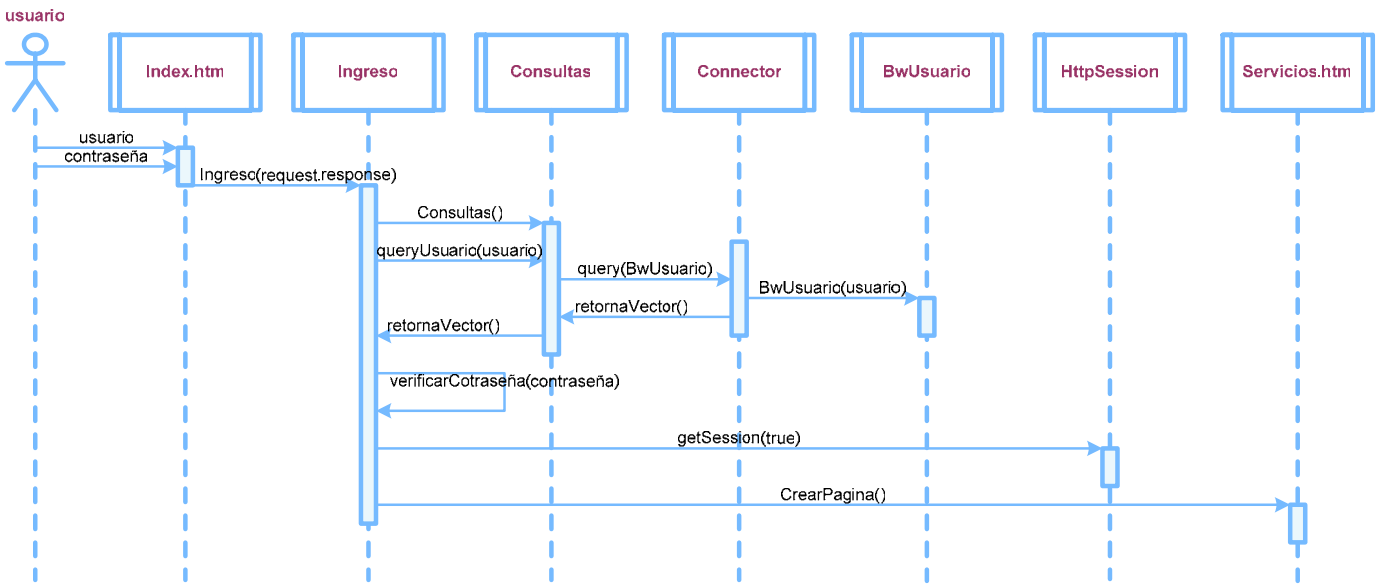


Figura C.1 Diagrama de Secuencia de Inicio de Sesión

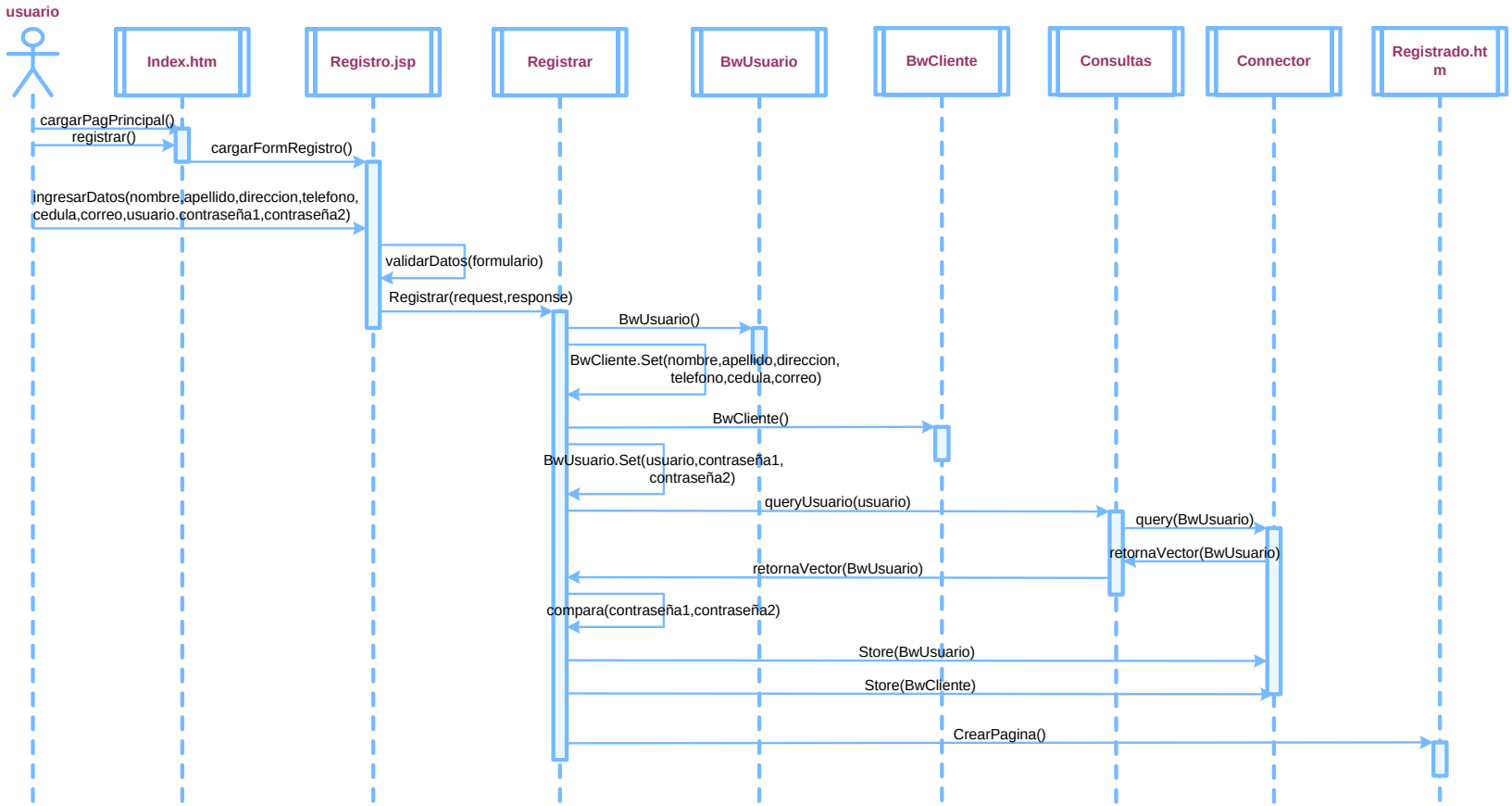


Figura C.2 Diagrama de Secuencia de Registro de Usuario

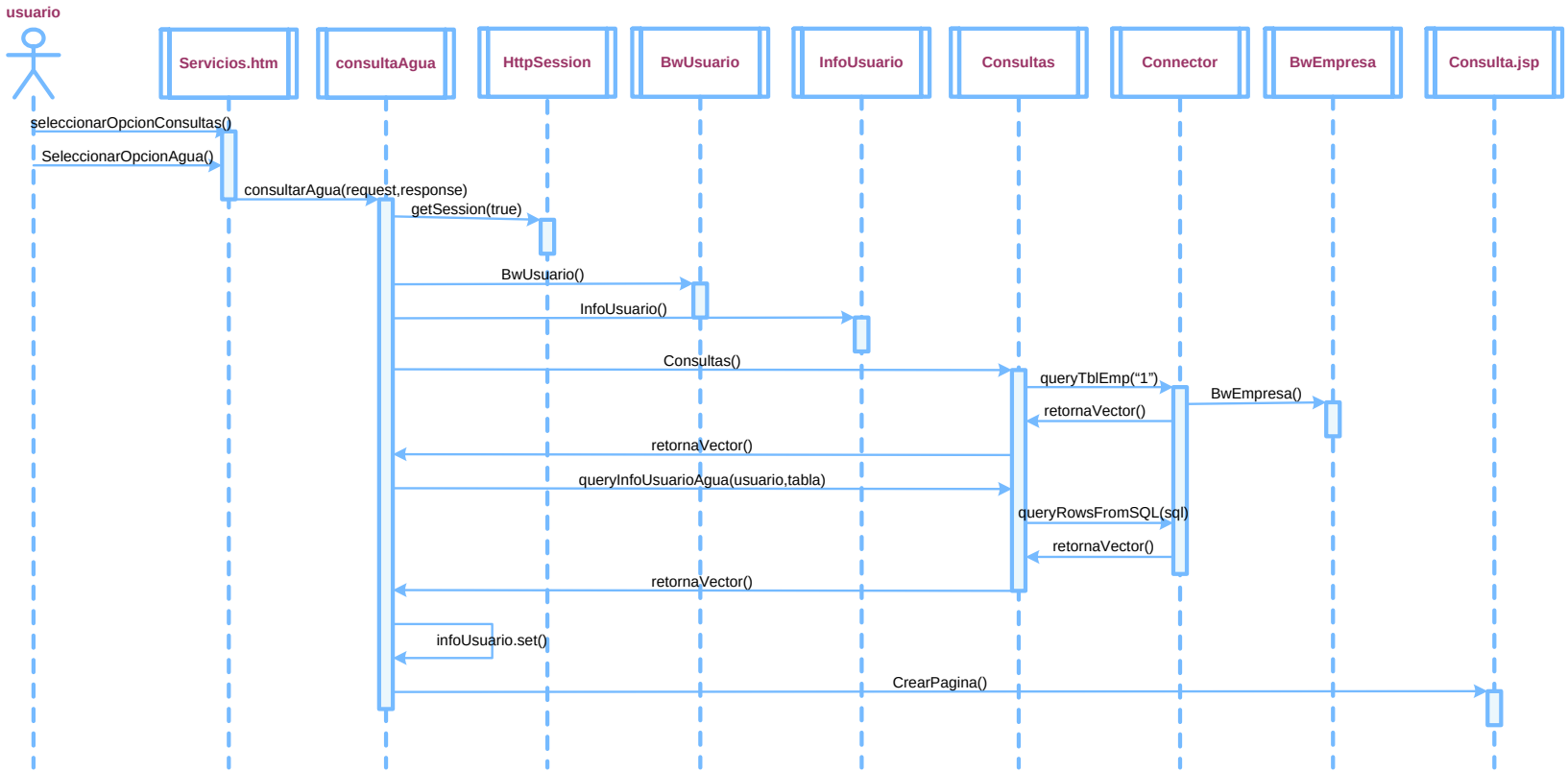


Figura C.3 Diagrama de Secuencia de Consultar Planilla de Agua

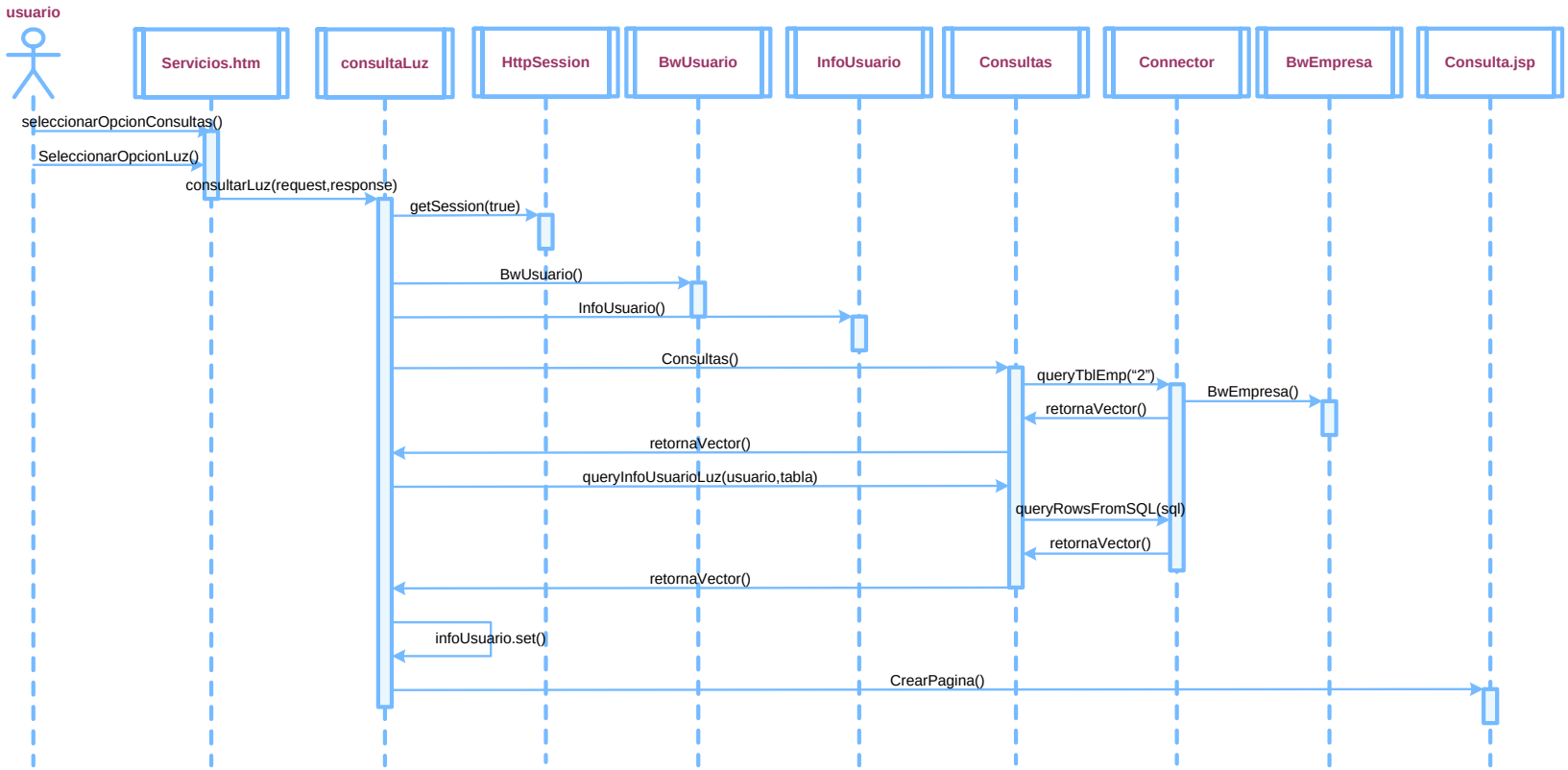


Figura C.4 Diagrama de Secuencia de Consultar Planilla de Luz

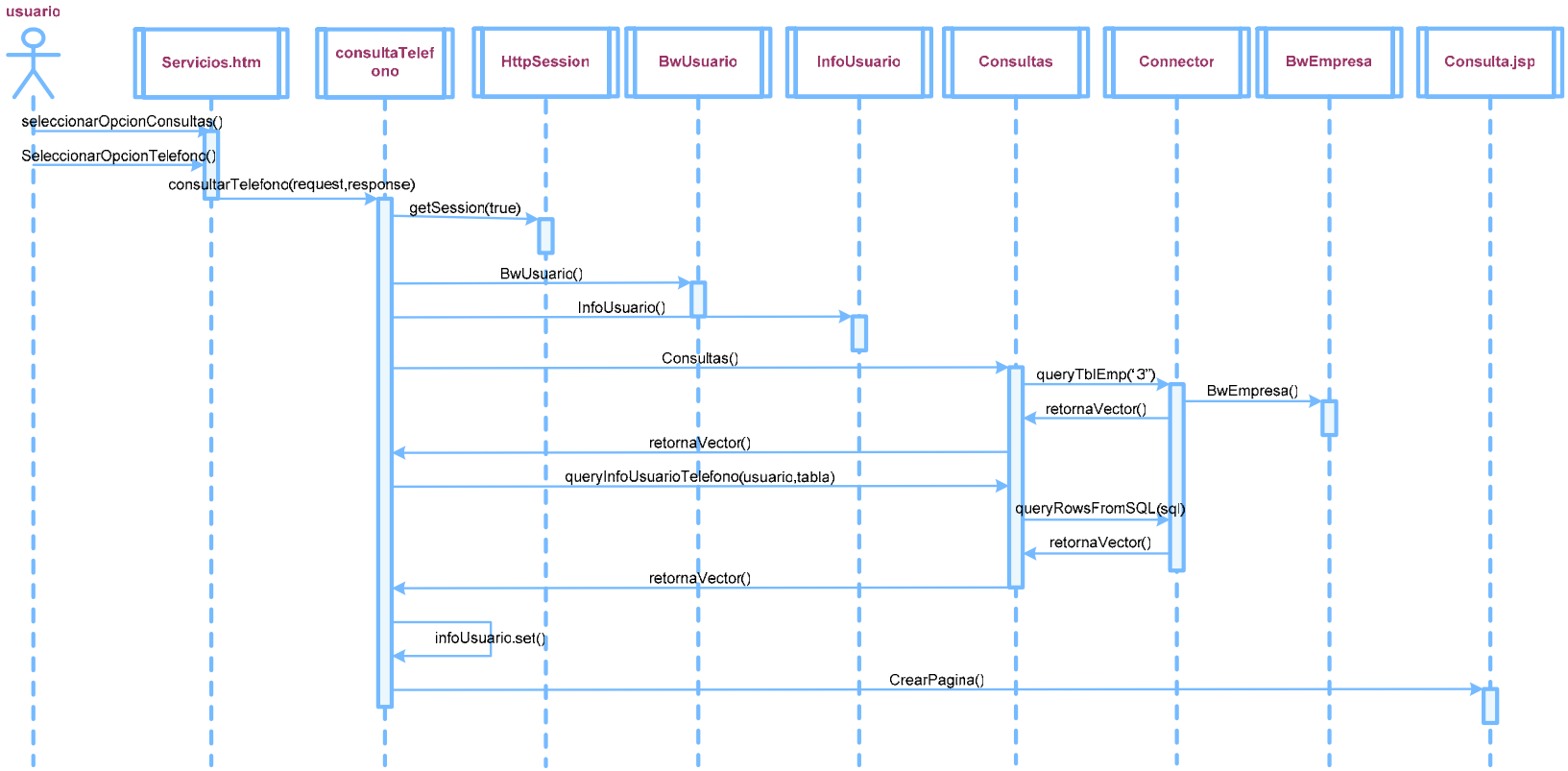


Figura C.5 Diagrama de Secuencia de Consultar Planilla de Teléfono

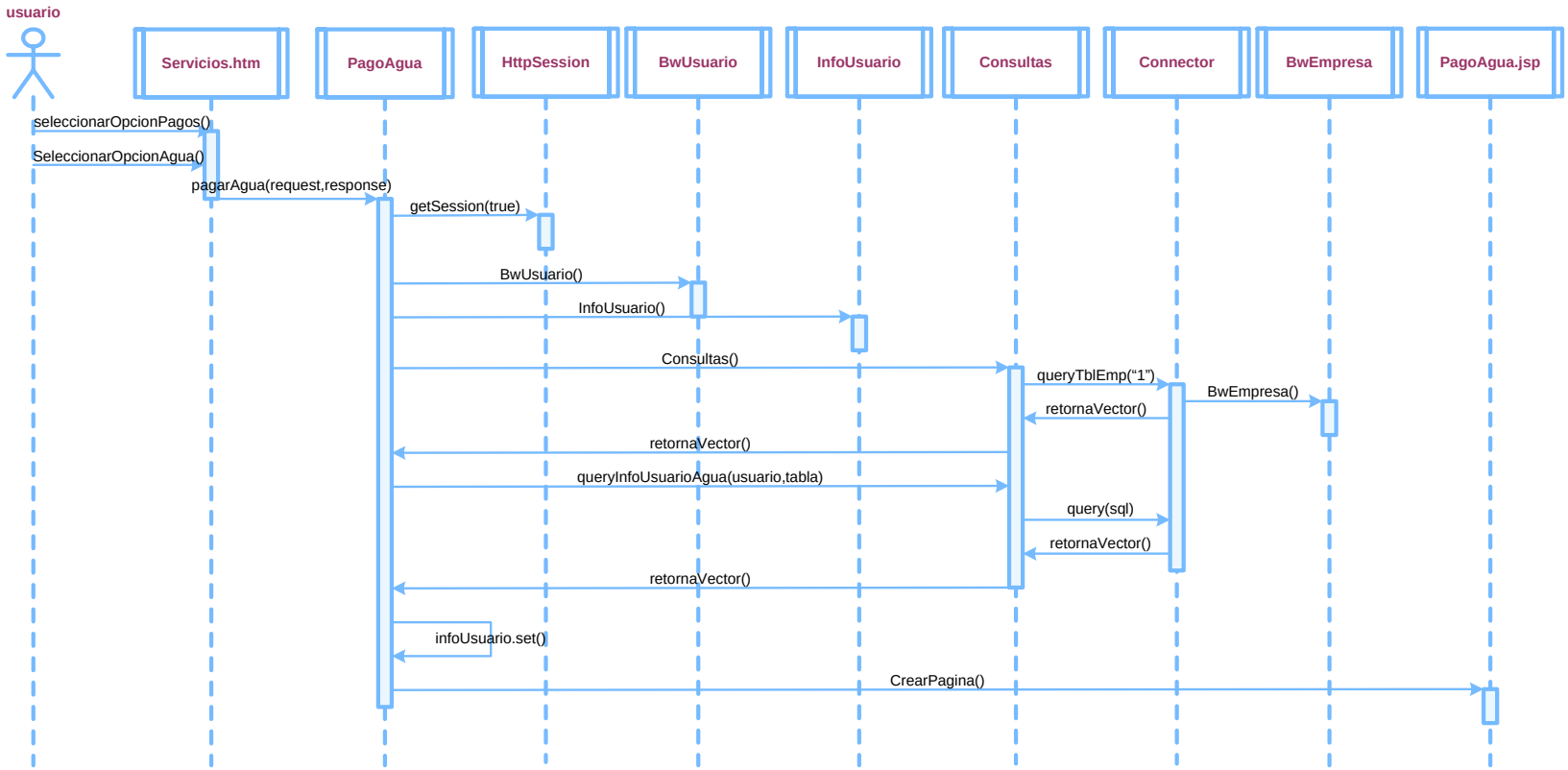


Figura C.6.a Diagrama de Secuencia de Seleccionar Opción Pagar Planilla de Agua

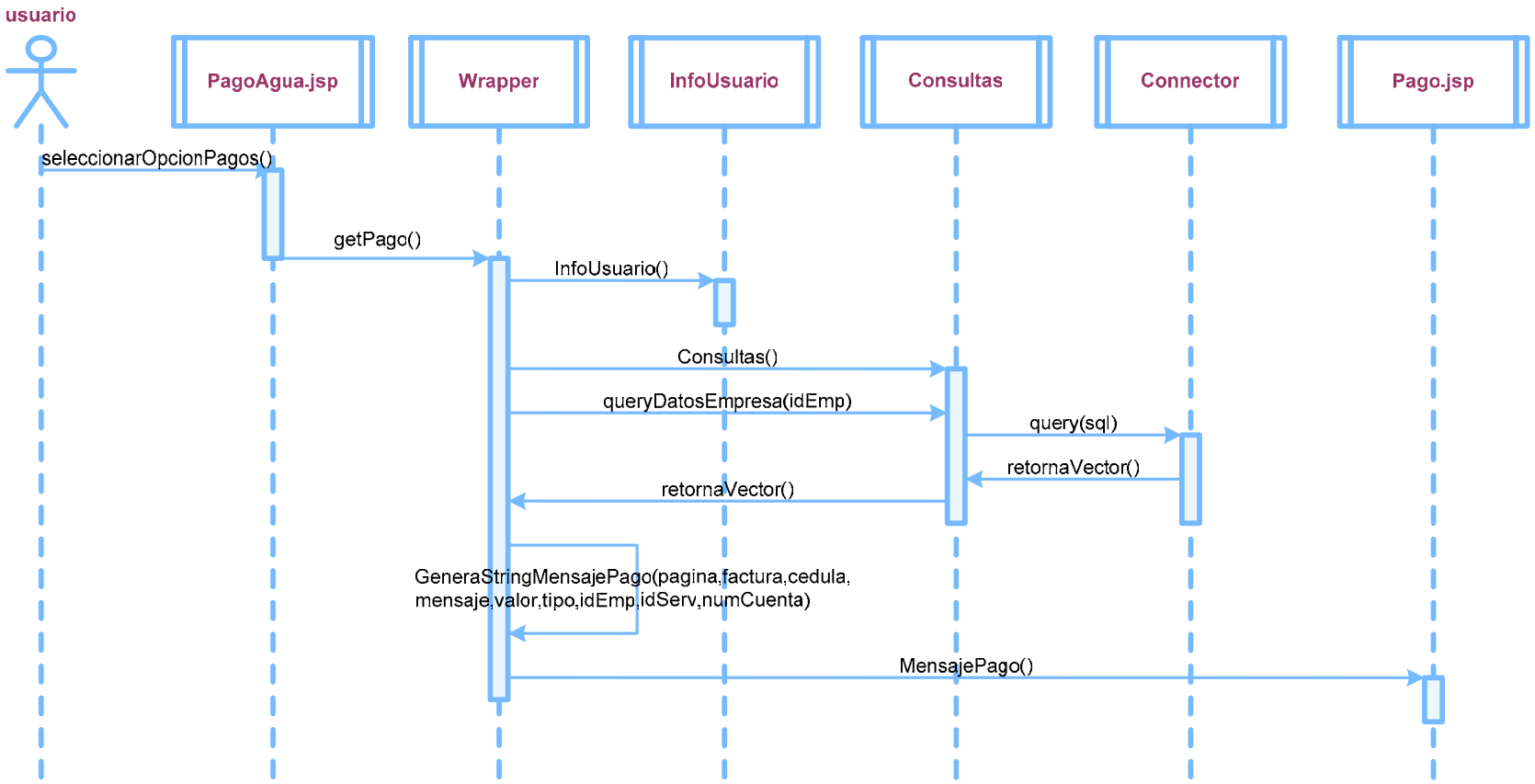


Figura C.6.b Diagrama de Secuencia de Selección de planilla de Agua a pagar

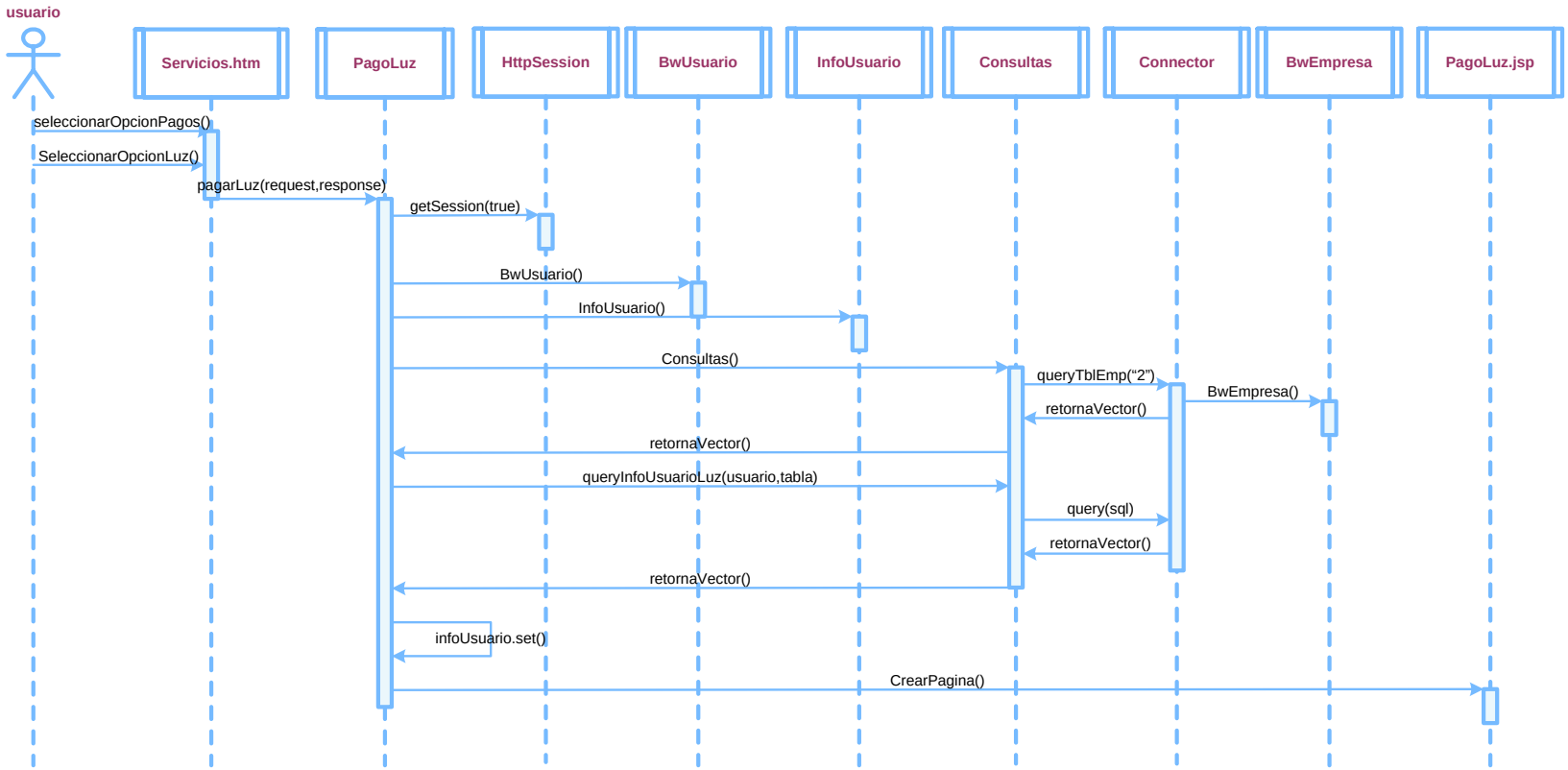


Figura C.7.a Diagrama de Secuencia de Seleccionar Opción Pagar Planilla de Luz

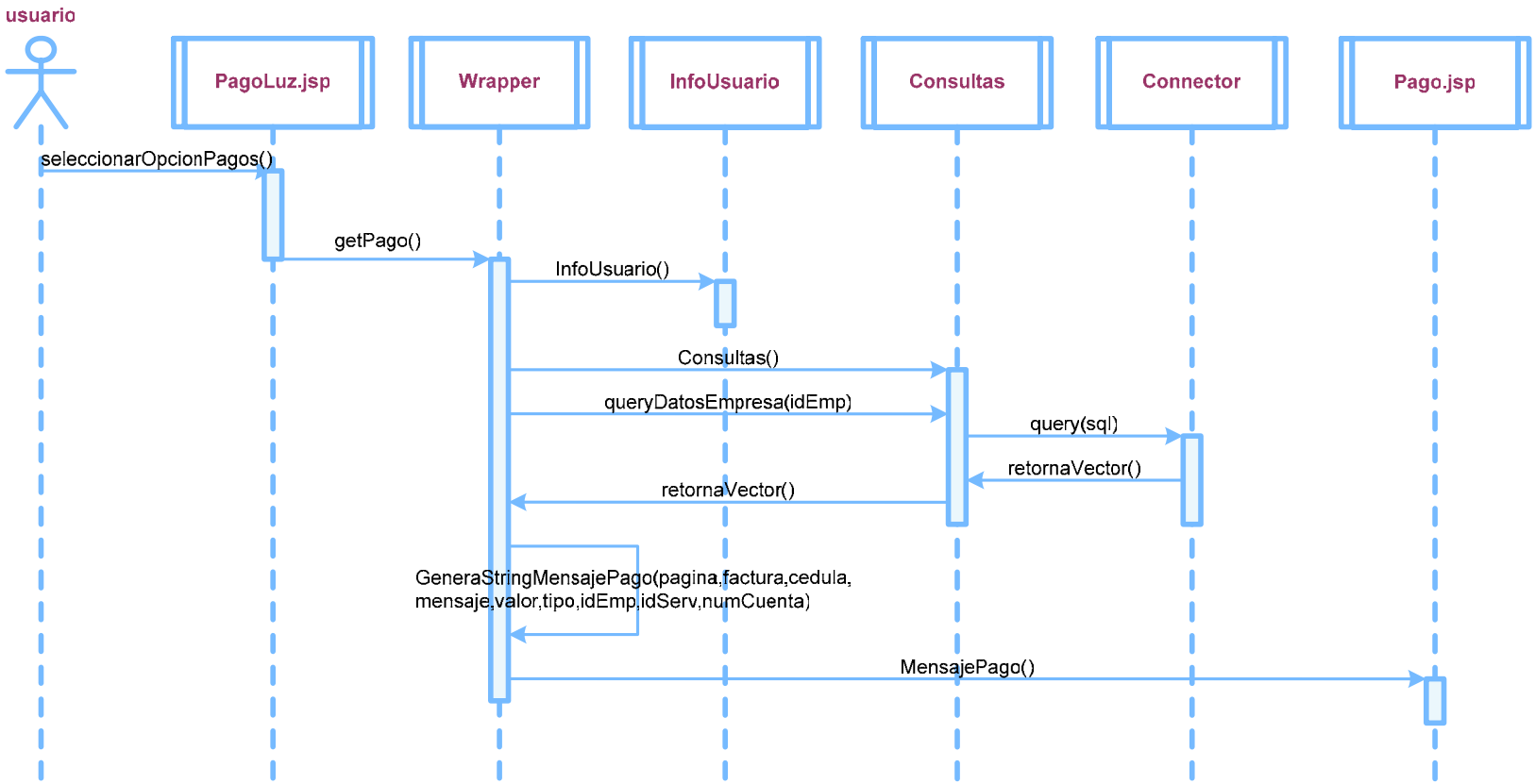


Figura C.7.b Diagrama de Secuencia de Selección de planilla de Luz a pagar

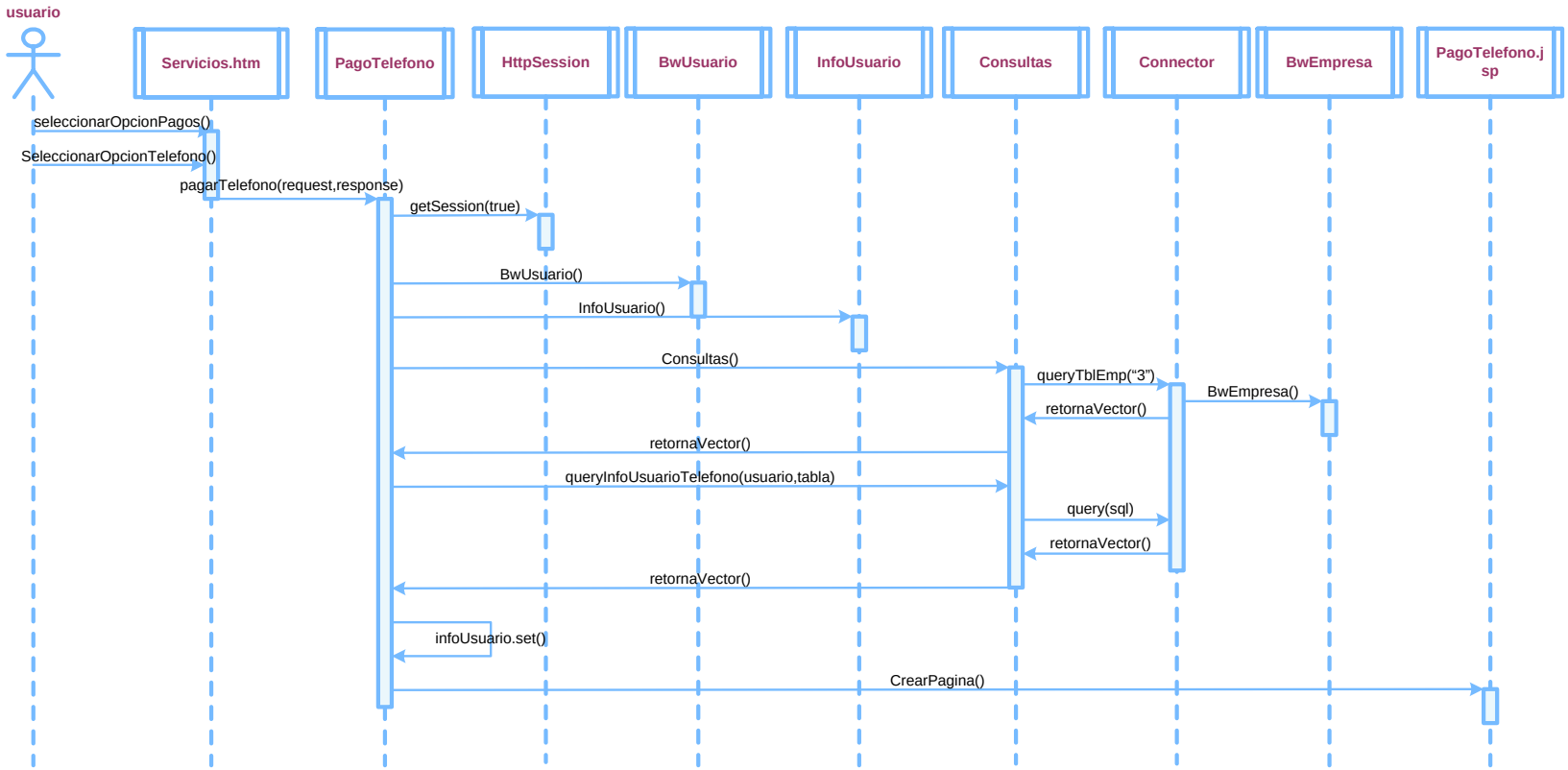


Figura C.8.a Diagrama de Secuencia de Seleccionar Opción Pagar Planilla de Teléfono

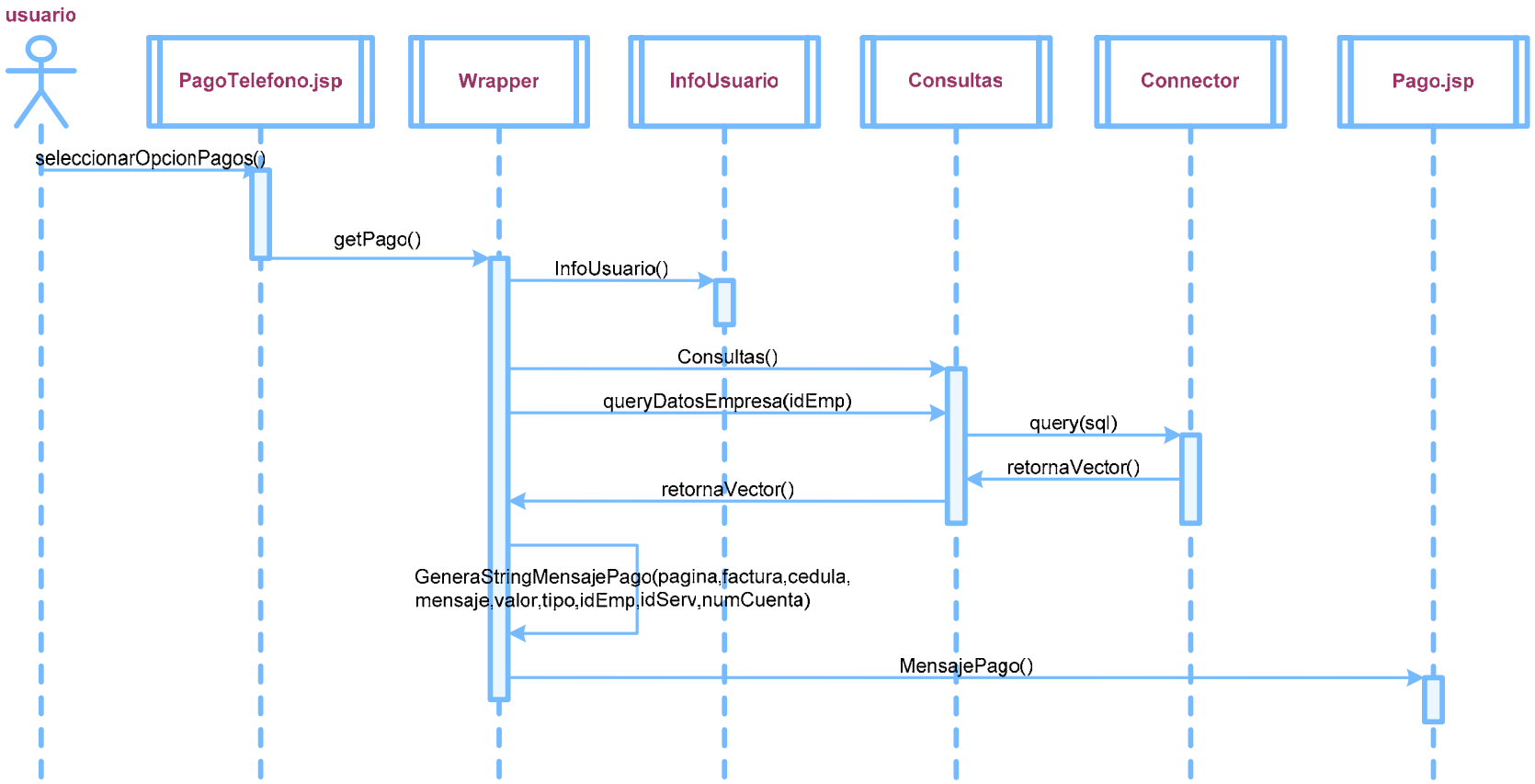


Figura C.8.b Diagrama de Secuencia de Selección de planilla de Teléfono a pagar

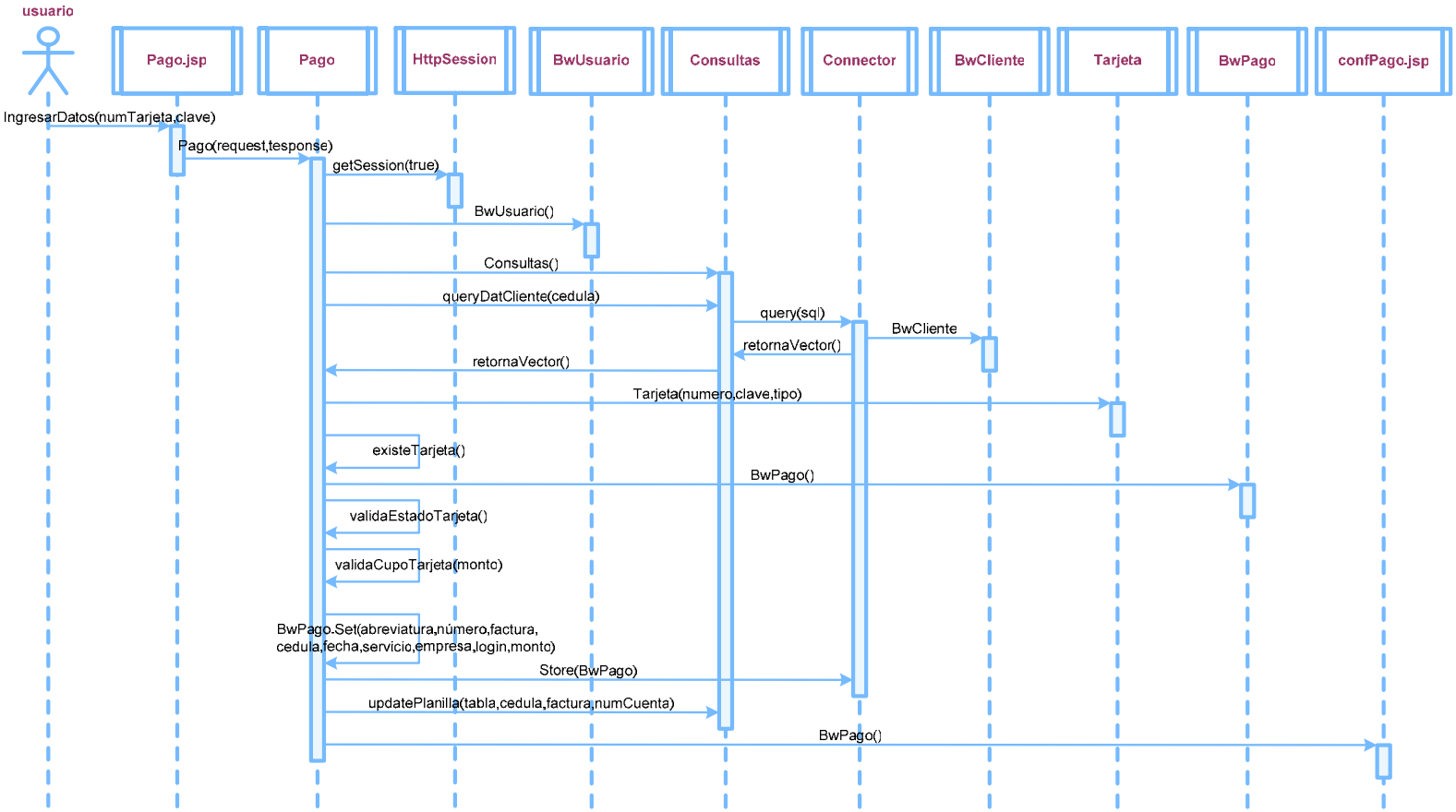


Figura C.9 Diagrama de Secuencia de Pago de Planilla

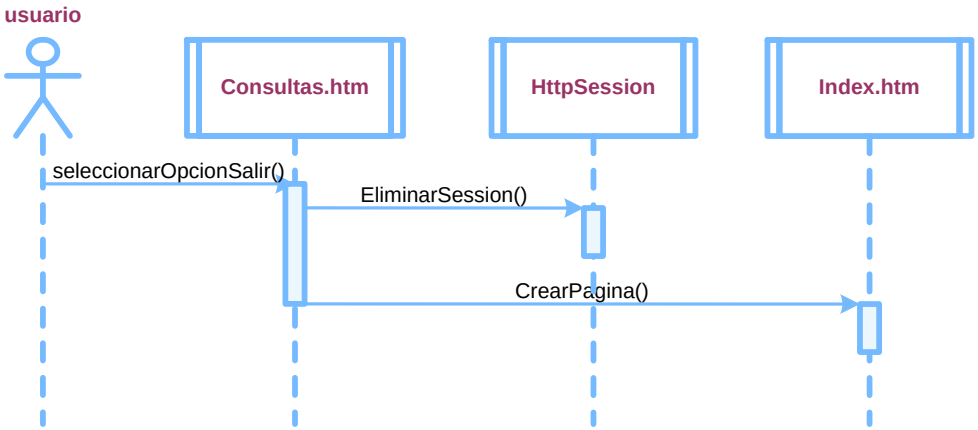


Figura C.10 Diagrama de Secuencia de Salir del Sistema

APÉNDICE D

INSTALACIÓN DE HERRAMIENTAS DE SOFTWARE

Servidor Web

Paquetes a usar:

- httpd-2.0-49.tar.gz
- j2sdk-1_4_2_04-linux-i586.bin
- jakarta-tomcat-4.1.30.tar.gz
- jakarta-tomcat-connectors-jk2-2.0.4-src.tar.gz

Instalación de Apache 2.0.49 como servidor seguro (mod_ssl)

Nota:

En apache Versión 2 ya vine incorporado el módulo mod_ssl, si se está usando apache Versión 1 necesita bajarse el modulo mod_ssl.

Los comandos a continuación deben ser ejecutados desde un terminal (xterm ó gnome-terminal) en Sistemas Linux.

Debemos estar como usuario root, ejecutar:

su – root

Verificamos que umask sea 0022 con el comando:

umask

Si no esta seteada a 0022 entonces, ejecutar:

```
# umask 0022
```

Descomprimimos el paquete httpd-2.0.49.tar.gz

Nos ubicamos en el directorio donde se encuentra el paquete, ejecutamos:

```
# tar xvzf httpd-2.0.49.tar.gz
```

Nos cambiamos de directorio:

```
# cd httpd-2.0.49
```

Configuramos el Fuente:

```
# CPPFLAGS="-I/usr/kerberos/include -DESECURITY_HOLE_PASS_AUTHORIZATION"
./configure --prefix=/usr/local/apache --with-mpm=worker --enable-mods-
shared=all --enable-ssl=shared --enable-cgi
```

Todas las tres líneas anteriores deben ser ejecutadas en una sola línea.

Luego ejecutamos:

```
# make
# make install
```

Editamos el archivo de configuración del Apache (httpd.conf), que se encuentra en /usr/local/apache/conf/httpd.conf, en la línea 218 encontramos

Listen 80, que es el puerto donde va a escuchar las peticiones el Apache; si uno desea, puede cambiar el 80.

Descomentamos la línea 326, ServerName y colocamos el nombre de nuestro portal con su respectivo puerto.

ServerName www.algo.com:puerto

Luego levantamos el apache con

/usr/local/apache/bin/apachectl start

y luego abrimos un browser y colocamos

http://localhost:80

o podemos escribir

http://localhost

si cambiaron el puerto default (80), entonces si hay que colocar el puerto

http://localhost:puerto

Bajamos el apache

/usr/local/apache/bin/apachectl stop

Instalación de java.

No se puede usar el JRE (Java Run Time) por lo tanto debemos bajar todo el paquete de java.

Para instalar solo tenemos que copiar j2sdk-1_4_2_04-linux-i586.bin en la carpeta /usr/local y luego ejecutar:

```
# ./j2sdk-1_4_2_04-linux-i586.bin
```

Creamos un link simbólico

```
# ln -s /usr/local/j2sdk-1_4_2_04 /usr/local/java
```

Crear un enlace simbólico en /usr/local, para que apunte a la carpeta j2sdk-1_4_2_04 que se creó al descomprimir el paquete.

Añadir JAVA_HOME como variable de ambiente. Para añadir necesitamos editar el archivo /etc/profile y ubicar al final del archivo la siguiente línea:

```
export JAVA_HOME=/usr/local/java
```

Instalación y Configuración Tomcat4.1.30

Ingresar como usuario root

su – root

Ubicarse en el directorio donde se encuentra el paquete jakarta-tomcat-4.1.30.tar.gz, en nuestro caso esta en /home/cjsr

cd /home/cjsr

Descomprimir el paquete

tar xvzf jakarta-tomcat-4.1.30.tar.gz -C /usr/local

La instrucción -C seguido por la ruta es para especificar donde quiero que se descomprima el paquete.

Crear un enlace simbólico en /usr/local para que apunte a la carpeta que se creó al descomprimir el paquete.

ln -s /usr/local/jakarta-tomcat-4.1.30 /usr/local/tomcat

Un enlace simbólico es lo mismo que un acceso directo en Windows

Añadir CATALINA_HOME como variable de ambiente. Para añadir necesitamos editar el archivo /etc/profile y ubicar al final del archivo las siguientes líneas:

```
export CATALINA_HOME=/usr/local/tomcat
export PATH=$PATH:$CATALINA_HOME/bin
```

Luego cerramos el archivo.

Crear usuario y grupo tomcat.

```
# groupadd tomcat
# useradd -g tomcat -c  "Tomcat User" tomcat
```

Cambiar de propietario a algunas carpetas que están en el directorio de Tomcat4.1.30.

```
# cd/usr/local/tomcat
# chown -R tomcat:tomcat conf temp webapps work logs
```

Ahora ya podemos subir o ejecutarTomcat.

Ingresar con el usuario Tomcat:

```
# su – tomcat
```

Ejecutar el siguiente comando:

```
# /usr/local/tomcat/bin/startup.sh
```

Probar en un navegador con la siguiente dirección

http://localhost:8080/

Notar que tomcat trabaja en el puerto 8080.

Para bajar tomcat, ejecutar:

```
# /usr/local/tomcat/bin/shutdown.sh
```

Compilar e Instalar el mod_jk2

El mod_jk2 es el módulo que nos ayuda a comunicar Apache con Tomcat, esta comunicación también se la puede hacer con el mod_jk. Además, hay más información de la comunicación con este módulo en internet.

Extraer el paquete del conector mod_jk2 en /usr/local/src

```
# tar zxvf jakarta-tomcat-connectors-jk2-2.0.4-src.tar.gz -C /usr/local/src
```

Cambiarse al directorio donde se extrajo el paquete del mod_jk2

```
cd /usr/local/src/jakarta-tomcat-connectors-jk2-2.0.4-src/jk/native2
```

Cambiar los permisos.

```
# chmod 755 buildconf.sh
```

Ejecutar:

```
# ./buildconf.sh
```

```
# ./configure --with-apxs2=/usr/local/apache/bin/apxs  --with-
tomcat41=/usr/local/tomcat/ --with-java-home=/usr/local/java -with-jni
```

Las dos líneas anteriores deben ser ejecutadas en una sola línea. Donde /usr/local/apache es donde tenemos instalado el apache y /usr/local/tomcat es donde esta apuntando la variable de ambiente CATALINA_HOME

Podemos observar el contenido de la variable de ambiente CATALINA_HOME escribiendo:

```
# echo $CATALINA_HOME
```

Ejecutar:

```
# make
# libtool --finish /usr/local/apache/modules
```

Copiamos el modulo generado que es el mod_jk2.so y también libjkjni.so.

```
# cp /usr/local/src/jakarta-tomcat-connectors-jk2-2.0.4-
src/jk/build/jk2/apache2/mod_jk2.so  /usr/local/apache/modules/
# cp /usr/local/src/jakarta-tomcat-connectors-jk2-2.0.4-
src/jk/build/jk2/apache2/libjkjni.so /usr/local/apache/modules/
```

Tenemos que configurar a Tomcat para que use AJP13 connector, que es quien nos va a ayudar con el módulo mod_jk2. Editamos el archivo de

configuración de Tomcat, que es el archivo server.xml y se encuentra en /usr/local/tomcat/conf/server.xml

Necesitamos añadir o cambiar si es que ya se encuentran en el archivo server.xml las siguientes líneas:

```
<!-- Define a non-SSL Coyote HTTP/1.1 Connector on port 8080 -->
   <Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
            port="8080" minProcessors="5" maxProcessors="75"
            enableLookups="true" redirectPort="443"
            acceptCount="100" debug="0" connectionTimeout="20000"
            useURIValidationHack="false" disableUploadTimeout="true" />
   <!-- Note : To disable connection timeouts, set connectionTimeout value
    to 0 -->

<!-- Define a Coyote/JK2 AJP 1.3 Connector on port 8009 -->
   <Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
            port="8009" minProcessors="5" maxProcessors="75"
            enableLookups="true" redirectPort="443"
            acceptCount="10" debug="0" connectionTimeout="0"
            useURIValidationHack="false"
protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler"/>
```

Donde la palabra “redirectPort”, es el puerto de nuestro servidor apache seguro, en nuestro caso es el 443; por lo tanto, si es que se encuentra otro número cambiarlo al puerto que usa el apache para el servidor seguro, generalmente “redirectPort” está con el puerto 8443 y se le cambió a 443. También, añadir, cambiar o quitar los comentarios, si es que se encuentra comentada, en los archivos xml. Los comentarios comienzan con <!-- y terminan con -->, así que hay que quitar solo esos símbolos para descomentarlo.

```
<Engine name="Standalone" defaultHost="localhost" debug="0" jvmRoute="tomcat1">
```

Comentamos la línea siguiente parecida a:

```
<!-- <Engine name="Standalone" defaultHost="localhost" debug="0"> -->
```

Grabamos el archivo server.xml

Configuración del mod_jk2 en el Servidor apache

Editar el archivo de configuración de apache httpd.conf que se encuentra en /usr/local/apache/conf/httpd.conf

Insertar después de la línea 216 la siguiente línea

LoadModule jk2_module modules/mod_jk2.so

Crear el archivo de configuración workers2.properties, crearlo en /usr/local/apache/conf/workers2.properties e insertar lo siguiente:

```
[logger]
level=DEBUG

[config:]
file=/usr/local/apache/conf/workers2.properties
debug=0
debugEnv=0

[uriMap:]
info=Maps the requests. Options: debug
debug=0

[workerEnv:]
info=Global server options
timing=1
debug=0
```

```
# Default Native Logger (apache2 or win32 )
# can be overriden to a file logger, useful
# when tracing win32 related issues
#logger=logger.file:0

[lb:lb]
info=Default load balancer.
debug=0

[channel.socket:localhost:8009]
info=Ajp13 forwarding over socket
debug=0
tomcatId=tomcat1
lb_factor=1
group=lb

[status:]
info=Status worker, displays runtime informations
[uri:/jkstatus/*]
info=Display status information and checks the config file for changes.
group=status:
[uri:/examples/*]
```

worker=ajp13:localhost:8009

[uri:/tomcat-docs/*]

worker=ajp13:localhost:8009

Probamos el servidor con mod_jk2, primero bajamos el tomcat y el apache.

/usr/local/tomcat/bin/shutdown.sh

/usr/local/apache/bin/apachectl stop

Subimos el Tomcat y el apache en el orden especificado.

/usr/local/tomcat/bin/startup.sh

Esperamos unos 10 segundos y subimos el apache.

/usr/local/apache/bin/apachectl startssl

Para probar si se ejecutan los Servlets y los Jsp tenemos que abrir nuestro browser y probar.

https://localhost:443/examples/

Tener en cuenta, que la palabra “uri” en el archivo workers2.properties son las carpetas que podemos acceder a tomcat y estas carpetas se encuentran en /usr/local/tomcat/webapps.

Servidor de Base de Datos

Para la instalación de Oracle 9i necesitamos, como mínimo:

- Una máquina Linux con kernel 2.4.7 o mayor.
- JRE 1.1.8v3 o mayor.

Pre – Instalación

- **Variables de ambiente y Perfiles**

  La instalación requiere que ciertas variables de ambientes sean seteadas antes de empezar. Estas variables pueden ser seteadas directamente en el perfil de usuario, en el archivo .bash_profile de cada cuenta, las cuales son usadas para acceder a la base de Datos, o también podemos colocar estas variables directamente en el archivo /etc/profile, donde estas variables se aplicarán en todas las cuentas. En esta instalación vamos a editar el archivo /etc/profile y colocaremos lo siguiente:

```
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/9.2.0.1.0
ORACLE_SID=ORTD
PATH=$PATH:$ORACLE_HOME/bin
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/network/lib
LD_ASSUME_KERNEL=2.4.1
export PATH ORACLE_BASE ORACLE_HOME ORACLE_SID
export LD_LIBRARY_PATH LD_ASSUME_KERNEL
```

- **Instalación JRE**

Descomprimir el paquete del jre118_v3 en el directorio /usr/local:

```
# tar jre118_v3.tar.gz -C /usr/local/
```

Creamos un enlace simbólico a la carpeta que se creó al descomprimir el paquete.

```
# ln –s jre1118_v3 jre
```

Añadimos la ruta del JRE a la variable de ambiente PATH y exportamos la variable PATH. Es mejor colocar las siguientes líneas dentro del archivo /etc/profile.

```
PATH=$PATH:/usr/local/jre/bin
export $PATH
```

- **Parámetros adicionales**

Ingresar como usuario root

```
# su – root
```

Cambiarse al directorio /proc/sys/kernel

```
# cd /proc/sys/kernel
```

Cambiar valores de semáforos.

```
# cat sem
# echo 100 32000 100 100 > sem
```

Lectura y cambio de valor de parámetros de memoria compartida.

```
# cat shm_param
# echo 2147483648 > shmmax
# echo 4096 > shmmni
# echo 2097152 > shmall
```

Cambio de parámetros de manejadores de archivos.

```
# echo 65536 > /proc/sys/fs/file-max
# ulimit -n 65536
```

Cambio de parámetros de sockets.

```
# echo 1024 65000 > /proc/sys/net/ipv4/ip_local_port_range
```

Cambio de parámetros de límites para procesos.

```
# ulimit -u 16384
```

- **Creación de Grupos y Cuentas**

Como en todas las Bases Oracle, se requiere de un propietario que tradicionalmente es nombrado oracle.

```
# groupadd dba
# groupadd oinstall
# groupadd oper
# useradd –c DBA –g oinstall –G dba oracle
```

Preparamos los puntos de montaje para la instalación de Oracle.

```
# mkdir /u01 /u02
# chown oracle.dba /u01 /u02
# chmod 755 /u01 /u02
```

Antes de la instalación de Oracle, es necesario que la variable umask sea seteada a 0022.

```
# umask 0022
```

Antes de empezar con la instalación, ingresar como usuario oracle y estar seguro de que las variables PATH y DISPLAY tengan los valores correctos.

```
# xhost +localhost
# export DISPLAY=localhost:0.0
```

Para observar las variables de ambiente se usa el comando env.

Instalación de Oracle

Montar el CD1 de instalación

```
# mount /dev/cdrom /mnt/cdrom
```

Cuando se está instalando Oracle, el instalador indica que se debe insertar el CD2 y CD3, si el CD no se desmonta automáticamente, entonces ejecutar en un terminal lo siguiente.

```
# cd /mnt
# eject
```

Para comenzar la instalación, insertar el CD1 de Oracle.

```
# cd /mnt/cdrom
# ./runInstaller &
```

Luego de ejecutar ”runInstaller”, aparecerá la ventana de bienvenida, en la cual uno puede elegir install or uninstall products.

Hacer click en Next para continuar

Aparecerá una nueva ventana de diálogo (Inventory Location)

Hacer click en OK si está de acuerdo con al ruta del directorio inventory

Luego nos vamos a la pantalla UNIX Group Name, la cual nos pregunta nuestro grupo de Instalación Oracle, por default es oinstall.

Escribir oinstall

Click Next.

Si todo va bien, aparecerá una ventana de diálogo con unas instrucciones de ejecutar el script /tmp/orainstRoot.sh en un terminal.

Para ejecutar:

./tmp/orainstRoot.sh

Después aparecerá un mensaje.

Creating Oracle Inventory pointer file (/etc/oraInst.loc).

Changing groupname of /u01/app/oracle/oraInventory to oinstall.

Ahora regresar a la ventana de diálogo que tenía las instrucciones de instalación.

Click Next para continuar

Aparecerá la pantalla de localización de archivos (File Locations), en la cual se mostrará la fuente y el destino de los archivos de instalación. Si todas las variables están bien no se debería cambiar nada.

Click Next para continuar

Mostrará la pantalla de productos disponibles (Available Products).

Click Next para continuar

Pantalla Installation Types

Click Standard Edition

Click Next para continuar

Se mostrará la pantalla Database Configuration.

Click General Purpose

Click Next para continuar

Pantalla para identificación (Database Identification). Aquí se tiene que ingresar el nombre de la base global (global database).

ORTD.zeus

En este caso zeus es el nombre del host, se puede usar cualquier nombre.

Aparecerá la pantalla que nos pide ingresar la ruta donde los archivos físicos de la base se localizarán, es preferible que estos datos se encuentren en discos separados pero por falta de recursos hemos utilizado un directorio o partición previamente creada.

/u02/oradata

Click Next para continuar

Pantalla, donde se tendrá un árbol de todos los componentes que serán instalados en el Servidor. Si no estamos de acuerdo con algún componente se puede elegir retroceder (Back) o si está todo bien entonces

Click install

Esto durará un largo tiempo hasta que instale todos los componentes, para esto requerimos intercambiar entre los CD1, CD2, CD3.

Durante la instalación, pueden surgir ciertos errores o inconvenientes que no nos permiten terminar la instalación correctamente.

El primer inconveniente que surgió fue que no avanzaba la instalación y parecía que se quedó ciclada leyendo un archivo. Este error ocurre cuando no se ha exportado la variable LD_ASSUME_KERNEL con el valor 2.4.1

Otro inconveniente es cuando estaba en el 79% de la instalación:

```
Error in invoking target install of makefile
/u01/app/oracle/product/9.2.0.1.0/ctx/lib/ins_ctx.mk
```

Para corregir este error editar el archivo:

gedit $ORACLE_HOME/ctx/lib/env_ctx.mk

Ir a la línea que dice “INSO_LINK =” y añadir “$(LDLIBFLAG)dl” todo el párrafo con el texto cambiado quedaría.

```
INSO_LINK = -L$(CTXLIB) $(LDLIBFLAG)m $(LDLIBFLAG)dl $(LDLIBFLAG)sc_ca
$(LDLIBFLAG)sc_fa   $(LDLIBFLAG)sc_ex   $(LDLIBFLAG)sc_da   $(LDLIBFLAG)sc_ut
$(LDLIBFLAG)sc_ch   $(LDLIBFLAG)sc_fi   $(LLIBCTXHX)   $(LDLIBFLAG)c   -Wl,-
rpath,$(CTXHOME)lib $(CORELIBS) $(COMPEOBJS)
```

Grabar los cambios del archivo y regresar al cuadro de diálogo donde salía el error.

Click Retry

Antes de finalizar la instalación, aparecerá un cuadro de diálogo el cual nos dice que ejecutemos en un terminal un script llamado root.sh, que debe ser ejecutado como usuario root. Este script puede demorarse un poco.

$ORACLE_HOME/root.sh

Pantalla de herramientas de configuración (Configuration Tools), trata de ejecutar tres tareas y reporta fallas en todas ellas, para resolver este problema, nos ubicamos en el directorio $ORACLE_HOME y ahí hay un link JRE que se enlaza con el directorio JRE de Oracle, borremos ese link y enlacémonos con el JRE que nosotros previamente instalamos.

cd $ORACLE_HOME

rm –f JRE

ln –s /usr/local/jre/ JRE

Después de ejecutar estos comandos, regresar a la pantalla de herramientas de configuración.

Click Retry

Otros problemas menores pero que igual hay que tener en consideración son los siguientes:

- JRE version compatibility
- Missing database configuration file
- Legato Single Server Version installation failure

Este problema existe porque RedHat 8 y 9 vienen con versión gcc 3.2 x, pero algunas viejas versiones de JRE son compiladas con versión 2.9 x. La solución es bajarse la última versión de JRE y crear un enlace simbólico.

```
ln -s $JRE_DIR/bin/java $JRE_DIR/bin/jre
```

Donde $JRE_DIR es el directorio del JRE actualizado.

Un error de mayor magnitud, es que el archivo de configuración de la base está perdido, este archivo init(ORACLE_SID).ora debería estar creado por default en $ORACLE_HOME/dbs/directory.

Hay dos formas de arreglar este error son:

Primera forma:

```
 # cd $ORACLE_HOME/dbs/
# ln -s spfile{ORACLE_SID}.ora init{ORACLE_SID}.ora
```

La solución que implementamos es la segunda, la cual es crear un archivo llamado init(ORACLE_SID).ora, que es una copia de init.ora y colocarlo en el directorio $ORACLE_HOME/dbs/.

Aquí encontramos al ruta donde está el archivo init.ora.

```
# find $ORACLE_HOME -name init.ora
```

Copiamos el archivo init.ora al directorio $ORACLE_HOME/dbs.

```
# cd /ruta_encontrada_del_archivo_init.ora
```

cp init.ora $ORACLE_HOME/dbs/initORTD.ora

Fin de la Instalación.

Post - Instalación

Editar el archivo oratab, este es un paso bien importante, en el cual se puede elegir si se quiere que la base se ejecute automáticamente. Para esto hay que colocar la letra Y al final de la línea que comienza con ORTD. La línea cambiada quedaría así.

ORTD:/u01/app/oracle/product/9.2.0.1.0:Y

- **Creación del archivo de Booteo**

```
#!/bin/bash
#
# Script de Booteo para Oracle.
# Run-level Startup script for the Oracle Instance and Listener
#
# chkconfig: 345 91 19
# description: Startup/Shutdown Oracle listener and instance

ORA_HOME="/u01/app/oracle/product/9.2.0.1.0"
ORA_OWNR="oracle"
```

```
# if the executables do not exist -- display error

if [ ! -f $ORA_HOME/bin/dbstart -o ! -d $ORA_HOME ]
then
        echo "Oracle startup: cannot start"
        exit 1
fi

# depending on parameter -- startup, shutdown, restart
# of the instance and listener or usage display

case "$1" in
    start)
        # Oracle listener and instance startup
        echo -n "Starting Oracle: "
        su - $ORA_OWNR -c "$ORA_HOME/bin/lsnrctl start"
        su - $ORA_OWNR -c $ORA_HOME/bin/dbstart
        touch /var/lock/subsys/oracle
        echo "OK"
        ;;
    stop)
```

```
        # Oracle listener and instance shutdown
        echo -n "Shutdown Oracle: "
        su - $ORA_OWNR -c "$ORA_HOME/bin/lsnrctl stop"
        su - $ORA_OWNR -c $ORA_HOME/bin/dbshut
        rm -f /var/lock/subsys/oracle
        echo "OK"
        ;;
    reload|restart)
        $0 stop
        $0 start
        ;;
    *)
        echo "Usage: $0 start|stop|restart|reload"
        exit 1
esac
exit 0
```

Copiar y pegar el script anterior en un archivo llamado IniciandoOracle y moverlo al directorio /etc/rc.d/init.d/ y cambiarle el propietario y los atributos, para que sean los mismos a los demás archivos dentro del directorio /etc/rc.d/init.d/

Luego de cambiar el propietario y los atributos del archivo ejecutar:

chkconfig -add IniciandoOracle

Ahora se puede reiniciar la máquina y se observará como se ejecuta Oracle automáticamente.

Servidor de correo

Obtener el siguiente archivo:

- *sendmail.8.12.11.tar.gz*

Crear Certificados para TLS

mkdir -p /etc/mail/certs

cd /etc/mail/certs

openssl req -new -x509 -keyout cakey.pem -out cacert.pem -days 365

openssl req -nodes -new -x509 -keyout sendmail.pem -out sendmail.pem -days 365

openssl x509 -noout -text -in sendmail.pem

chmod 600 ./sendmail.pem

Instalar Sendmail

Para instalar sendmail ejecutamos los siguientes comandos:

```
# cd /usr/local
# tar xvfz sendmail.8.12.11.tar.gz
# cd sendmail-8.12.11/devtools/Site/
```

Crear el archivo *site.config.m4* (en *devtools/Site/*):

```
# SASL2 (smtp authentication)
APPENDDEF(`confENVDEF', `-DSASL=2')
APPENDDEF(`conf_sendmail_LIBS', `-lsasl2')
#
# STARTTLS (smtp + tls/ssl)
APPENDDEF(`conf_sendmail_ENVDEF', `-DSTARTTLS')
APPENDDEF(`conf_sendmail_ENVDEF', `-D_FFR_SMTP_SSL')
APPENDDEF(`conf_sendmail_LIBS', `-lssl -lcrypto -L/usr/share/ssl/lib')
```

Con los siguientes comandos crear los directorios /usr/man, /usr/man/man1 /usr/man/mn8.

```
# mkdir -p /usr/man
# mkdir -p /usr/man/man1
# mkdir -p /usr/man/man8
```

También ejecutar los siguientes comandos, para luego poder configurar la autenticación con la librería sasl2:

```
#cp -pfr /usr/local/lib/sasl2 /usr/lib/sasl2
# echo /usr/lib/sasl2 >> /etc/ld.so.conf
# ldconfig
# ln -s /usr/local/ssl/include/openssl /usr/include/openssl
```

Compilamos sendmail:

```
# cd /usr/local/sendmail-8.12.11/
# useradd smmsp
# groupadd smmsp
# ln -s /usr/kerberos/include/* /usr/include
# sh Build –c
# sh Build install
```

Creamos el archivo sendmail.cf:

```
# cd cf/cf/
```

Creamos el archivo sendmail.mc con el siguiente contenido:

```
# gedit sendmail.mc &
```

```
dnl ### do SMTPAUTH
define(`confAUTH_MECHANISMS', `LOGIN PLAIN DIGEST-MD5 CRAM-
MD5')dnl
```

```
TRUST_AUTH_MECH(`LOGIN PLAIN DIGEST-MD5 CRAM-MD5')dnl


dnl ### do STARTTLS
define(`confCACERT_PATH', `/etc/mail/certs')dnl
define(`confCACERT', `/etc/mail/certs/cacert.pem')dnl
define(`confSERVER_CERT', `/etc/mail/certs/sendmail.pem')dnl
define(`confSERVER_KEY', `/etc/mail/certs/sendmail.pem')dnl
define(`confCLIENT_CERT', `/etc/mail/certs/sendmail.pem')dnl
define(`confCLIENT_KEY', `/etc/mail/certs/sendmail.pem')dnl
DAEMON_OPTIONS(`Family=inet, Port=465, Name=MTA-SSL, M=s')dnl


dnl ###
define(`confDEF_CHAR_SET', `iso-8859-1')dnl
define(`confMAX_MESSAGE_SIZE', `15000000')dnl Denial of Service Attacks
define(`confMAX_DAEMON_CHILDREN', `30')dnl Denial of Service Attacks
define(`confCONNECTION_RATE_THROTTLE', `2')dnl Denial of Service Attacks
define(`confMAXRCPTSPERMESSAGE', `50')dnl Denial of service Attacks
define(`confSINGLE_LINE_FROM_HEADER', `True')dnl
define(`confSMTP_LOGIN_MSG', `$j')dnl
define(`confDONT_PROBE_INTERFACES', `True')dnl
define(`confTO_INITIAL', `6m')dnl
define(`confTO_CONNECT', `20s')dnl
define(`confTO_HELO', `5m')dnl
define(`confTO_HOSTSTATUS', `2m')dnl
```

```
define(`confTO_DATAINIT', `6m')dnl
define(`confTO_DATABLOCK', `35m')dnl
define(`confTO_DATAFINAL', `35m')dnl
define(`confDIAL_DELAY', `20s')dnl
define(`confNO_RCPT_ACTION', `add-apparently-to')dnl
define(`confALIAS_WAIT', `0')dnl
define(`confMAX_HOP', `35')dnl
define(`confQUEUE_LA', `5')dnl
define(`confREFUSE_LA', `12')dnl
define(`confSEPARATE_PROC', `False')dnl
define(`confCON_EXPENSIVE', `true')dnl
define(`confWORK_RECIPIENT_FACTOR', `1000')dnl
define(`confWORK_TIME_FACTOR', `3000')dnl
define(`confQUEUE_SORT_ORDER', `Time')dnl
define(`confPRIVACY_FLAGS',
`authwarnings,goaway,restrictmailq,restrictqrun,needmailhelo')dnl
OSTYPE(linux)dnl
FEATURE(`delay_checks')dnl
FEATURE(`generics_entire_domain')dnl
FEATURE(`local_procmail')dnl
FEATURE(`masquerade_envelope')dnl
FEATURE(`nouucp',`reject')dnl
FEATURE(`redirect')dnl
FEATURE(`relay_entire_domain')dnl
```

```
FEATURE(`use_cw_file')dnl
FEATURE(`virtuser_entire_domain')dnl

FEATURE(dnsbl,`blackholes.mail-abuse.org',
` Mail from $&{client_addr} rejected; see http://mail-abuse.org/cgi-bin/lookup?$&
{client_addr}')dnl
FEATURE(dnsbl,`dialups.mail-abuse.org',
` Mail from dial-up rejected; see http://mail-abuse.org/dul/enduser.htm')dnl

FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable')dnl
FEATURE(access_db)dnl
FEATURE(lookupdotdomain)dnl
FEATURE(`blacklist_recipients')dnl
FEATURE(`no_default_msa')dnl
DAEMON_OPTIONS(`Port=smtp, Name=MTA')dnl
MAILER(local)dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
```

Para crear el archivo */etc/mail/sendmail.cf* ejecutar los siguientes comandos:

```
# sh Build sendmail.cf
# cp sendmail.cf /etc/mail/sendmail.cf
```

Finalmente creamos los siguientes archivos:

```
# cd /etc/mail/

# touch /etc/mail/local-host-names

# touch /etc/mail/virtusertable

# /usr/sbin/makemap hash virtusertable < virtusertable

# mkdir -p /var/spool/mqueue

# chmod 700 /var/spool/mqueue

# chown root:root /var/spool/mqueue

# chown root:root /etc/mail/sendmail.cf

# chmod 444 /etc/mail/sendmail.cf

# touch /etc/mail/submit.cf

# chown root:root /etc/mail/submit.cf

# chmod 444 /etc/mail/submit.cf

# touch /etc/mail/aliases

# newaliases

# touch /etc/mail/access

# /usr/sbin/makemap hash access < access
```

Creamos un script para iniciar sendmail (este debe ser copiado a */etc/init.d/sendmail):*

```
#! /bin/sh
```

```
case "$1" in
   start)
      echo "Initializing SMTP port. (sendmail)"
      /usr/sbin/sendmail -bd -q1h
      ;;
   stop)
      echo "Shutting down SMTP port:"
      killall /usr/sbin/sendmail
      ;;
   restart|reload)
      $0 stop  &&  $0 start
      ;;
   *)
      echo "Usage: $0 {start|stop|restart|reload}"
      exit 1
esac
exit 0
```

Ahora cambiamos los permisos de lectura, escritura y ejecución del archivo recientemente creado:

```
# chmod 755 /etc/init.d/sendmail
```

Para ejecutar sendmail automáticamente al inicio del sistema, creamos los siguientes enlaces simbólicos:

```
# ln -s /etc/init.d/sendmail /etc/rc2.d/S20sendmail
# ln -s /etc/init.d/sendmail /etc/rc3.d/S20sendmail
# ln -s /etc/init.d/sendmail /etc/rc4.d/S20sendmail
# ln -s /etc/init.d/sendmail /etc/rc5.d/S20sendmail
# ln -s /etc/init.d/sendmail /etc/rc0.d/K20sendmail
# ln -s /etc/init.d/sendmail /etc/rc1.d/K20sendmail
# ln -s /etc/init.d/sendmail /etc/rc6.d/K20sendmail
```

Configurar Saslauthd

Ahora vamos a crear un script para iniciar saslauthd:

```
#!/bin/sh -e

NAME=saslauthd
DAEMON="/usr/sbin/${NAME}"
DESC="SASL Authentication Daemon"
DEFAULTS=/etc/default/saslauthd

test -f "${DAEMON}" || exit 0
```

```
# Source defaults file; edit that file to configure this script.
if [ -e "${DEFAULTS}" ]; then
    . "${DEFAULTS}"
fi

# If we're not to start the daemon, simply exit
if [ "${START}" != "yes" ]; then
    exit 0
fi

# If we have no mechanisms defined
if [ "x${MECHANISMS}" = "x" ]; then
    echo "You need to configure ${DEFAULTS} with mechanisms to be
used"
    exit 0
fi

# Add our mechanisms with the necessary flag
for i in ${MECHANISMS}; do
    PARAMS="${PARAMS} -a ${i}"
done
```

```
# Consider our options
case "${1}" in
  start)
        echo -n "Starting ${DESC}: "
        ln -fs /var/spool/postfix/var/run/${NAME} /var/run/${NAME}
        ${DAEMON} ${PARAMS}
        echo "${NAME}."
        ;;
  stop)
        echo -n "Stopping ${DESC}: "
        PROCS=`ps aux | grep -iw '/usr/sbin/saslauthd' | grep -v 'grep' |awk
'{print $2}' | tr '\n' ' '`
        if [ "x${PROCS}" != "x" ]; then
          kill -15 ${PROCS} &> /dev/null
        fi
        echo "${NAME}."
        ;;
  restart|force-reload)
        $0 stop
        sleep 1
        $0 start
        echo "${NAME}."
```

```
        ;;
  *)
        echo "Usage: /etc/init.d/${NAME} {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac


exit 0
```

Ahora cambiamos los permisos de lectura, escritura y ejecución del archivo recien creado:

```
# chmod 755 /etc/init.d/saslauthd
```

Para ejecutar saslauthd automáticamente al inicio del sistema, creamos los siguientes enlaces simbólicos:

```
# ln -s /etc/init.d/saslauthd /etc/rc2.d/S20saslauthd
# ln -s /etc/init.d/saslauthd /etc/rc3.d/S20saslauthd
# ln -s /etc/init.d/saslauthd /etc/rc4.d/S20saslauthd
# ln -s /etc/init.d/saslauthd /etc/rc5.d/S20saslauthd
# ln -s /etc/init.d/saslauthd /etc/rc0.d/K20saslauthd
# ln -s /etc/init.d/saslauthd /etc/rc1.d/K20saslauthd
```

ln -s /etc/init.d/saslauthd /etc/rc6.d/K20saslauthd

Si *saslauthd* está localizado en */usr/local/sbin* en lugar de */usr/sbin,* crear un enlace simbólico:

ln -s /usr/local/sbin/saslauthd /usr/sbin/saslauthd

Ahora creamos /etc/default/saslauthd:

```
# This needs to be uncommented before saslauthd will be run automatically
START=yes

# You must specify the authentication mechanisms you wish to use.
# This defaults to "pam" for PAM support, but may also include
# "shadow" or "sasldb"
MECHANISMS=shadow
```

Ahora iniciamos saslauthd y sendmail:

/etc/init.d/saslauthd start

/etc/init.d/sendmail start

Probar la configuración

Verificar que sendmail fue compilado con las opciones correctas:

/usr/sbin/sendmail -d0.1 -bv root

Se debería ver que sendmail fue compilado con SASLv2 y STARTTLS:

```
server1:~# /usr/sbin/sendmail -d0.1 -bv root
Version 8.12.11
 Compiled with: DNSMAP LOG MATCHGECOS MIME7TO8 MIME8TO7 NAMED_BIND
                NETINET NETUNIX NEWDB PIPELINING SASLv2 SCANF STARTTLS USERDB
                XDEBUG

============ SYSTEM IDENTITY (after readcf) ============
      (short domain name) $w = server1
  (canonical domain name) $j = server1.example.com
         (subdomain name) $m = example.com
              (node name) $k = server1.example.com
=======================================================

root... deliverable: mailer local, user root
```

Para ver si SMTP-AUTH y TLS trabajan correctamente, ejecutar el siguiente comando:

telnet localhost 25

Después de establecer la conexión, ejecutar el siguiente comando:

ehlo localhost

Si se ven las líneas:

250-STARTTLS

y

250-AUTH

Todo está bien.

```
server1:~# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 server1.example.com ESMTP
ehlo localhost
250-server1.example.com Hello localhost [127.0.0.1], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 15000000
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
```

250-AUTH

250-STARTTLS

Ahora escribir:

quit

Para retornar al shell del sistema.

Servidor de nombres de dominio

El código fuente puede ser obtenido en la dirección www.isc.org, que es la página oficial de Bind.

Configurando e instalando el fuente

Después de haber bajado el fuente de la dirección anterior, tenemos que descomprimirlo para poder configurar e instalar el paquete en nuestra máquina que será el servidor DNS

Descomprimimos el archivo en el directorio que deseamos

```
# tar -xvzf bind-9.2.3.tar.gz
```

Al descomprimir, se crea una carpeta con el mismo nombre del paquete.

Nos ubicamos en esa carpeta

```
# cd bind-9.2.3
```

Configuramos el paquete y establecemos que la carpeta de instalación va a ser /usr/local.

```
# ./configure --prefix=/usr/local --disable-ipv6 --disable-threads
```

Y luego la instalación

```
# make
# make install
```

Construyendo y configurando chroot

Los pasos iniciales para configurar el chroot son:

Crear usuario named y grupo named:

```
# groupadd named
# useradd -g named -d /chroot/named -s /bin/true named
# passwd -l named     "lock" the account
```

Remover todo lo relacionado con el usuario named en el directorio /chroot

```
# rm -rf /chroot/named
```

Recrear el top level jail directory

```
# mkdir -p /chroot/named
# cd /chroot/named
```

Crear jerarquía:

```
# mkdir dev
# mkdir etc
# mkdir logs
# mkdir -p var/run
# mkdir -p conf/secondaries
```

Crear los dispositivos, pero especificando el mayor y el menor número de dispositivo "ls -lL /dev/zero /dev/null /dev/random"

```
# mknod dev/null c 1 3
# mknod dev/zero c 1 5
# mknod dev/random c 1 8
```

Copiar el archivo de zona:

```
# cp /etc/localtime etc
```

Construyendo archivos de configuración

Comenzamos con el named.conf que se encuentra en el directorio etc de chroot, osea en /chroot/named/etc/named.conf

```
options {
        directory       "/conf";
        pid-file        "/var/run/named.pid";
        statistics-file "/var/run/named.stats";
        dump-file       "/var/run/named.db";

        # hide our "real" version number
        version         "[secured]";
};

# The root nameservers
zone "." {
        type   hint;
        file   "db.rootcache";
};

# localhost - forward zone
zone      "localhost" {
```

```
        type    master;
        file   "db.localhost";
        notify  no;
};

# localhost - inverse zone
zone    "0.0.127.in-addr.arpa" {
        type   master;
        file   "db.127.0.0";
        notify no;
};
```

Creamos un link simbólico:

```
# ln -s /chroot/named/etc/named.conf /etc/named.conf
```

Note que el directorio clave dice /conf y no dice /chroot/named/conf, esto es sobreentendido. Cuando ejecutemos el nameserver dentro de chroot todos los directorios se refieren dentro de /chroot/named.

Este archivo named.conf se está refiriendo a tres archivos db.rootcache, db.localhost y db.127.0.0

El archivo que vamos a generar es el db.rootcache que es un archivo donde están los servidores a los que nos vamos a conectar a través de Internet para poder efectuar el funcionamiento como un DNS.

Para crearlo necesitamos ejecutar:

```
# dig @a.root-servers.net . ns > /chroot/named/conf/db.rootcache
```

Observe que el archivo se lo crea en el directorio /chroot/named/conf/

Para generar los otros dos archivos db.localhost y db.127.0.0 solo tenemos que copiar y pegar lo siguiente en el directorio /chroot/named/conf/:

En el archivo db.localhost

```
;
; db.localhost
;
$TTL    86400

@       IN SOA   @ root (
                 42             ; serial (d. adams)
                 3H             ; refresh
                 15M            ; retry
                 1W             ; expiry
                 1D )           ; minimum
```

```
        IN NS        @
        IN A         127.0.0.1
```

En el archivo db.localhost

```
;
; db.127.0.0
;
$TTL    86400
@       IN      SOA     localhost. root.localhost.  (
                        1 ; Serial
                        28800      ; Refresh
                        14400      ; Retry
                        3600000    ; Expire
                        86400 )    ; Minimum
        IN      NS      localhost.
1       IN      PTR     localhost.
```

Verificando permisos

Creamos un archivo que es requerido dentro de chroot. Lo ubicamos dentro de /chroot y lo llamamos named.perms, o sea el archivo se encontrará en /chroot/named.perms

Configurar el propietario y los permisos en el directorio named:

```
cd /chroot/named
```

Por defecto, el usuario root es el propietario de todo, y solamente el tiene permisos de escritura, pero los directorios tiene que tener permisos de ejecución también.

```
chown -R root.named .

find . -type f -print | xargs chmod u=rw,og=r          # archivos regulares
find . -type d -print | xargs chmod u=rwx,og=rx        # directories
```

Los archivos named.conf y rndc.conf deben proteger sus claves:

```
chmod o= etc/*.conf
```

El directorio *"secondaries"* es donde se guardan archivos desde el servidor de nombres maestro, y *named* necesita ser capaz de actualizar estos archivos y crear unos nuevos.

```
touch conf/secondaries/.empty   # placeholder
find conf/secondaries/ -type f -print | xargs chown named.named
find conf/secondaries/ -type f -print | xargs chmod ug=r,o=
chown root.named conf/secondaries/
chmod ug=rwx,o=  conf/secondaries/
```

Asignamos el propietario y grupo al que pertenece el directorio /var

```
chown root.root  var/
```

Asignamos los permisos de lectura, escritura y ejecución del directorio /var

```
chown chmod u=rwx,og=x var/
```

Los pasos anteriores aplicamos para el directorio var/run

```
chown root.named  var/run/
chmod ug=rwx,o=rx var/run/
```

El usuario named debe ser capaz de crear archivos de registro, para esto asignamos a named como propietario del directorio /logs

```
chown root.named  logs/
chmod ug=rwx,o=rx logs/
```

Todos los comandos anteriores deben ir en el archivo named.perms.

Este archivo se lo ejecuta con los siguientes comandos:

```
# sh -x /chroot/named.perms
```

Al ejecutar esta sentencia obtendremos una salida parecida a la siguiente:

```
+ cd /chroot/named
+ chown -R root.named .
```

```
+ find . -type f -print
+ xargs chmod u=rw,og=r
+ find . -type d -print
+ xargs chmod u=rwx,og=rx
+ chmod o= etc/named.conf etc/rndc.conf
+ touch conf/secondaries/.empty
+ find conf/secondaries/ -type f -print
+ xargs chown named.named
+ find conf/secondaries/ -type f -print
+ xargs chmod ug=r,o=
+ chown root.named conf/secondaries/
+ chmod ug=rwx,o= conf/secondaries/
+ chown root.root var/
+ chmod u=rwx,og=x var/
+ chown root.named var/run/
+ chmod ug=rwx,o=rx var/run/
```

Ejecutando el DNS

Ahora se puede levantar el dns con el comando # rndc Stara

Nosotros hemos creado un archivo que nos permitirá iniciarlo con el comando # named.start. Este archivo lo llamamos named.start, lo hemos creado en /chroot/named.start y contiene las siguientes líneas:

```
#
# named.start
#
#       Note: the path given to the "-c" parameter is relative
#       to the jail's root, not the system root.
#
#       Add "-n2" if you have multiple CPUs
#
# usage: named [-c conffile] [-d debuglevel] [-f|-g] [-n number_of_cpus]
#               [-p port] [-s] [-t chrootdir] [-u username]
```

Antes de ejecutarlo tenemos que hacer lo siguiente:

```
# cd /chroot/named
```

Asegurarse de que el usuario named tenga permisos de escritura en el archivo named.run

```
# touch named.run
# chown named.named named.run
```

```
# chmod ug=rw,o=r   named.run

# PATH=/usr/local/sbin:$PATH named  \
        -t /chroot/named \
        -u named \
        -c /etc/named.conf
```

Luego lo hacemos ejecutable:

```
# chmod a+x /chroot/named.start
```

Lo iniciamos de la siguiente manera:

```
# sh /chroot/named.start
```

Nuestro DNS esta corriendo actualmente, para verificar si efectivamente está nuestro proceso en background podemos ejecutar:

```
# ps -fCnamed
```

Control con rndc

El comando rndc lee el archivo de configuración en /usr/local/etc/rndc.conf pero nosotros lo vamos ubicar en nuestra área chroot. Por lo tanto vamos a crear un rndc.conf en la dirección /chroot/named/etc/rndc.conf

```
#
# /chroot/named/etc/rndc.conf
#

options {
        default-server  127.0.0.1;
        default-key     "rndckey";
};

server 127.0.0.1 {
        key     "rndckey";
};

key "rndckey" {
        algorithm       "hmac-md5";
        secret          "secret key here";
};
```

rndckey es solo un nombre, uno puede colocar cualquiera. En la parte que dice *“secret key here”* se va a ubicar una clave que la vamos a generar de la siguiente manera:

```
# cd /chroot/named/etc
# /usr/local/sbin/dnssec-keygen -a HMAC-MD5 -b 256 -n HOST rndc
Krndc.+157+13856
# cat Krndc.+157+13856.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: hU9utBAdP6/dVKKfxOlv0bPOTnAd4A1qosMbs/dwVJI=
```

Por lo tanto nos quedaría de esta forma el archivo rndc.conf

```
#
# /chroot/named/etc/rndc.conf
#

options {
        default-server  127.0.0.1;
        default-key     "rndckey";
};

server 127.0.0.1 {
        key     "rndckey";
};
```

```
key "rndckey" {
        algorithm       "hmac-md5";
        secret          "hU9utBAdP6/dVKKfxOlv0bPOTnAd4A1qosMbs/dwVJI=";
};
```

Borramos el archivo de clave que se nos creó

```
# rm Krndc.+157+13856.*          after key has been saved
```

Crear un link simbólico para que el comando rndc sirva:

```
# ln -s /chroot/named/etc/rndc.conf /usr/local/etc/rndc.conf
# ln -s /chroot/named/etc/rndc.conf /etc/rndc.conf
```

Ahora el servidor de nombres (DNS), debe ser configurado para que escuche en el canal de control y use la clave en particular, por lo tanto añadimos estas líneas al principio del archivo /chroot/named/conf/named.conf

```
controls {
        inet 127.0.0.1 allow { 127.0.0.1; } keys { rndckey; };
};

key "rndckey" {
        algorithm       "hmac-md5";
        secret          "hU9utBAdP6/dVKKfxOlv0bPOTnAd4A1qosMbs/dwVJI=";
```

```
};
```

Tenemos que hacer que el DNS lea la nueva configuración, detenemos el DNS y luego lo iniciamos:

```
# rndc stop

#/chroot/named.start
```

O también podemos hacer lo siguiente:

```
# ps -fCnamed
UID        PID  PPID  C STIME TTY          TIME CMD
named    12527     1  0 12:42 ?        00:00:00 named -t /chroot/named {...}

# kill -1 12527        12527 is process ID
```

Ahora el DNS debe poder releer la configuración de archivo y empezar a escuchar en la interfaz de control. Para saber el estado del DNS utilizamos el siguiente comando:

```
# /usr/local/sbin/rndc status
number of zones: 2
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
```

Hasta aquí hemos configurado nuestro servidor DNS de solo cacheo, o sea que podemos navegar agregando nuestra IP de la máquina como servidor DNS pero aún no tenemos creado un dominio para nuestra red.

Crear Nuestro Propio Dominio

Ahora vamos a explicar como definir un dominio propio. Crearemos el dominio linux.bogus y definiremos máquinas en él. Usamos un nombre de dominio totalmente falso para estar seguro de que no molestamos a nadie de fuera.

Ahora con esta configuración vamos a establecer que un nombre de dominio, en este caso ns.linux.bogus, sea reconocido como una IP válida.

Una cosa más antes de empezar: No se permiten todos los caracteres en nombres de máquina. Estamos restringidos a los caracteres del alfabeto inglés: a-z, números 0-9 y el caracter '-' (guión). Es recomendado utilizar estos caracteres. Las mayúsculas y minúsculas son indistintas para el DNS, así pat.uio.no es idéntico a Pat.UiO.No.

Hemos comenzado esta parte con la siguiente línea en named.conf:

```
zone "0.0.127.in-addr.arpa" {
        type master;
```

```
        file "db.127.0.0";
            notify no;
};
```

Observar la ausencia de '.' al final de los nombres de dominio en este fichero. Esto dice que ahora vamos a definir la zona 0.0.127.in-addr.arpa, de la que somos servidor principal ("master") y que está definida en un fichero llamado db.127.0.0. Editar este archivo para que sea configurado con lo siguiente:

```
$TTL 86400
@       IN      SOA  ns.linux.bogus. hostmaster.linux.bogus. (
                        1       ; Serie
                        8H      ; Refresco
                        2H      ; Reintento
                        4W      ; Expira
                        1D)     ; Minimo TTL
                    NS      ns.linux.bogus.
1                   PTR     localhost.
```

Observar el '.' al final de los nombres de dominio completo, en contraste con el archivo named.conf anterior. A algunas personas les gusta iniciar cada zona del archivo con una directiva $ORIGIN, pero esto es superfluo. El origen

(lugar de la jerarquía DNS a donde pertenece) de un fichero de zona se especifica en la sección zona del archivo named.conf; en este caso es 0.0.127.in-addr.arpa.

Este "fichero de zona" contiene tres registros de recursos (RRs): Un RR SOA, Un RR NS y un RR PTR. SOA, es una abreviatura de Start Of Authority. La '@' es una notación especial que simboliza el origen, y como la columna dominio para este archivo indica 0.0.127.in-addr.arpa. La primera línea realmente significa:

0.0.127.in-addr.arpa. IN SOA ...

NS es el RR Name Server (Servidor de Nombres). No hay '@' al comienzo de esta línea; es implícita ya que la línea previa comenzaba con '@'. Eso ahorra algo de tecleo. Así la línea NS se podría haber escrito también como:

0.0.127.in-addr.arpa. IN NS ns.linux.bogus

Esto le indica al DNS qué máquina es el servidor de nombres del dominio 0.0.127.in-addr.arpa, este es ns.linux.bogus. 'ns' es un nombre habitual para servidores de nombres, pero como con los servidores web que habitualmente se llaman www.*loquesea* el nombre puede ser cualquiera.

Y finalmente el registro PTR (Domain Name Pointer, Puntero de nombre de dominio), le dice que el host con dirección 1 (igual a 1.0.0.127.IN-ADDR.ARPA, esto es, 127.0.0.1) es el llamado localhost

El registro SOA, es el preámbulo de *todos* los archivos de zona y debe haber uno exactamente en cada archivo de zona,(pero tras la directiva $TTL). El registro SOA describe la zona, de dónde proviene (una máquina llamada linux.bogus), quién es el responsable de su contenido (hostmaster@linux. Bogus), debe poner su dirección de correo aquí, qué versión del archivo de zona es (Número de Serie, 1), y otras cosas que tienen que ver con el caché y los servidores secundarios DNS. Para el resto de los campos (Tasa de Refresco, Tasa de Reintento, Caducidad para secundario y Tiempo de Validez para Clientes), usar los valores que aparecen aquí para mayor seguridad. Antes de SOA viene una línea obligatoria, la línea $TTL 3D, ponerla en todos los ficheros de zona.

Ahora reiniciar el named (la orden es rndc stop y /chroot/named.start) y usar dig para examinar el trabajo. -x pregunta por resolución inversa:

```
$ dig -x 127.0.0.1

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30944
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL:
0

;; QUESTION SECTION:
```

```
;1.0.0.127.in-addr.arpa.                IN      PTR

;; ANSWER SECTION:
1.0.0.127.in-addr.arpa. 259200  IN      PTR     localhost.

;; AUTHORITY SECTION:
0.0.127.in-addr.arpa.   259200  IN      NS      ns.linux.bogus.

;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:02:39 2001
;; MSG SIZE  rcvd: 91
```

Así se gestiona para obtener localhost de 127.0.0.1, bien, ahora para nuestra tarea principal, el dominio linux.bogus, insertar una nueva sección 'zone' en named.conf:

```
zone "linux.bogus" {
        notify no;
        type master;
        file "db.linux.bogus";
};
```

Observar de nuevo la ausencia de '.' final en el nombre de dominio en el fichero named.conf.

En el fichero de zona linux.bogus pondremos datos totalmente ficticios:

```
;
```

```
; Fichero de zona para linux.bogus
;
; El fichero de zona completo
;
$TTL 86400
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151       ; serie, fecha de hoy + serie de hoy #
                        8H              ; refresco, segundos
                        2H              ; reintento, segundos
                        4W              ; expira, segundos
                        1D )            ; mínimo, segundos
;
                NS      ns              ; Dirección Inet del servidor de nombres
                MX      10 mail.linux.bogus     ; Relay de correo primario
                MX      20 mail.friend.bogus.   ; Relay de correo secundario
;
localhost       A       127.0.0.1
ns              A       192.168.196.2  ;esto nos ayuda a crear subdominios
mail            A       192.168.196.4  ;de linux.bogus
```

Deben observarse dos cosas sobre los registros SOA. ns.linux.bogus debe ser una máquina actual con un registro A. No es legal tener un registro

CNAME (la explicación del registro CNAME estará en las hojas posteriores) para la máquina mencionada en el registro SOA. Su nombre no necesita ser ns, podría ser cualquier nombre legal de máquina. A continuación, en hostmaster.linux.bogus deberá aparecer algo como hostmaster@linux.bogus; esto sería un alias de email, o una cuenta de correo, donde la(s) persona(s) que realizan el mantenimiento de DNS deberían leer con frecuencia el correo. Cualquier mail respecto del dominio será mandado a la dirección aquí indicada. El nombre no tiene por que ser hostmaster, puede ser cualquier dirección mail legal, pero la dirección hostmaster funcionará también.

Hay un nuevo tipo de RR en este archivo, el MX, o Mail eXchanger. Este indica al sistema de correo donde mandar el correo dirigido a alguien@linux.bogus, pudiendo ser también mail.linux.bogus o mail.friend.bogus, osea mandar el correo a la IP 192.168.196.4 que en este caso debería ser nuestro servidor de Correos. El número que precede a cada nombre de máquina es la prioridad del RR MX. El RR con el número más bajo (10), es aquel, al que el correo será enviado primero. Si este falla, puede ser mandado a otro con un número más alto, que será gestor secundario de correo, como mail.friend.bogus que tiene una prioridad 20 aquí.

Reiniciar named ejecutando rndc stop y luego /chroot/named.start. Examinar los resultados con dig.

La barra (\ ) significa que lo de abajo es continuación de la línea que contiene (\ )

```
$ dig any linux.bogus

; <<>> DiG 9.1.3 <<>> any linux.bogus
;; global options:  printcmd
;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55239
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 1, ADDITIONAL:
1

;; QUESTION SECTION:
;linux.bogus.            IN      ANY

;; ANSWER SECTION:
linux.bogus.        259200  IN      SOA     ns.linux.bogus. \
     hostmaster.linux.bogus. 199802151 28800 7200 2419200 86400
linux.bogus.        259200  IN      NS      ns.linux.bogus.
linux.bogus.        259200  IN      MX      20 mail.friend.bogus.
linux.bogus.        259200  IN      MX      10 mail.linux.bogus.linux.bogus.

;; AUTHORITY SECTION:
linux.bogus.        259200  IN      NS      ns.linux.bogus.

;; ADDITIONAL SECTION:
ns.linux.bogus.     259200  IN      A       192.168.196.2

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:06:45 2001
;; MSG SIZE  rcvd: 184


$ dig any linux.bogus

; <<>> DiG 9.1.3 <<>> any linux.bogus
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55239
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 1, ADDITIONAL:
1
```

```
;; QUESTION SECTION:
;linux.bogus.            IN      ANY

;; ANSWER SECTION:
linux.bogus.       259200  IN      SOA     ns.linux.bogus. \
     hostmaster.linux.bogus. 199802151 28800 7200 2419200 86400
linux.bogus.       259200  IN      NS      ns.linux.bogus.
linux.bogus.       259200  IN      MX      20 mail.friend.bogus.
linux.bogus.       259200  IN      MX      10 mail.linux.bogus.linux.bogus.

;; AUTHORITY SECTION:
linux.bogus.       259200  IN      NS      ns.linux.bogus.

;; ADDITIONAL SECTION:
ns.linux.bogus.    259200  IN      A       192.168.196.2

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:06:45 2001
;; MSG SIZE  rcvd: 184


$ dig any linux.bogus

; <<>> DiG 9.1.3 <<>> any linux.bogus
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55239
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 1, ADDITIONAL:
1

;; QUESTION SECTION:
;linux.bogus.            IN      ANY

;; ANSWER SECTION:
linux.bogus.       259200  IN      SOA     ns.linux.bogus. \
     hostmaster.linux.bogus. 199802151 28800 7200 2419200 86400
linux.bogus.       259200  IN      NS      ns.linux.bogus.
linux.bogus.       259200  IN      MX      20 mail.friend.bogus.
linux.bogus.       259200  IN      MX      10 mail.linux.bogus.linux.bogus.

;; AUTHORITY SECTION:
linux.bogus.       259200  IN      NS      ns.linux.bogus.
```

```
;; ADDITIONAL SECTION:
ns.linux.bogus.     259200  IN      A       192.168.196.2

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:06:45 2001
;; MSG SIZE  rcvd: 184
```

```
$ dig any linux.bogus

; <<>> DiG 9.1.3 <<>> any linux.bogus
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55239
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 1, ADDITIONAL:
1

;; QUESTION SECTION:
;linux.bogus.               IN      ANY

;; ANSWER SECTION:
linux.bogus.        259200  IN      SOA     ns.linux.bogus. \
     hostmaster.linux.bogus. 199802151 28800 7200 2419200 86400
linux.bogus.        259200  IN      NS      ns.linux.bogus.
linux.bogus.        259200  IN      MX      20 mail.friend.bogus.
linux.bogus.        259200  IN      MX      10 mail.linux.bogus.linux.bogus.

;; AUTHORITY SECTION:
linux.bogus.        259200  IN      NS      ns.linux.bogus.

;; ADDITIONAL SECTION:
ns.linux.bogus.     259200  IN      A       192.168.196.2

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:06:45 2001
;; MSG SIZE  rcvd: 184
```

Con un examen cuidadoso puede descubrir fallos. La línea

linux.bogus. 3D IN MX 10 mail.linux.bogus.linux.bogus.

está completamente equivocada. Debería ser

linux.bogus. 3D IN MX 10 mail.linux.bogus.

Deliberadamente se cometió este error para ir aprendiendo de él. Mirando en el fichero de zona encontramos la línea:

MX 10 mail.linux.bogus ; Relay de correo primario

Falta un punto. O tiene demasiados 'linux.bogus'. Si un nombre de máquina no termina en punto, en un fichero de zona, se le añade el origen al final ocasionando el doble linux.bogus.linux.bogus. Entonces, o bien

MX 10 mail.linux.bogus. ; Relay de correo primario

o

MX 10 mail ; Relay de correo primario

es correcto. Se prefiere la última forma, hay que teclear menos. Hay algunos expertos de BIND que no están de acuerdo con esto y otros que sí. En un fichero de zona, el dominio debería de escribirse bien finalizado con un '.' o no debería incluirse, en cuyo caso toma como valor predeterminado el origen.

No olvidar que en el fichero named.conf no se tiene que poner '.'s tras los nombres de dominio. Un '.' por ponerlo o por no ponerlo puede estropear todo.

Aquí está el nuevo fichero de zona, con alguna información extra también:

```
;
; Fichero de zona para linux.bogus
;
; Fichero de zona completo
;
$TTL 86400
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151       ; seriel, fecho de hoy + serie de hoy #
                        8H              ; refresco, segundos
                        2H              ; reintento, segundos
                        4W              ; expira, segundos
                        1D )            ; mínimo, segundos
;
                TXT     "Linux.Bogus, sus consutas DNS"
                NS      ns              ; Inet Address of name server
                NS      ns.friend.bogus.
                MX      10 mail         ; Relay primario de correo
```

```
            MX      20 mail.friend.bogus. ; Relay secundario de correo

localhost       A       127.0.0.1

gw              A       192.168.196.1
            TXT     "El router"

ns              A       192.168.196.2
            MX      10 mail
            MX      20 mail.friend.bogus.
www             CNAME   ns

donald          A       192.168.196.3
            MX      10 mail
            MX      20 mail.friend.bogus.
            HINFO   "i486"  "Linux 2.0"
            TXT     "DEK"

mail            A       192.168.196.4
            MX      10 mail
            MX      20 mail.friend.bogus.
```

```
ftp          A      192.168.196.5
             MX     10 mail
             MX     20 mail.friend.bogus.
```

El registro TXT es un texto en formato libre que se puede usar para cualquier cosa. CNAME, (Canonical NAME), es una forma de dar a cada máquina varios nombres. Por tanto www es un alias para ns.

El uso de registros CNAME es controvertido. Pero es más seguro seguir la norma de que los registros MX, CNAME y SOA, nunca se debe hacer referencia a un registro CNAME, sólo deben referirse a registros A, en consecuencia no es aconsejable tener

```
foobar       CNAME   www                ; NO!
```

lo correcto sería tener :

```
foobar       CNAME   ns                 ; Si!
```

También es mejor suponer que un CNAME, no es un nombre de máquina legal para direcciones de correo: webmaster@www.linux.bogus, es una dirección email ilegal dada en la configuración anterior. Son muy pocos los administradores de correo que recomienden esta regla, incluso si funciona. La forma de evitar esto es usar un registro A (y quizá algunos otros también, como un registro MX) en su lugar:

```
www          A      192.168.196.2
```

Algunos expertos de named recomiendan no usar CNAME. Por tanto se debe considerar el no utilizarlo. Pero la discusión de por qué o no, está más allá de los objetivos de este documento.

Pero como se puede ver, en este manual y en muchos sitios no siguen esta regla.

Cargar la nueva base de datos ejecutando rndc stop y luego /chroot/named.start, esto provoca la lectura de sus archivos de nuevo.

```
$ dig linux.bogus axfr

; <<>> DiG 9.1.3 <<>> linux.bogus axfr
;; global options:  printcmd
linux.bogus.                    259200    IN          SOA          ns.linux.bogus.
hostmaster.linux.bogus. 199802151 28800 7200 2419200 86400
linux.bogus.          259200  IN      NS      ns.linux.bogus.
linux.bogus.          259200  IN      MX      10 mail.linux.bogus.
linux.bogus.          259200  IN      MX      20 mail.friend.bogus.
donald.linux.bogus.     259200  IN      A       192.168.196.3
donald.linux.bogus.     259200  IN      MX      10 mail.linux.bogus.
donald.linux.bogus.     259200  IN      MX      20 mail.friend.bogus.
donald.linux.bogus.     259200  IN      TXT     "DEK"
ftp.linux.bogus.        259200  IN      A       192.168.196.5
ftp.linux.bogus.        259200  IN      MX      10 mail.linux.bogus.
ftp.linux.bogus.        259200  IN      MX      20 mail.friend.bogus.
gw.linux.bogus.         259200  IN      A       192.168.196.1
gw.linux.bogus.         259200  IN      TXT     "The router"
localhost.linux.bogus.  259200  IN      A       127.0.0.1
mail.linux.bogus.       259200  IN      A       192.168.196.4
mail.linux.bogus.       259200  IN      MX      10 mail.linux.bogus.
mail.linux.bogus.       259200  IN      MX      20 mail.friend.bogus.
```

```
ns.linux.bogus.        259200  IN      MX      10 mail.linux.bogus.
ns.linux.bogus.        259200  IN      MX      20 mail.friend.bogus.
ns.linux.bogus.        259200  IN      A       192.168.196.2
www.linux.bogus.       259200  IN      CNAME   ns.linux.bogus.
linux.bogus.                   259200    IN          SOA          ns.linux.bogus.
hostmaster.linux.bogus. 199802151 28800 7200 2419200 86400
;; Query time: 41 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:12:31 2001
;; XFR size: 23 records
```

Está bien. Como se ve, se parece bastante al propio archivo de zona.

Comprobamos que dice para www solo:

```
$ dig www.linux.bogus

; <<>> DiG 9.1.3 <<>> www.linux.bogus
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16633
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL:
0

;; QUESTION SECTION:
;www.linux.bogus.              IN      A

;; ANSWER SECTION:
www.linux.bogus.       259200  IN      CNAME   ns.linux.bogus.
ns.linux.bogus.        259200  IN      A       192.168.196.2

;; AUTHORITY SECTION:
linux.bogus.           259200  IN      NS      ns.linux.bogus.

;; Query time: 5 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:14:14 2001
;; MSG SIZE  rcvd: 80
```

En otras palabras, el nombre real de www.linux.bogus es ns.linux.bogus, y da parte de la información que se tiene de ns también.

La zona inversa

Ahora los programas pueden convertir los nombres de linux.bogus a direcciones a las que uno puede conectarse. Pero también se necesita una zona inversa, una para hacer que DNS sea capaz de convertir de direcciones a nombres. Ese nombre lo utilizan distintos tipos de servidores: (FTP, IRC, WWW y otros), para decidir si quieren o no hablar con nosotros, y en caso afirmativo, qué prioridad se le debería dar. Para un acceso completo a los servicios de internet se necesita una zona inversa.

Poner esto en named.conf:

```
zone
 "196.168.192.in-addr.arpa" {
        notify no;
        type master;
        file "db.192.168.196";
};
```

Esto es exactamente como en 0.0.127.in-addr.arpa, y los contenidos son similares:

```
$TTL 86400
```

```
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151 ; Serial, todays date + todays serial
                        8H      ; Refresco
                        2H      ; Reintento
                        4W      ; Expira
                        1D)     ; Minimo TTL
                NS      ns.linux.bogus.

1               PTR     gw.linux.bogus.
2               PTR     ns.linux.bogus.
3               PTR     donald.linux.bogus.
4               PTR     mail.linux.bogus.
5               PTR     ftp.linux.bogus.
```

Ahora reiniciar el named (rndc stop y /chroot/named.start) y examinar el trabajo con dig de nuevo:

```
$ dig -x 192.168.196.4

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58451
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL:
1

;; QUESTION SECTION:
;4.196.168.192.in-addr.arpa.    IN      PTR
```

```
;; ANSWER SECTION:
4.196.168.192.in-addr.arpa. 259200 IN   PTR     mail.linux.bogus.

;; AUTHORITY SECTION:
196.168.192.in-addr.arpa. 259200 IN     NS      ns.linux.bogus.

;; ADDITIONAL SECTION:
ns.linux.bogus.         259200  IN      A       192.168.196.2

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:16:05 2001
;; MSG SIZE  rcvd: 107
```

```
$ dig 196.168.192.in-addr.arpa. AXFR

; <<>> DiG 9.1.3 <<>> 196.168.192.in-addr.arpa. AXFR
;; global options:  printcmd
196.168.192.in-addr.arpa. 259200 IN     SOA     ns.linux.bogus. \
        hostmaster.linux.bogus. 199802151 28800 7200 2419200 86400
196.168.192.in-addr.arpa. 259200 IN     NS      ns.linux.bogus.
1.196.168.192.in-addr.arpa. 259200 IN   PTR     gw.linux.bogus.
2.196.168.192.in-addr.arpa. 259200 IN   PTR     ns.linux.bogus.
3.196.168.192.in-addr.arpa. 259200 IN   PTR     donald.linux.bogus.
4.196.168.192.in-addr.arpa. 259200 IN   PTR     mail.linux.bogus.
5.196.168.192.in-addr.arpa. 259200 IN   PTR     ftp.linux.bogus.
196.168.192.in-addr.arpa. 259200 IN     SOA     ns.linux.bogus. \
        hostmaster.linux.bogus. 199802151 28800 7200 2419200 86400
;; Query time: 6 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Dec 23 03:16:58 2001
;; XFR size: 9 records
```

Red Virtual Privada (VPN)

Hemos instalado una VPN en la red LAN para proteger la información que va desde la DMZ hacia la red en la que se encuentra la base de datos. Los usuarios de la red interna no podrán tener acceso a la información que viaja por el túnel VPN ya que estos datos están encriptados.

El modelo de conexión es red a red. La siguiente figura muestra este modelo.

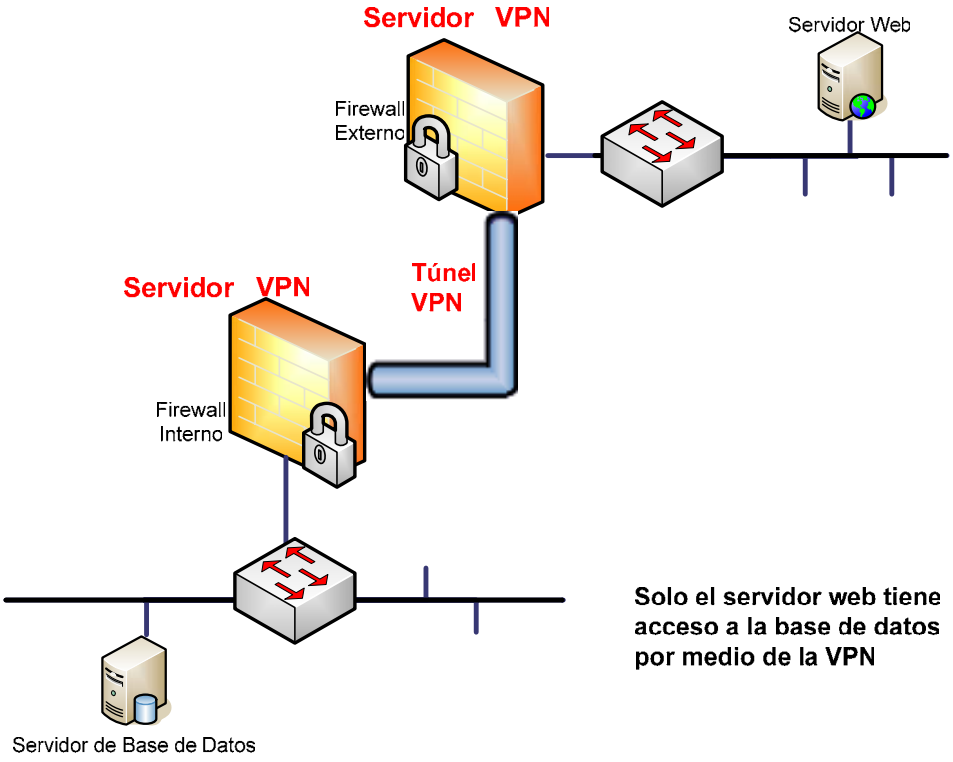


Figura D.1 Modelo de configuración de VPN red a red

El software utilizado para crear la VPN es FreeS/wan, para su instalación y configuración requerimos los siguientes paquetes:

- freeswan-module-2.06_2.4.20_8-0.i386.rpm
- freeswan-userland-2.06-2.4.20_8-0.i386.rpm

Instalación

Usar el comando rpm -ivh para instalar los dos paquetes RPM:

```
# rpm -ivh freeswan*.rpm

warning: freeswan-module-2.00_2.4.18_14-0.i386.rpm: V3 RSA/MD5 signature: NOKEY, key ID 5a7e4731
Preparing...???????? ########################################### [100%]
?? 1:freeswan-module ########################################### [ 50%]
do not forget to install the userland utilities
?? 2:freeswan-userland? ##################################### [100%]
invoke "service ipsec start" or reboot to begin
```

Iniciando FreeS/WAN por primera vez

El siguiente comando iniciará FreeS/WAN, pero también se lo puede hacer creando un script en el directorio **/etc/init.d,** para que se ejecute automáticamente al iniciar el sistema.

```
# service ipsec start

ipsec_setup: Starting FreeS/WAN IPsec 2.00...
ipsec_setup: insmod: ipsec: no module by that name found
ipsec_setup: insmod failed, but found matching template module 30ae280d.
ipsec_setup: Copying /lib/modules/2.4.18-14/kernel/net/ipsec/30ae280d to /lib/modules/2.4.18-14/kernel/net/ipsec/ipsec.o.
ipsec_setup: /sbin/insmod /lib/modules/2.4.18-14/kernel/net/ipsec/ipsec.o
ipsec_setup: Using /lib/modules/2.4.18-14/kernel/net/ipsec/ipsec.o
```

ipsec_setup: Symbol version prefix ''
ipsec_setup: WARNING: changing route filtering on wlan0 (changing /proc/sys/net/ipv4/conf/wlan0/rp_filter from 1 to 0)

Generamos nuestras claves rsa(lo hacemos en cada máquina)

ipsec rsasigkey --verbose 2192 > keys.tmp

getting 128 random bytes from /dev/random...
looking for a prime starting there (can take a while)...
found it after 129 tries.
getting 128 random bytes from /dev/random...
looking for a prime starting there (can take a while)...
found it after 662 tries.
computing modulus...
computing lcm(p-1, q-1)...
computing d...
computing exp1, exp1, coeff...
output...

Luego lo que contiene el archivo keys.tmp lo copiamos en el archivo

/etc/ipsec.secrets entre las llaves RSA {}

Para observar las claves que generamos podemos tipear

ipsec showhostkey --left

??????? # RSA 2192 bits?? vpn1?? Mon Jun 16 21:15:31 2003
???????
leftrsasigkey=0sAQNrV9AYdaW94FXvIxu5p54+MRaW0wy0+HHQrdGofklZY
Q4TCBlL+Ym00Ahfc8mqXlerZY12Os41G8SIV+zzIO04WZ4wmOvEr8DZaldT
bfCuvUvMhrTtCpZdm53yF5rCaUbg+Vmx71jcIVZqwd2AAocrthuC1riwI8yK9H
rSVHzpNQxfRX+F9B//gwWEu1UVcEPkAuY2+Q2tBjg/wmFLEhOrdey7X3NR
KyEa6LqgM2Igjhx6vflQg1ImFFyUAL37ie4YSpDnUVzy3tzBVgyPuFHOGyqZt
uD/+YkNIhrHfgyVmGu8/kuhzB7nWtOYqDFO8OHDGePOyOVPQi73KfRoDbd
b3ND0EtfnRhRPblKJ239OlIq1

Generamos el archivo de configuración, que es el /etc/ipsec.conf para cada equipo. Tenemos dos máquinas que las llamaremos: izquierda y derecha con sus respectivas direcciones ip e interfaces de red. En la siguiente figura podemos observar las configuraciones.

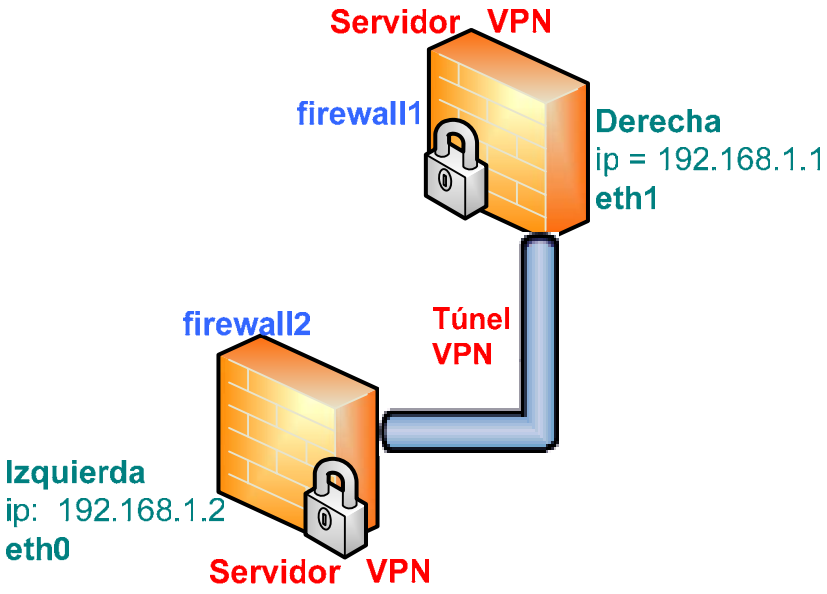


Figura D.2 Configuración de equipos servidores de VPN

Para la máquina de la izquierda con ip = 192.168.1.2, e interfaz eth0 nuestro archivo sería:

```
# basic configuration
config setup
```

```
        # Debug-logging controls:  "none" for (almost) none, "all" for lots.
        # klipsdebug=all
        # plutodebug=dns
        #interfaces="ipsec0=eth0"
        interfaces=%defaultroute
        klipsdebug=none
        plutodebug=none
        uniqueids=yes


# Add connections here.

# sample VPN connection
#sample#     conn sample
#sample#             # Left security gateway, subnet behind it, next hop toward
right.
#sample#             left=10.0.0.1
#sample#             leftsubnet=172.16.0.0/24
#sample#             leftnexthop=10.22.33.44
#sample#             # Right security gateway, subnet behind it, next hop
toward left.
#sample#             right=10.12.12.1
```

```
#sample#        rightsubnet=192.168.0.0/24
#sample#        rightnexthop=10.101.102.103
#sample#        # To authorize this connection, but not actually start it, at
startup,
#sample#        # uncomment this.
#sample#        #auto=start

conn %default
        left=192.168.1.2
        #left=%defaultroute
        leftnexthop=
        leftrsasigkey=0sAQN9BvaENV/4wRpDPPt9mokVc/YAm2MouYrgBi28
AmkO/5BOMHMsnFN84dfih4gJWte8op8N1hLjAodfg7hjmp2SDYaSra+FEsIW
rCAPjpKl7ivSitYTVQ4wOMgN/Ule+BoyaZswdH4iKjHWvg/ODmRKSWZe3A6
vSRv0yBNtYlxF0K3PWJ2o+cQTXuYFmA/Z4BSPOWhjX2jjdG6eaKOZd4qlkF
YBdRJut07UGbrvz+5uIulkTluyT6RX38gBG/W/1Dt/22Hk774pBHKuTJ7dosX8
NiIpHi0aidgc2Q+MMzgiEI7Ve88EezEhm+Ci9P92pIeq6iO4rSceh9fOxVLNUS
mMOPQaa0KzSSjSAlCvJglxoel3
        leftid=@firewall2.freeswan

conn net1-to-net2
        right=192.168.1.1
```

```
        #right=%any
        rightrsasigkey=0sAQOZ6hPwYhtqiS0hYmI2rq4n/haBDXgLUs8Ng+/ak
QWmVONIb3hMlzkoLwccFdkaW5iXMt+XZM5n9tvXTYxAnvaH9E8MnWhqLm
drkt3yo78gwGkTfVot0xUSa+5YSqKGbZEHdKTI/9xxva8kik6ltx+d+lXxyG6FY
XuWHcigOdErBV7mmhp12X+zp4Zz8jSZQRMcNxshWczLXzQnQKuxEvxbs4
AOe8CEcX4p8phaTzEp0eaEU/cWssPgDcx/rbJcdunExJAi0tMtFSu9bs4YjR/r
mzz3xIumxM1JMLwICZHuIDGxWLSJqPD1cn9SuJjm4GvqkSi3xDFjV9KSr5x
9/j+z4ZkoUFbCUXGSukuZy816eM+L
        rightid=@firewall1.freeswan
        rightsubnet=192.168.0.2/32
        leftsubnet=192.168.2.5/32
        auto=start
```

Para la máquina de la derecha con ip = 192.168.1.1, e interfaz eth1 nuestro archivo sería:

```
# basic configuration
config setup
        # Debug-logging controls:  "none" for (almost) none, "all" for lots.
        # klipsdebug=all
        # plutodebug=dns
```

```
        #interfaces="ipsec0=eth1"
        interfaces=%defaultroute
        klipsdebug=none
        plutodebug=none
        uniqueids=yes

# Add connections here.

# sample VPN connection
#sample#     conn sample
#sample#             # Left security gateway, subnet behind it, next hop toward
right.
#sample#             left=10.0.0.1
#sample#             leftsubnet=172.16.0.0/24
#sample#             leftnexthop=10.22.33.44
#sample#             # Right security gateway, subnet behind it, next hop
toward left.
#sample#             right=10.12.12.1
#sample#             rightsubnet=192.168.0.0/24
#sample#             rightnexthop=10.101.102.103
#sample#             # To authorize this connection, but not actually start it, at
startup,
```

```
#sample#            # uncomment this.
#sample#            #auto=start

conn %default
        left=192.168.1.1
        #left=%defaultroute
        leftnexthop=
        leftrsasigkey=0sAQOZ6hPwYhtqiS0hYmI2rq4n/haBDXgLUs8Ng+/akQ
WmVONIb3hMlzkoLwccFdkaW5iXMt+XZM5n9tvXTYxAnvaH9E8MnWhqLmd
rkt3yo78gwGkTfVot0xUSa+5YSqKGbZEHdKTI/9xxva8kik6ltx+d+lXxyG6FYX
uWHcigOdErBV7mmhp12X+zp4Zz8jSZQRMcNxshWczLXzQnQKuxEvxbs4A
Oe8CEcX4p8phaTzEp0eaEU/cWssPgDcx/rbJcdunExJAi0tMtFSu9bs4YjR/rm
zz3xIumxM1JMLwICZHuIDGxWLSJqPD1cn9SuJjm4GvqkSi3xDFjV9KSr5x9/j
+z4ZkoUFbCUXGSukuZy816eM+L
        leftid=@firewall1.freeswan

conn net1-to-net2
        right=192.168.1.2
        #right=%any
        rightrsasigkey=0sAQN9BvaENV/4wRpDPPt9mokVc/YAm2MouYrgBi2
8AmkO/5BOMHMsnFN84dfih4gJWte8op8N1hLjAodfg7hjmp2SDYaSra+FEsI
WrCAPjpKl7ivSitYTVQ4wOMgN/Ule+BoyaZswdH4iKjHWvg/ODmRKSWZe3A
```

6vSRv0yBNtYlxF0K3PWJ2o+cQTXuYFmA/Z4BSPOWhjX2jjdG6eaKOZd4qlk
FYBdRJut07UGbrvz+5uIulkTluyT6RX38gBG/W/1Dt/22Hk774pBHKuTJ7dosX
8NiIpHi0aidgc2Q+MMzgiEI7Ve88EezEhm+Ci9P92pIeq6iO4rSceh9fOxVLNU
SmMOPQaa0KzSSjSAlCvJglxoel3

rightid=@firewall2.freeswan

rightsubnet=192.168.2.5/32

leftsubnet=192.168.0.2/32

auto=start

Para que la configuración tenga efecto, se debe reiniciar el servicio de FreeS/WAN en cada máquina con el siguiente comando:

service ipsec restart

Servidor Proxy

Paquete:

- squid-2.5.STABLE5.bz2

Compilación e instalación

Terminada la descarga de los fuentes, debe procederse a descomprimirlos, para su posterior compilación. La descompresión puede efectuarse por varios

medios, pero aquí detallamos los pasos necesarios para hacerlo desde la consola o terminal:

```
# cd /usr/local/
# tar xvjz squid-2.5.STABLE5.bz2
# cd squid-2.5.STABLE5

# ./configure
# make all
# make install
```

Una vez efectuada la instalación, es necesario efectuar algunas tareas previas a la configuración. Para su funcionamiento, squid demanda la creación de una serie de directorios, típicamente en /usr/local/squid/var/cache, que contendrán la llamada zona de caché. Para hacerlo, debemos verificar que el dueño del directorio /usr/local/squid y todos los directorios por debajo de él, sea el usuario root, para luego ejecutar la orden:

```
# /usr/local/squid/sbin/squid –z
```

Si da algún mensaje de error y no se pudo crear el directorio /usr/local/squid/cache, especificando que no tiene permisos, deberemos ejecutar:

```
# mkdir /usr/local/squid/var/cache
# chmod -R 777 /usr/local/squid/var/cache
# /usr/local/squid/sbin/squid -z
```

Hecho esto, ya estamos en condiciones de leer el archivo de configuración de squid para efectuar algunos cambios mínimos. El archivo de configuración se llama squid.conf y se encuentra en /usr/local/squid/etc. Como se puede constatar, tiene un mar de opciones de configuración.

Configuración del archivo squid.conf

Si se lo instaló con rpm, el archivo se encuentra en /etc/squid/squid.conf

Abrimos el archivo con un editor:

```
# gedit squid.conf
```

http_port. Este parámetro, define en qué puerto responderá a las solicitudes squid. Para nuestra instalación utilizamos el puerto 3128:

```
http_port 3128
```

icp_port. Este parámetro, define el puerto en que el servidor squid recibe solicitudes ICP (Inter-Cache Protocol). Desactivarlo asignando el valor de cero al parámetro:

```
icp_port 0
```

cache_mem. Memoria utilizada por squid para ciertos procesos. En la instalación para un computador de 64MB de memoria utilizamos 8 MB y para un computador con 16MB utilizamos 4 MB y se ha tenido buenos resultados:

cache_mem 16 MB

cache_swap_low. Indica el nivel en porcentaje de capacidad mínima aceptada por squid, es decir, los objetos se mantendrán en el caché hasta que se cope el límite mínimo:

cache_swap_low 90

cache_swap_high. Parámetro, que especifica en porcentaje el límite máximo que utiliza squid para mantener objetos en el caché. Si el valor asignado es del 95%, squid comenzará a eliminar los objetos del caché cuando se tope el 95% de la capacidad asignada a squid:

cache_swap_high 95

maximum_object_size. Este parámetro, especificado en KB, indica el tamaño máximo que se almacena en el caché. Por defecto se utiliza 4MB:

maximum_object_size 8192 KB

cache_dir. Directorio de ubicación del caché, por defecto /usr/local/squid/cache. Este parámetro incluye tres parámetros numéricos adicionales. El primero incluye el número de MB que se utilizarán en este directorio para el caché, por defecto 100MB, el segundo el número de directorios a utilizar en el primer nivel (16 por defecto) y el tercero el número de subdirectorios en el segundo nivel (256 por defecto):

cache_dir ufs /usr/local/squid/var/cache 100 16 256

cache_access_log. Especifica en que directorio se realizará el registro de accesos al squid. Este parámetro es importante para definir posteriormente en el sistema de análisis de estadísticas, webalizer, la ubicación del registro de accesos:

cache_access_log /usr/local/squid/var/logs/access.log

cache_log. Define en donde se almacenan los mensajes del sistema:

cache_log /usr/local/squid/var/logs/cache.log

cache_store_log. Este parámetro, especifica la ubicación del archivo de registro de objetos sacados del caché. No es necesario activarlo. Es mejor desactivarlo para ahorrar espacio en disco:

cache_store_log none

mime_table. Define la ubicación del archivo mime.conf, se utiliza el valor por defecto:

mime_table /usr/local/squid/etc/mime.conf

pid_filename. Define la ubicación del archivo squid.pid, se utiliza el valor por defecto:

pid_filename /usr/local/squid/var/logs/squid.pid

debug_options. Opciones de depuración, se utiliza el valor por defecto:

debug_options ALL,1

quick_abort. Define si un objeto debe almacenarse en el caché, cuando el usuario ha interrumpido una solicitud. Si el objeto tiene el valor especificado en min o falta más del valor especificado en max se abortará la transferencia. Si se ha realizado una transferencia mayor del valor en porcentaje especificado en pct, no se abortará el almacenamiento del objeto. Se recomienda utilizar los valores por defecto:

quick_abort_min 16 KB

quick_abort_max 16 KB

quick_abort_pct 95

negative_ttl. Utilizado para definir cuánto tiempo debe esperar squid para procesar nuevamente una página, que no ha sido encontrada. Se recomiendan 5 minutos:

negative_ttl 5 minutes

positive_dns_ttl. Este parámetro, especifica el tiempo que squid mantendrá la dirección de un sitio visitado exitosamente. El valor por defecto es de 6 horas:

positive_dns_ttl 6 hours

negative_dns_ttl. Especifica el tiempo que espera squid antes de intentar nuevamente determinar la dirección de un sitio solicitado y que no ha sido encontrado. Por defecto 1 minuto:

negative_dns_ttl 5 minutes

Luego, para habilitar la creación de archivos logs para el usuario default, que es el usuario nobody, debido a la directiva cache_effective_user nobody si se desea especificar un usuario perteneciente a un grupo especifico, entonces también se cambia la directiva cache_effective_group grupoX donde grupoX es un grupo ya creado.

chown nobody. /usr/local/squid/var/logs (sin especificar un grupo)

chown nobody.grupoX /usr/local/squid/var/logs (especificando un grupo)

Permisos a la Red Local

En el archivo /usr/local/squid/etc/squid.conf

```
acl redlocal1 src 192.168.0.0/255.255.255.0
acl redlocal2 src 192.168.1.0/255.255.255.0
acl redlocal3 src 192.168.2.0/255.255.255.0
acl ncsa_users proxy_auth REQUIRED

acl redDMZ src 192.168.0.0/255.255.255.0
acl redlocal1 src 192.168.1.0/255.255.255.0
acl redlocal2 src 192.168.2.0/255.255.255.0
acl ncsa_users proxy_auth REQUIRED

http_access allow ncsa_users redDMZ
http_access allow ncsa_users  redLocal1
http_access allow ncsa_users  redLocal2
http_access deny all
```

Creación del fichero de contraseñas.

Se requerirá la creación previa de un fichero, que contendrá los nombres de usuarios y sus correspondientes contraseñas (cifradas). El fichero puede

localizarse en cualquier lugar del sistema, con la única condición que sea asequible para el usuario squid.

Debe procederse a crear un fichero /etc/squid/squid-passwd:

```
# touch /etc/squid/squid-passwd
```

Como medida de seguridad, este fichero debe tener permisor de escritura y lectura solo para el usuario squid:

```
# chmod 600 /etc/squid/squid-passwd
# chown squid:squid /etc/squid/squid-passwd
```

A continuación, deberemos dar de alta las cuentas que sean necesarias, utilizando el comando htpasswd mismo que viene incluido en el paquete *apache-1.3.22* y posteriores. Ejemplo:

```
# htpasswd /etc/squid/squid-passwd joseperez
```

Lo anterior, solicitará teclear una nueva contraseña para el usuario *joseperez* y confirmar tecleando ésta de nuevo. Repita con el resto de las cuentas que requiera dar de alta.

Todas las cuentas que se den de alta de este modo son independientes a las ya existentes en el sistema. Al dar de alta una cuenta o cambiar una contraseña, lo estará haciendo **EXCLUSIVAMENTE** para el acceso al

servidor Proxy. Las cuentas son independientes a las que se tengan existentes en el sistema como serían shell, correo y Samba.

Parámetros en /etc/squid/squid.conf

Lo primero, será especificar qué programa de autenticación se utilizará. Agregar la sección que corresponde a la etiqueta auth_param. Por defecto, no está especificado programa alguno. Considerando que ncsa_auth se localiza en /usr/lib/squid/ncsa_auth, procederemos a añadir el siguiente parámetro:

auth_param basic program /usr/lib/squid/ncsa_auth /usr/etc/passwd

/usr/lib/squid/ncsa_auth, corresponde a la localización de el programa para autenticar y /etc/squid/squid-passwd al fichero que contiene las cuentas y sus contraseñas.

El siguiente paso corresponde a la definición de una Lista de Control de Acceso. Especificaremos una denominada passwd la cual se configurará para utilizar obligatoriamente la autenticación para poder acceder a Squid. Debe localizarse la sección de Listas de Control de Acceso y añadirse la siguiente línea:

acl ncsa_users proxy_auth REQUIRED

Habiendo hecho lo anterior, deberemos tener en la sección de Listas de Control de Acceso algo como lo siguiente:

```
# Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl redlocal src 192.168.1.0/255.255.255.0
acl ncsa_users proxy_auth REQUIRED
```

Procedemos entonces, a modificar la regla de control de accesos que ya teníamos, para permitir el acceso a Internet. Donde antes teníamos lo siguiente:

```
http_access allow redlocal
```

Le añadimos passwd, la definición de la Lista de Control de Acceso que requiere utilizar contraseña, a nuestra regla actual, de modo que quede como mostramos a continuación:

```
http_access allow ncsa_users redlocal
```

Habiendo hecho lo anterior, la zona de reglas de control de acceso debería quedar más o menos de este modo:

```
#
```

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
http_access allow localhost
http_access allow ncsa_users redlocal
http_access deny all
```

Solo bastará reiniciar squid, para que tomen efecto los cambios y podamos hacer pruebas.

```
/etc/rc.d/init.d/squid restart
```

Comentar la línea en Squid.conf que comienza

```
auth_param basic realm Squid proxy-caching web server
```

y añadimos la siguiente

```
auth_param basic realm paguitos.es.mw
```

Donde paguitos.es.mw es el nombre del dominio para nuestro website

Iniciando Squid Automaticamente

Para que el squid se inicie de manera automática

```
/sbin/chkconfig --level 345 squid on
```

Detector de Intrusos

Paquetes:

- mysql-4.1.9.tar.gz
- httpd-2.0.49.tar.gz
- gd-2.0.33.tar.gz
- php-4.3.11.tar.bz2
- libcap-0.7.2-1.i386.rpm
- snort-2.3.3.tar.gz
- jpgraph-1.17.tar.gz
- adodb390.tgz
- acid-0.9.6b23.tar.gz

Instalación MYSQL

Crear el usuario y grupo mysql:

```
# groupadd mysql
# useradd -g mysql mysql
```

Editar el archivo .bash_profile del root:

```
# gedit /root/.bash_profile
```

Actualizar la siguiente línea del archivo:

PATH=$PATH:$HOME/bin:/usr/local/mysql/bin

Descomprimimos el fuente de mysql

```
# tar -xvzf mysql-4.1.9.tar.gz
# cd mysql-4.1.9
# ./configure --prefix=/usr/local/mysql
# make
# make install
```

Creamos la carpeta var dentro del directorio de la instalación de mysql.

```
# mkdir /usr/local/mysql/var
```

Ejecutamos:

```
# scripts/mysql_install_db
```

Colocamos los permisos respectivos:

```
# chown -R root /usr/local/mysql
# chown -R mysql /usr/local/mysql/var
# chgrp -R mysql /usr/local/mysql
```

Copiamos un archivo de configuración:

```
# cp support-files/my-medium.cnf /etc/my.cnf
```

Añadir las siguientes líneas en el archivo /etc/ld.so.conf

/usr/local/mysql/lib/mysql

/usr/local/lib

Después de añadir las líneas ejecutamos:

ldconfig -v

Para verificar que todo este bien ejecutamos.

/usr/local/mysql/bin/mysqld_safe --user=mysql &

Si sale algún mensaje de (mysql ended) significa que hubo un error, generalmente los errores son por causa de permisos, si desea ver el error que surgió, visualice el archivo /usr/local/mysql/var/mi_maquina.domain.err. También puede volver a ejecutar las tres líneas de permisos para ver si se arregla el problema.

Hacemos que mysql se ejecute automáticamente

Nos ubicamos donde descomprimimos el fuente:

cd mysql-4.1.9

cp support-files/mysql.server /etc/init.d/mysql

cd /etc/rc3.d/

#l n -s ../init.d/mysql S85mysql

```
# ln -s ../init.d/mysql K85mysql
# cd /etc/rc5.d/
# ln -s ../init.d/mysql S85mysql
# ln -s ../init.d/mysql K85mysql
# cd ../init.d/
# chmod 755 mysql
```

Y ahora podemos arrancar mysql con el siguiente comando:

```
# service mysql start
```

Y podemos parar el servicio con el comando:

```
# service mysql stop
```

Instalación y Configuración de APACHE

Compilamos e instalamos el fuente de apache:

```
# ./configure –prefix=/usr/local/apache –enable-module=so
# make
# make install
```

Podemos levantar el servicio apache con el comando

```
# /usr/local/apache/bin/apachectl start
```

Y para detener el servicio apache.

```
# /usr/local/apache/bin/apachectl stop
```

Ahora, vamos a crear un servicio para apache para poder levantarlo automáticamente.

```
# cp /usr/local/apache/bin/apachectl /etc/init.d/apache
# cd /etc/rc3.d/
# ln -s ../init.d/apache S85apache
# ln -s ../init.d/apache K85apache
# cd /etc/rc5.d/
# ln -s ../init.d/apache S85apache
# ln -s ../init.d/apache K85apache
```

Ahora podemos usar los siguientes comandos para subir y bajar el servicio de apache:

```
# service apache start
# service apache stop
```

Instalación del paquete GD

Nos ubicamos en la ruta donde nos bajamos el paquete gd-2.0.33.tar.gz

```
# cd /home/
```

Descomprimimos el archivo

tar –xvzf gd-2.0.33.tar.gz

#cd gd-2.0.33

Instalamos el paquete:

./configure

make

make install

La ruta por defecto donde se instala el paquete GD es /usr/local/

Instalación y configuración de php-4.3.11.tar.bz2

cd /home/

tar -xvjf php-4.3.11.tar.bz2

cd php-4.3.11

./configure --prefix=/usr/local/apache/php --with-apxs2=/usr/local/apache/bin/apxs --with-config-file-path=/usr/local/apache/php --enable-sockets --with-mysql=/usr/local/mysql --with-zlib-dir=/usr/local --with-gd

make

make install

Copiar el archivo de configuración que se encuentra en los fuentes y pasarlo a la ruta donde se instaló nuestro php.

#cp php.ini-dist /usr/local/apache/php/php.ini

Editar el archivo /usr/local/apache/conf/httpd.conf

gedit /usr/local/apache/conf/httpd.conf

Cargar el módulo para php:

LoadModule php4_module modules/libphp4.so

Añadir o quitar los comentarios a las siguientes líneas, si es que no están añadidas:

AddType image/x-icon .ico

AddType application/x-httpd-php .php .phtml .html

AddType application/x-httpd-php-source .phps

DirectoryIndex index.php index.html index.html.var

Para probar la instalación del php podemos crear un archivo de prueba llamado index.php que contendrá <?php phpinfo(); ?>

Luego levantamos nuestro apache:

/usr/local/apache/bin/apachectl start

Y colocamos http://localhost

Y nos mostrará información de nuestro php.

Instalación del paquete LIBCAP-0.7.2-1

Instalamos la librería libpcap para poder instalar el snort:

Verificamos que no este instalada o si no obviamos la instalación del paquete.

```
# rpm -q libcap
```

Si es que no está instalada entonces buscamos el paquete. Se encuentra en el disco 2 de Red Hat Linux.

Instalamos el paquete libcap-0.7.2-1.i386.rpm

```
# rpm -ivh /home/libpcap-0.7.2-1.i386.rpm
```

Instalación SNORT-2.3.3

```
# groupadd snort
# useradd -g snort snort
# mkdir /etc/snort
# mkdir /var/log/snort
```

```
# tar -xvzf snort-2.3.3.tar.gz
# cd snort-2.3.3
# ./configure –with-mysql=/usr/local/mysql
# make
# make install
```

Copiando las reglas y el archivo de configuración:

```
# cd /snort-2.3.3/rules
# cp * /etc/snort
# cd ../etc
# cp snort.conf /etc/snort
# cp unicode.map /etc/snort/
# cp *.config /etc/snort
```

Modificar el archivo de configuración snort.conf localizado en /etc/snort/

```
# gedit /etc/snort/snort.conf
```

Modificar la variable HOME_NET con la red interna:

var HOME_NET 10.2.2.0/24

Modificamos la variable que contendrá nuestro servidor de correos

var SMTP_SERVERS 192.168.1.3

Cambiar la ruta de las reglas de snort:

var RULE_PATH /etc/snort/

Decirle a snort que guarde sus logs en la base de datos:

output database: log, mysql, user=snort password=snort dbname=snort host=localhost

Configuración de Mysql para que interactúe con snort.

/usr/local/mysql/bin/mysql

Colocamos un password para el usuario root.

mysql> SET PASSWORD FOR root@localhost=PASSWORD('mi_password');

Creamos la Base que usamos para guardar los logs de snort.

mysql> create database snort;

Podemos verificar si la Base se creó con el comando

mysql> show databases;

mysql> grant INSERT,SELECT on root.* to snort@localhost;

Colocamos un password para el usuario snort.

mysql> SET PASSWORD FOR snort@localhost=PASSWORD('mi_password2');

Colocamos permisos al usuario snort para la base recién creada snort.

mysql> grant CREATE,INSERT,SELECT,DELETE,UPDATE on snort.* to snort@localhost;

mysql> grant CREATE,INSERT,SELECT,DELETE,UPDATE on snort.* to snort;

Creamos las tablas para nuestra base snort.

Nos colocamos en la ruta donde descomprimimos los fuentes de snort.

cd /home/snort-2.3.3

Ejecutamos el siguiente comando para crear las tablas:

/usr/local/mysql/bin/mysql -u root -p < ./schemas/create_mysql snort

Luego verificamos que se crearon las tablas

/usr/local/mysql/bin/mysql -p

Colocamos el password para el usuario root:

Enter password: mi_password

Estando en el prompt de mysql.

mysql> SHOW DATABASES;

mysql> use snort;

mysql> SHOW TABLES;

mysql> exit;

Instalación el paquete JPGRAPH-1-17

Ir donde se tiene el paquete del jpgraph

cd /home/

Y ejecutar

tar -xvzf jpgraph-1.17.tar.gz -C /usr/local/apache/htdocs/

Instalación ADODB390

Ir donde se tiene el paquete del ADODB

cd/home/

Y ejecutar:

tar -xvzf adodb390.tgz -C /usr/local/apache/htdocs/

Instalación y configuración ACID-0.0.6b23

Ir donde se tiene el paquete del acid

```
# cd /home/
```

Y ejecutar

```
# tar -xvzf acid-0.9.6b23.tar.gz -C /usr/local/apache/htdocs/
```

Configurando ACID

Abrir el archivo de configuración de Acid:

```
# gedit /usr/local/apache/htdocs/acid/acid_conf.php
```

Modificar ciertas líneas con lo siguiente.

```
$DBlib_path = "/usr/local/apache/htdocs/adodb";

/* The type of underlying alert database
 *
 *  MySQL       : "mysql"
 *  PostgresSQL : "postgres"
 *  MS SQL Server : "mssql"
 */
$DBtype = "mysql";
```

```
/* Alert DB connection parameters
 *   - $alert_dbname   : MySQL database name of Snort alert DB
 *   - $alert_host     : host on which the DB is stored
 *   - $alert_port     : port on which to access the DB
 *   - $alert_user     : login to the database with this user
 *   - $alert_password : password of the DB user
 *
 *  This information can be gleaned from the Snort database
 *  output plugin configuration.
 */
$alert_dbname   = "snort";
$alert_host     = "localhost";
$alert_port     = "";
$alert_user     = "snort";
$alert_password = "snort";

/* Archive DB connection parameters */
$archive_dbname   = "snort";
$archive_host     = "localhost";
$archive_port     = "";
$archive_user     = "snort";
$archive_password = "snort";
```

```
/* Path to the graphing library
 *  (Note: DO NOT include a trailing backslash after the directory)
 */
$ChartLib_path = "/usr/local/apache/htdocs/jpgraph-1.17";
```

Ahora podemos probar el acid. Primero levantamos nuestro servidor apache, el mysql y el snort.

```
# /usr/local/apache/bin/apachectl start
# service mysql start
# snort -c /etc/snort/snort.conf
```

Luego en nuestro navegador colocamos la dirección http://localhost/acid/acid_main.php

Aplicando seguridad para el directorio de ACID

```
# mkdir /usr/local/apache/passwords
```

Crear un archivo passwords que contendrá los usuarios permitidos, en este caso el usuario acid:

```
#  /usr/local/apache/bin/htpasswd  -c  /usr/local/apache/passwords/passwords acid
```

Luego nos pedirá que ingresemos un passowrd y que lo confirmemos.

Editamos el archivo httpd.conf e insertamos las siguientes líneas en la sección </Directory>

```
<Directory "/usr/local/apache/htdocs/acid">
    AuthType Basic
    AuthName "SnortIDS"
    AuthUserFile /usr/local/apache/passwords/passwords
    Require user acid
</Directory>
```

Bajamos el servicio de apache.

```
# service apache stop
```

Y lo levantamos de nuevo:

```
# service apache start
```

Podemos probar ingresando a la página de administración de ACID:

http://localhost/acid/

Como es la primera vez que se ingresa a la página de administración de Acid, este nos pedirá ciertos pasos de configuración adicionales.

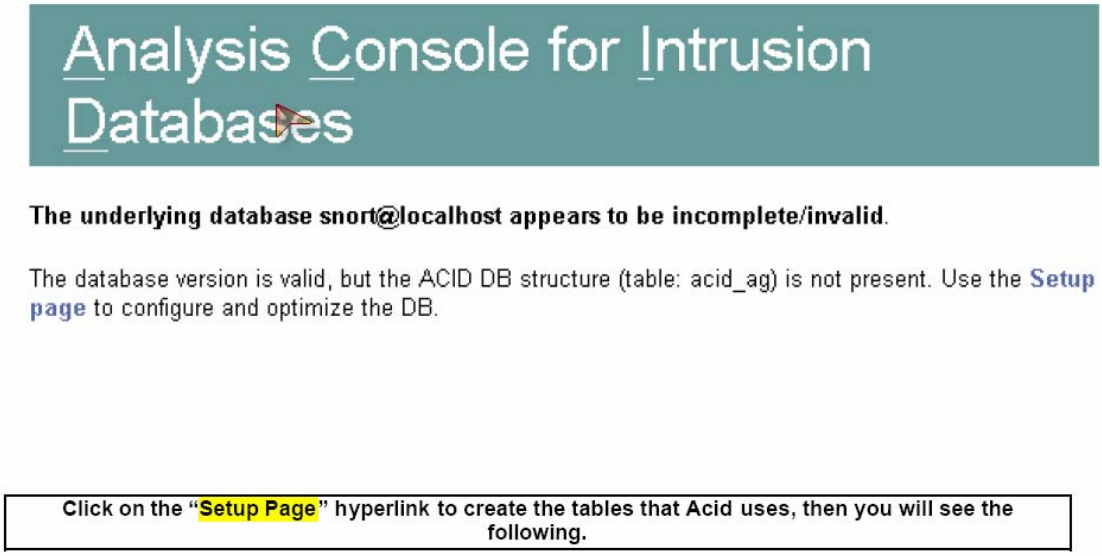
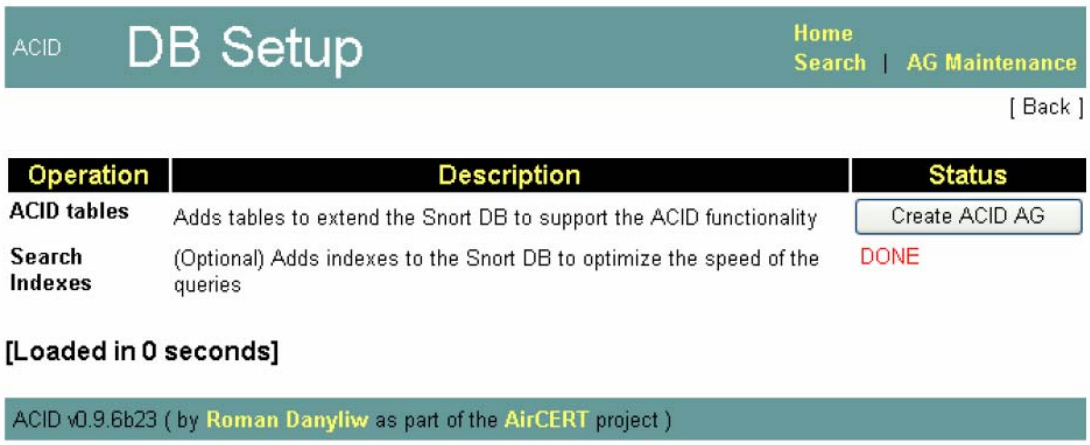


Figura D.3 Pantalla mostrada cuando se ejecuta por primera vez ACID.



Then click the button that says "Create Acid AG"

Figura D.4 Pantalla para optimización y configuración de la Base de datos para ACID.

Una vez creada las tablas adicionales para acid, nos mostrará la consola para administración de logs donde podemos monitorear las alertas del snort.

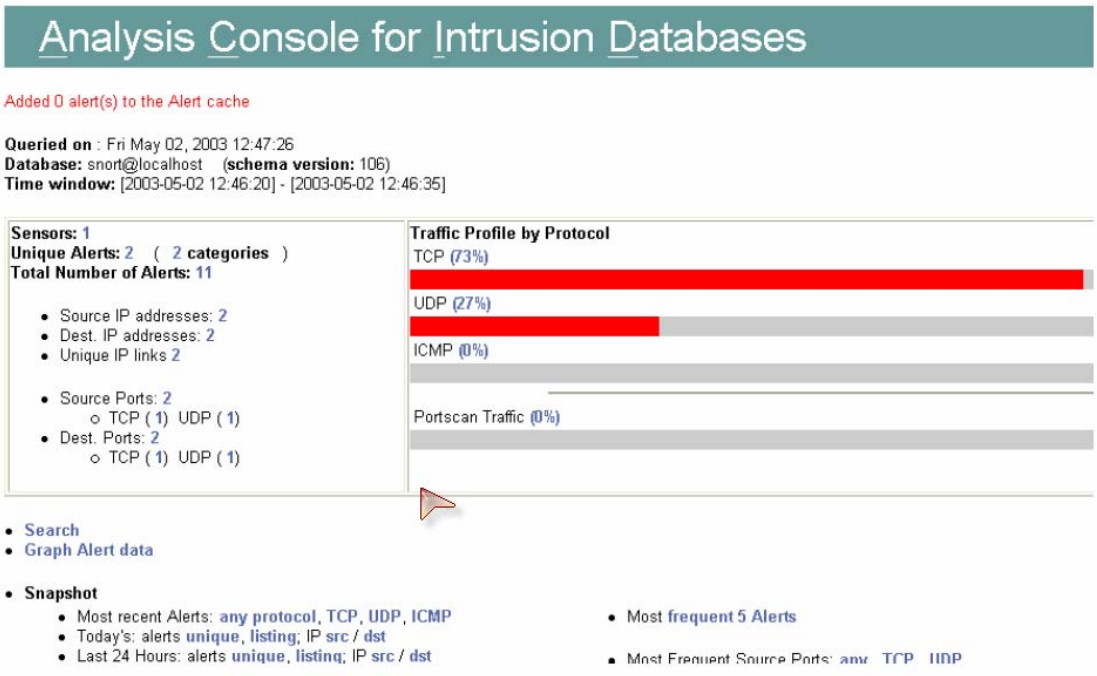


Figura D.5 Pantalla de Análisis de alertas.

BIBLIOGRAFÍA

1. Altadill Izura Pello Xabier, Iptables Manual Práctico, http://www.pello.info/filez/firewall/iptables.html

2. Alvarez Marañón Gonzalo; 2000, La necesidad de un canal seguro, http://www.iec.csic.es/criptonomicon/comercio/ssl.html

3. Alvarez Marañón Gonzalo; 2000, Transacciones Electrónicas Seguras (SET), http://www.iec.csic.es/criptonomicon/comercio/set.html

4. Andreasson Oskar, 2003, Iptables Tutorial 1.1.19, http://iptables-tutorial.frozentux.net/iptables-tutorial.html

5. Apache el Comienzo, http://www.hospedajeydominios.com/mambo/documentacion-manual_apache-pagina-44.html.

6. Borghello Cristian F.; 2001, Tesis sobre seguridad informática, http://www.htmlweb.net/seguridad/tesis/Cap8.pdf

7. Cisco Systems; CISCO CATALYST 2950 12 SWITCH, http://www.cisco.com/en/US/products/hw/switches/ps628/ps626/index.html

8. Copyright © 2003 Red Hat, Inc; 2003,Notas de última hora de Red Hat Linux 9, http://redhat.mirrors.redwire.net/install9/RELEASE-NOTES-es.html

9. Corporación Ecuatoriana de Comercio Electrónico; abril 2003, Ley de Comercio Electrónico, Firmas Electronica y mensaes d datos del Ecuador, http://www.corpece.org.ec/documentos/ley/ley_ce.zip.

10. Corporación Ecuatoriana de Comercio Electrónico; abril 2003, Reglamento a la Ley de Comercio Electrónico Firmas Electrónicas y Mensajes de datos del Ecuador, http://www.corpece.org.ec/documentos/leyes_ecuador/reglamento%20Ley%20CE%20Corpece.zip.

11. Del Castillo San Félix Alvaro, 2000, El Servidor de Web Apache, http://usuarios.lycos.es/b0ls0p3ll0n/manuales/apache.pdf.

12. Devoto Mauricio; 2005, Argentina: La Economía Digital el dinero electrónico y el lavado de dinero, http://www.alfa-redi.org/revista/data/2-3.asp

13. Díaz Oscar Miguel; 2003, Tu Propio Servidor Web, http://business.fortunecity.com/bren/126/servidor1.htm.

14. Evgueni Tzvetanov; Junio 2003, Oracle 9i under RedHat Linux 8.x and 9.x - Simple Installation HOWTO, http://www.tldp.org/HOWTO/Oracle-9i-RH8-and-RH9-HOWTO-3.html

15. Froufe Agustín, Enero 1997, Tutorial de Java, http://www.cica.es/formacion/JavaTut/

16. Grennan Mark; febrero 2000, Cómo de cortafuegos y servidor Proxy, http://www.grennan.com/Firewall-HOWTO.html

17. Iturraspe Urtza, Zaballa Ibon; ¿Qué es un certificado Digital?, http://revista.robotiker.com/articulos/articulo51/pagina1.jsp

18. Jaume Sabater; marzo 2003, El sistema de nombres de dominio: Bind 9.2.1, http://www.linuxsilo.net/articles/bind.html.

19. Laboratorio DEI, Abril 2005, Programación Web con Servlets y JSP, http://www.dei.inf.uc3m.es/docencia/p_s_ciclo/pa4/practicas/jsp.pdf.

20. Larman Craig; 1999, UML y Patrones.

21. Marketing Express; MARKETING como generar MAS VISITAS => MAS VENTAS en su sitio Web, http://www.mke.com.ar

22. Merlat Máximo, Paz Gonzalo, Sosa Matías, Martínez Marcelo; 1997, Hackers, http://www.monografias.com/trabajos/hackers/hackers.shtml

23. Miranda Pablo,Sahakian Arsen, 2004, Introducción a la plataforma Eclipse, http://www.fing.edu.uy/inco/grupos/coal/investigacion/proyectos/lead/docs/IntroduccionEclipse.pdf.

24. OpenSTA; marzo 2005, OpenSTA user Home, http.//www.opensta.org

25. Peruserver S.A.C., 2005, Servidor Proxy, http://www.peruserver.com/sop_proxy.php

26. Red Hat, Inc.; 2003, Berkeley Internet Name Domain (BIND), http://linux-cd.com.ar/manuales/rh9.0/rhl-rg-es-9/ch-bind.html.

27. Red Hat, Inc.; 2003, Configuración del Servidor Seguro Apache HTTP, http://www.europe.redhat.com/documentation/rhl9/rhl-cg-es-9/ch-httpd-secure-server.php3

28. Red Hat, Inc.; 2003, Iptables, http://linux-cd.com.ar/manuales/rh9.0/rhl-rg-es-9/ch-iptables.html.

29. RedIRIS, Agosto 2000, Instalación de una caché WWW, http://www.rediris.es/si/cache/instalar-cache.es.html

30. Ruz Miguel A.; 2001, PROTOCOLO SSL, http://www.delitosinformaticos.com/especial/seguridad/ssl.shtml

31. Sabater Malondra Jaume Andreu, Marzo 2003, El sistema de nombre de dominio: Bind 9.2.1, http://www.linuxsilo.net/articles/bind.html.

32. Spenneberg Ralf, 2003, Ipsec Cómo, http://www.ipsec-howto.org/spanish/t1.html.

33. Storkel Scott, Noviembre 2002, An Introduction to the Eclipse IDE, http://www.onjava.com/pub/a/onjava/2002/12/11/eclipse.html.

34. The Apache Software Foundation, 2005, Apache Jakarta Tomcat, http://jakarta.apache.org/tomcat.

35. Tobar Donna; 2002, Conozca la Ley, http://www.corpece.org.ec/informante/18_22_febrero_2002.htm

36. Valera Gilda Isabel,Inoa Frank Joel, Herrera Elaine Altagracia; 1997, Programación Orientada a Objetos, Oracle y Sql Server, http://www.monografias.com/trabajos4/basesdatos/basesdatos.shtml

37. Viana David Rubert, 1998, Tutorial de Sendmail, http://www.linux.org.ni/LuCAS/LuCAS/Universitarios/tutorial-sendmail.html

38. Welsh Matt, Agosto 1998, Linux: Instalación y Primeros Pasos, http://www.ciberdroide.com/misc/novato/lipp-1.1-html-1.1/lipp.htm

39. Alfon, Febrero 2004, Taller de Sistemas de Detección de Intrusos, http://www.nautopia.net/archives/es/varios_redes/snort/snort.php.