ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UN PORTAL CAUTIVO QUE PERMITA LA VENTA DE TICKETS DE INTERNET PARA UN HOTSPOT, EMPLEANDO HERRAMIENTAS DE SOFTWARE LIBRE

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN

DAVID RICARDO CRUZ HERRERA
davc2011@hotmail.com

DIRECTOR: ING. CARLOS HERRERA
carlos.herrera@epn.edu.ec

Quito, 2011

DECLARACIÓN

Yo, David Ricardo Cruz Herrera, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____

David Ricardo Cruz Herrera

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por David Ricardo Cruz Herrera, bajo mi supervisión.

_____

Ing. Carlos Herrera

DIRECTOR DE PROYECTO

AGRADECIMIENTO

A Dios, gracias por permitirme vivir y alcanzar la meta propuesta.

A mis padres y hermano, quienes supieron ser mi guía y mi apoyo en todas las circunstancias de mi vida.

A mis profesores, y en especial a mi director de Proyecto de Titulación, por brindarme todo el conocimiento necesario para la realización del presente proyecto.

A mis familiares y amigos, por todo el apoyo y confianza que me brindan.

DEDICATORIA

Todo el esfuerzo empleado en la realización de este trabajo va dedicado a mis padres y hermano, que sin importar las circunstancias me han brindado su completo y total apoyo.

CONTENIDO

CAPÍTULO 1: PORTALES CAUTIVOS

CAPÍTULO 2: DISEÑO E IMPLEMENTACIÓN DEL PORTAL CAUTIVO

CAPÍTULO 3: PRUEBAS Y RESULTADOS

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

LISTADO DE FIGURAS

LISTADO DE TABLAS

RESUMEN

Este Proyecto de Titulación se enfoca en el diseño e implementación de un Portal Cautivo que permita la venta de tickets de Internet para un HotSpot, empleando herramientas de software libre.

La estructura de los capítulos que forman la documentación de Proyecto se presenta a continuación.

En el Capítulo 1 se estudian las definiciones, características y funcionalidades que tiene y provee un HotSpot y un Portal Cautivo. Se realiza una breve descripción de las Redes WLAN (Redes de Área Local Inalámbricas) cubriendo conceptos de: funcionalidad, estándares que ocupan, seguridad, aplicaciones, etc. para ser aplicados en el HotSpot. Se realiza una breve descripción de uno de los principales componentes del sistema operativo Linux, Iptables que permiten interceptar y manipular paquetes de red. Se realiza además una descripción del conjunto de especificaciones desarrolladas por Visa y MasterCard denominada SET (Transacción Electrónica Segura).

En el Capítulo 2 se orienta al diseño del Portal Cautivo, que permite la venta de tickets de Internet para un HotSpot, partiendo de que un usuario que desea ingresar a la red primero debe interactuar con una página web, la cual le solicita un nombre de usuario y una contraseña, permitiéndole así autenticarse con la red. Se realiza además una descripción de las herramientas de software libre utilizadas para la implementación del Portal Cautivo, entre ellas: PHP, MySQL, BIND, Apache, DHCPD, Iptables, entre otras.

En el Capítulo 3 se exponen las pruebas realizadas que permiten el acceso al Portal Cautivo, de parte de los usuarios y del vendedor. Se explica además el algoritmo utilizado para la validación de la tarjeta de crédito.

En el Capítulo 4 se exponen las principales conclusiones y recomendaciones a las que se llegó una vez finalizado el Proyecto. Se expone, además, la bibliografía de consulta utilizada para el desarrollo del Proyecto.

PRESENTACIÓN

En la actualidad, debido al alto crecimiento y evolución que se ha dado en la tecnología inalámbrica, tanto en dispositivos móviles como en dispositivos de red, el uso de Portales Cautivos es más frecuente, permitiendo a un usuario de una red pública y/o privada interactuar primero con una página web antes de garantizar su acceso a las funciones normales de la red.

Cuando un usuario potencial se autentica por primera vez ante una red con un Portal Cautivo, se presenta una página web en la cual se requieren ciertas acciones antes de proceder con el acceso.

En otros Portales Cautivos se provee además publicidad para el proveedor y/o sus patrocinadores antes de continuar con el uso del servicio. En la mayoría de casos se debe ingresar una identificación y/o clave asignada antes de acceder al Internet, con el objetivo de desalentar a quienes quieran usar estos servicios para usos no autorizados.

Estos Portales Cautivos son principalmente utilizados por centros de negocio, aeropuertos, hoteles, cafeterías, cafés Internet y otros proveedores que ofrecen HotSpots de Wi-Fi para usuarios de Internet.

CAPÍTULO 1: PORTALES CAUTIVOS

1.1 PORTAL CAUTIVO

1.1.1 DEFINICIÓN

Un portal cautivo (o captivo) es un programa o máquina de una red pública y/o privada que vigila el tráfico HTTP[1] y obliga a los usuarios a pasar por una página Web especial, en la cual deben ingresar un nombre de usuario (username) y una contraseña (password) asignadas, para así poder navegar por Internet de forma normal. En la Figura 1.1 se indica un ejemplo de Portal Cautivo.

Figura 1.1 Ejemplo de Portal Cautivo

Como se indica en la Figura 1.2, estos portales son principalmente utilizados por centros de negocios, aeropuertos, hoteles, cafeterías, cafés Internet y otros proveedores que ofrecen HotSpots[2] de Wi-Fi para usuarios de Internet.

Figura 1.2 Sitios en donde principalmente son utilizados los Portales Cautivos

[1] HTTP: Protocolo de Transferencia de Hipertexto (*HyperText Transfer Protocol*).
[2] HotSpot: Zona de cobertura Wi-Fi. Generalmente se encuentran en lugares públicos como bibliotecas, aeropuertos, centros de convenciones, bibliotecas, etc.

1.1.2 CARACTERÍSTICAS GENERALES DE LOS PORTALES CAUTIVOS[3]

A continuación se mencionan las características más relevantes de un Portal Cautivo:

- Independiente de la plataforma (Windows, Linux).
- Soporte para clientes con acceso alámbrico o inalámbrico.
- Filtrado de paquetes, ya sea por dirección MAC[4], IP o por URLs. Se puede especificar reglas de bloqueo para una IP o puertos para tráfico saliente. La opción puede ser usada para limitar específicamente ciertos servicios, como limitar el uso de FTP[5] o correo electrónico bloqueando esos puertos.
- No se necesita una configuración del lado del cliente, no requiere instalarse ningún programa en el PC del cliente, el Portal Cautivo asegura el enrutamiento de todos los clientes a la pantalla de inicio de sesión.
- Cualquier sistema operativo o navegador puede ser usado en el PC del cliente (Internet Explorer, Firefox, Opera, Safari, Konqueror).
- Control de información de los usuarios (Datos Personales).
- Tickets pre pagados, se pueden crear cuentas por adelantado e imprimirlas como tickets. Se puede vender estos tickets en el mostrador, recepción o usando máquinas de venta.
- Se puede configurar los tickets para mostrar el logotipo de su empresa, nombre del HotSpot, encabezado y pie de página. Opcionalmente se puede imprimir fecha y hora en el ticket.
- Soporta múltiples localizaciones, los usuarios podrán usar el tiempo restante en cualquier HotSpot de la red.
- Las cuentas tienen límite de tiempo, cuando la sesión termina, el cliente puede continuar usando Internet después de comprar tiempo adicional.
- Pagos con tarjeta de crédito, el proceso de pago es completamente automatizado y el cliente puede escoger usuario, contraseña y plan.
- Existen portales cautivos por hardware y por software.

[3] http://www.antamedia.com/hotspot/
[4] MAC: Control de Acceso al Medio (*Media Access Control*)
[5] FTP: Protocolo de Transferencia de Archivos (*File Transfer Protocol*)

- No se necesita obligatoriamente de un personal para vender tickets. Una interfaz simple de usar, habilita el proceso de transacción con los mayores servicios de pago electrónico de Internet: PayPal, AlertPay, Google Chekout, entre otros.

1.1.3 PORTALES CAUTIVOS POR HARDWARE

Son dispositivos dedicados, que permiten la administración del acceso de usuarios a Internet, en negocios y en áreas públicas. No necesitan de un computador, tampoco de otros dispositivos adicionales, entre ellos: Servidor de Base de Datos, Servidor DHCP[6], Firewall, Servidor Web. Todo viene integrado en el mismo dispositivo.

A continuación se mencionan dos ejemplos de Portales Cautivos por hardware, detallando sus principales características:

1.1.3.1 Endian HotSpot[7]

Endian HotSpot es una herramienta completa y flexible para la administración del acceso de usuarios a Internet en negocios y en áreas públicas. El HotSpot de Endian ofrece a sus clientes y usuarios un acceso fácil y seguro a Internet en hoteles, bibliotecas, escuelas, aeropuertos, bancos y cyber-cafés.

Se puede elegir la forma de navegación de los usuarios: sesiones basadas en tiempo o en tráfico, administradas con tickets prepago, pospago o de libre acceso.

Se puede realizar un control de cuentas y registros de acceso de los usuarios en tiempo real. Además, se puede monitorear el estado de los clientes y llevar una lista detallada de sus movimientos y el balance de sus cuentas.

[6] DHCP: Protocolo de Configuración Dinámica de Host (*Dynamic Host Configuration Protocol*)
[7] http://www.endian.com/en/solutions/technology/endianhotspot/

Endian HotSpot ofrece una solución libre de complicaciones para el acceso seguro a internet en áreas públicas. Se puede crear una página de inicio para el HotSpot con la ayuda del "editor inteligente" que viene ya integrado a la interfaz de la web de Endian HotSpot.

Se puede conectar computadores de escritorio, computadores portátiles y puntos de acceso a Internet alámbricos e inalámbricos sin importar el tipo de sistema operativo utilizado.

Con la ayuda del Servidor SSL[8] VPN[9] y el Cliente VPN seguro de Endian, los usuarios con acceso inalámbrico pueden comunicarse sin esfuerzo y con completa seguridad. El Firewall SSL VPN de Endian permite establecer conexiones SSL encriptadas entre los clientes inalámbricos y la red de forma rápida y fácil. Esto ayuda a impedir la intromisión de hackers en las comunicaciones inalámbricas. El Cliente VPN seguro de Endian trabaja con Windows, Linux y Mac OSX. Así, los clientes siempre estarán protegidos sin importar cuál sea el sistema operativo utilizado.

Se pueden especificar filtros de contenido, filtros web, anti-virus, antispam y otras opciones de protección para dar una mayor confiabilidad a la hora de proteger el sistema.

En la Figura 1.3 se muestra un ejemplo de diagrama de red usando Endian HotSpot.

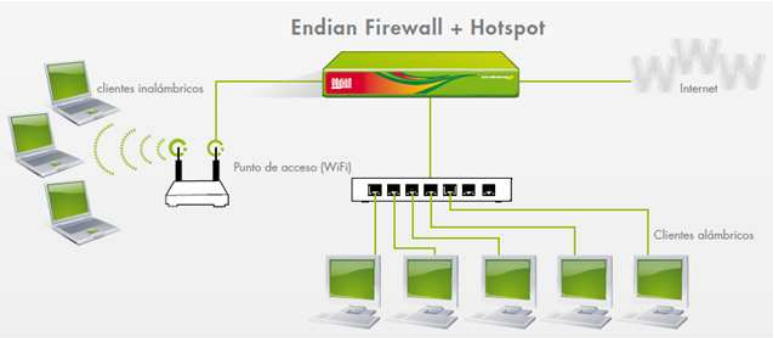


Figura 1.3 Ejemplo de una red con Endian HotSpot

[8] SSL: Protocolo de Capa de Conexión Segura (*Secure Socket Layer*)
[9] VPN: Red Privada Virtual (*Virtual Private Network*)

1.1.3.2 4ipnet HSG300 Wireless HotSpot Gateway[10]

Es una herramienta de fácil manejo que permite controlar el acceso de los usuarios al Internet, en una red inalámbrica. No se necesita de un experto para la configuración del dispositivo, presenta una interfaz de administración web muy sencilla de manejar.

Está integrado con un Access Point, el cual permite el acceso a usuarios inalámbricos. No se necesita de dispositivos Wi-Fi adicionales, reduciendo así los costos de la implementación (Figura 1.4)

Figura 1.4 HSG300 Wireless HotSpot Gateway

Está integrado con una base de datos, con capacidad suficiente para abarcar las diferentes cuentas de usuario y en diferentes lugares de operación. Provee 1000 cuentas para lo que son usuarios propios de la red y 2000 para lo que son visitantes (usuarios ajenos a la red). Soporta múltiples métodos de autenticación de usuarios: RADIUS[11], entre otros. La página de ingreso está protegida con SSL, asegurando así que las cuentas de los usuarios están encriptadas. Permite además múltiples sesiones con la misma cuenta. Soporta varios estándares de seguridad: WEP[12] (64/128/152 bits), WPA[13]/WPA2-PSK[14], IEEE[15] 802.1X, y TKIP[16] & AES[17]. Provee varios tipos de cuentas, por: duración, uso y volumen de datos. Soporta diferentes mecanismos de pago: en efectivo, por tarjeta de crédito.

---

[10] http://www.4ipnet.com/en/products_detail.php?name=HSG300
[11] RADIUS: Protocolo de autenticación y autorización para aplicaciones de acceso a la red. (*Remote Authentication Dial-In User Server*)
[12] WEP: Privacidad Equivalente a Cableado (*Wired Equivalent Privacy*)
[13] WPA: Acceso Protegido Wi-Fi (*Wi-Fi Protected Access*)
[14] PSK: Llave Pre-Compartida (*Pre-Shared Key*)
[15] IEEE: Instituto de Ingenieros Eléctricos y Electrónicos (*Institute of Electrical and Electronics Engineers*)
[16] TKIP: Protocolo de Integridad de Llave Temporal (*Temporal Key Integrity Protocol*)
[17] AES: Estándar de Encripción Avanzada (*Advanced Encryption Standard*)

Filtro de contenido, filtro de la web y otras opciones de protección están todas disponibles para dar mayor seguridad a la hora de proteger el sistema.

En la Figura 1.5 se muestra un ejemplo de diagrama de red usando Wireless HotSpot Gateway HSG300.

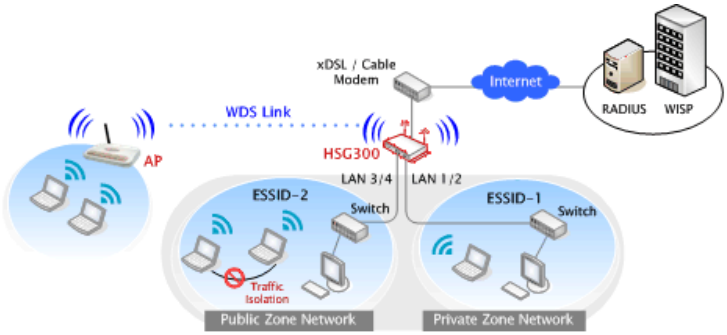


Figura 1.5 Ejemplo de una red con HSG300 Wireless HotSpot Gateway

1.1.4 PORTALES CAUTIVOS POR SOFTWARE

Son programas (para PC) que implementan un Portal Cautivo. Necesitan de un computador, el cual debe tener por lo menos 2 interfaces de red: la una para conectarse al Access Point y la otra a la salida del Internet. Depende del sistema operativo del computador, para poder seleccionar un portal cautivo por software que más convenga.

A continuación se muestra una lista con los portales cautivos por software más representativos

LINUX:

- PepperSpoT
- NoCatAuth
- Chillispot
- CoovaChilli
- WifiDog
- AirMarshal
- ZeroShell

WINDOWS:

- Antamedia HotSpot
- WifiDog
- FirstSpot

FreeBSD:

- m0n0wall
- PfSense

A continuación se describen los principales portales cautivos por software:

1.1.4.1 ChilliSpot[18]

ChilliSpot es un portal cautivo de código abierto para una red inalámbrica o LAN. Es usado para autenticar a los usuarios de una red inalámbrica. Soporta un ingreso basado en Web. Authentication, Authorization y Accounting (AAA) es necesario para poder instalar ChilliSpot. Trabaja en diferentes distribuciones Linux: Redhat, Fedora, Debian, Mandrake y OpenWRT.

Los principales requisitos para que funcione un HotSpot son los siguientes:

- Conexión a Internet.
- Access Point.
- Software ChilliSpot.
- Servidor RADIUS.
- Servidor WEB.

En la Figura 1.6 se muestran los diferentes elementos que conforman una red con ChilliSpot.

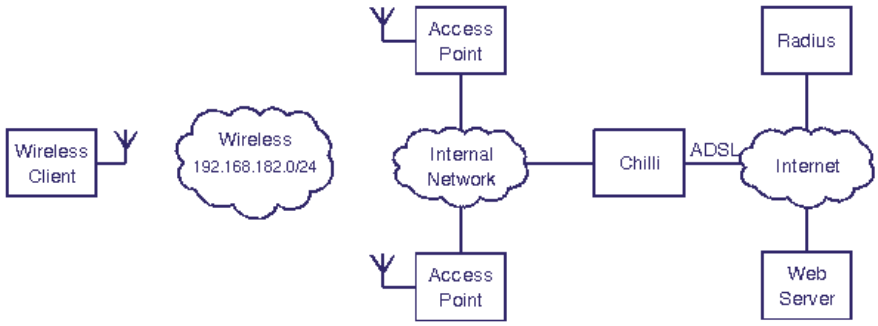


Figura 1.6 Ejemplo de una red con ChilliSpot

[18] http://www.chillispot.info/

No es necesario tener el servidor RADIUS y el Servidor WEB en diferentes computadores, se los puede tener instalados y configurados en la misma PC de ChilliSpot.

1.1.4.2 Antamedia HotSpot[19]

Antamedia HotSpot es un programa que permite controlar y cobrar por acceso al Internet. Es adecuado para todos los propósitos donde los clientes deben tener acceso rápido a Internet sin ninguna instalación de programas en sus computadores.

No requiere instalarse ningún programa en la PC del cliente. El programa asegura al HotSpot enrutando a todos los clientes a la pantalla de inicio de sesión.

Cualquier sistema operativo o navegador puede ser usado en la PC del cliente (Internet Explorer, Firefox, Opera, Safari, Konqueror).

Al conectarse a la red (usando tarjetas inalámbricas o cable UTP[20]), los clientes requieren ingresar el nombre de usuario y la clave para acceder al Internet. Los clientes verán su tiempo, uso de ancho de banda y podrán adicionar más tiempo si es necesario.

Es compatible con Windows, con una interfaz amigable. No requiere de ningún equipo adicional, de manera que puede ser usado en cualquier PC con dos tarjetas de red.

Los pagos se pueden realizar con tarjeta de crédito, el proceso de pago es completamente automatizado y el cliente puede escoger usuario, contraseña y plan (ejemplo, 1 hora de Internet o un mes de acceso ilimitado). No se necesita de un personal para vender tickets. Una interfaz simple de usar habilita el proceso de

[19] http://www.antamedia.com/hotspot/
[20] UTP: Par Trenzado No Apantallado *(Unshielded Twisted Pair*)

transacción con los mayores servicios de pago electrónico de Internet incluido PayPal (el cliente no necesita una cuenta PayPal).

Se pueden crear cuentas por adelantado e imprimirlas como tickets, se puede vender estos tickets en el mostrador, recepción o usando máquinas de venta, etc.

Se puede configurar los tickets para mostrar el logotipo de la empresa, nombre del HotSpot, encabezado y pie de página. Opcionalmente, se puede imprimir la fecha y hora en el ticket.

En la Figura 1.7 se muestra un ejemplo de diagrama de red usando Antamedia HotSpot.

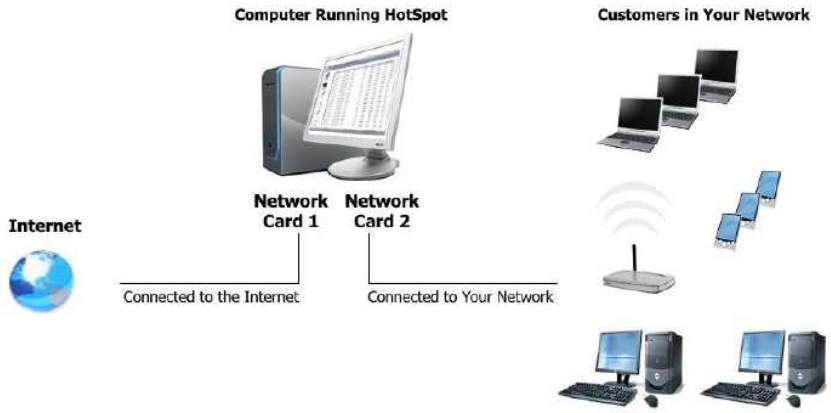


Figura 1.7 Ejemplo de una red con Antamedia HotSpot[21]

1.1.4.3 pfSense[22]

pfSense es una distribución BSD (*Berkeley Software Distribution*) basada en FreeBSD (*Free Berkeley Software Distribution*) y pensada para utilizarse como un router con firewall. Hasta ahora, con más de 1 millón de descargas, pfSense logra un lugar dentro del gusto de los usuarios BSD.

[21] http://www.antamedia.com/manuals/hotspot/topology2.htm
[22] http://www.pfsense.org/

Una vez instalado pfSense, se debe activar el servicio Captive Portal, para así poder configurar la forma en que los usuarios de una red entran a navegar por Internet (Figura 1.8)

Figura 1.8 Activación del Portal Cautivo en pfSense

El portal cautivo admite desde sencillas configuraciones, donde sólo aparece una página de información al usuario, hasta distintos sistemas de validación.

1.2 REDES DE ÁREA LOCAL INALÁMBRICAS (WLAN)

1.2.1 DEFINICIÓN[23]

WLAN (Wireless Local Area Network) es un sistema de comunicación de datos inalámbrico flexible, que ofrece todas las características y beneficios de las tecnologías LAN tradicionales, muy utilizado como complemento de éstas (no reemplazan soluciones cableadas). Utilizan radio frecuencia, siendo este el medio más popular, sobre todo si se trata de obtener un gran ancho de banda y una mayor cobertura.

[23] "Materia Redes Inalámbricas de Área Local, Capítulo 1 INTRODUCCIÓN, Características de las WLANs", Ing. Soraya Sinche, 2009.

El objetivo principal de las WLANs es el brindar conectividad en lugares de difícil acceso a tecnologías cableadas, ofreciendo además flexibilidad para realizar cambios, tales como movimiento de dispositivos o extensiones de red.

1.2.2 IEEE 802.11 (Wi-Fi)[24]

1.2.2.1 Introducción

La especificación IEEE 802.11 es un estándar internacional que define las características de una red de área local inalámbrica (WLAN) en los dos niveles inferiores del modelo OSI (capa física y de enlace de datos).

- La capa física (a veces abreviada capa "PHY[25]") ofrece cuatro tipos de codificación de información.
    - Tres técnicas de radio frecuencia: FHSS[26], DSSS[27] y OFDM[28].
    - Una especificación para infrared difusa DFIR[29]
- La capa de enlace de datos compuesta por dos subcapas: control de enlace lógico (LLC) y control de acceso al medio (MAC).

La capa física define las características y métodos de transmisión y recepción de datos a través del medio inalámbrico entre 2 o más estaciones, mientras que la capa de enlace de datos se encarga de describir como se empacan y verifican los bits de modo que no tengan error, en particular un método de acceso parecido al utilizado en el estándar Ethernet, y las reglas para la comunicación entre las estaciones de la red.

[24] http://www.ieee802.org/11/
[25] PHY: Capa Física (*PHysical laYer*)
[26] FHSS: Espectro ensanchado por salto de frecuencia (*Frequency-Hopping Spread Spectrum*)
[27] DSSS: Espectro ensanchado por secuencia directa (*Direct Sequence Spread Spectrum*)
[28] OFDM: Multiplexación por División de Frecuencias Ortogonales (*Orthogonal Frequency Division Multiplexing)*
[29] DFIR: Infrarojo Difuso (*Diffuse Infrared*)

En la Figura 1.9 se muestra un diagrama descriptivo de la capa física del 802.11 y sus extensiones.

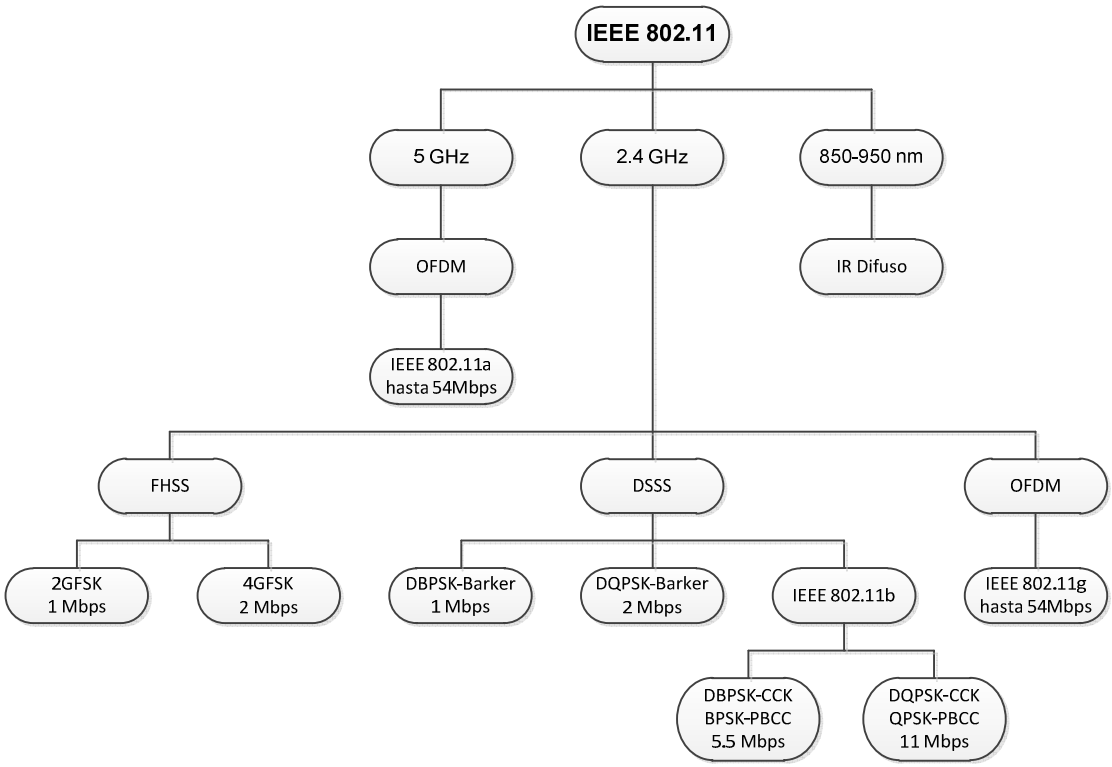


Figura 1.9 Diagrama descriptivo de la capa física del 802.11 y sus extensiones[30]

Wi-Fi (Wireless-Fidelity)[31] es el nombre de la certificación otorgada por la Wi-Fi Alliance, anteriormente WECA (Wireless Ethernet Compatibility Alliance), grupo que garantiza la compatibilidad entre dispositivos que utilizan el estándar 802.11. Por el uso indebido de los términos (y por razones de marketing) el nombre del estándar se confunde con el nombre de la certificación. Una red Wi-Fi es en realidad una red que cumple con el estándar 802.11. A los dispositivos certificados por la Wi-Fi Alliance se les permite usar el logotipo indicado en la Figura 1.10

Figura 1.10 Logotipo Wi-Fi

[30] http://agora.ya.com/biblio81/wireless/802-11-PHY.pdf
[31] http://www.wi-fi.org/discover_and_learn.php

Con Wi-Fi se pueden crear redes de área local inalámbricas de alta velocidad siempre y cuando el equipo que se vaya a conectar no esté muy alejado del punto de acceso. En la práctica, Wi-Fi admite computadores portátiles, equipos de escritorio, asistentes digitales personales (PDA) o cualquier otro tipo de dispositivo de alta velocidad con propiedades de conexión también de alta velocidad (54 Mbps o superior) dentro de un radio de varias docenas de metros en ambientes cerrados (de 20 a 50 metros en general) o dentro de un radio de cientos de metros al aire libre. Los proveedores de Wi-Fi están cubriendo áreas con una gran concentración de usuarios (como estaciones de trenes, aeropuertos y hoteles) con redes inalámbricas.

1.2.2.2 Estándares IEEE 802.11[32]

El estándar 802.11 en realidad es el primer estándar y permite velocidades de transmisión de 1 a 2 Mbps. El estándar original se ha modificado para optimizar el ancho de banda (incluidos los estándares 802.11a, 802.11b, 802.11g y 802.11n denominados estándares físicos 802.11) o para especificar componentes de mejor manera con el fin de garantizar mayor seguridad o compatibilidad.

La Tabla 1.1 indica las distintas modificaciones del estándar 802.11 y sus significados:

Estándar	Descripción
802.11	Control de Acceso al Medio CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Fue creado para operar en la banda de 2.4 GHz y con velocidades de transmisión de 1 y 2 Mbps. Define 3 tipos de medios físicos: • DFIR (Difuse Infrared). • FHSS (Frequency-Hopping Spread Spectrum). • DSSS (Direct Sequence Spread Spectrum).

[32] http://standards.ieee.org

802.11a	Opera en la banda de 5 GHz y utiliza 52 subportadoras OFDM (Orthogonal Frequency Division Multiplexing) permitiendo velocidades de 6, 9, 12, 18, 24, 36, 48 y 54 Mbps.
802.11b	Opera en la banda de 2.4 GHz y utiliza el mismo método de acceso definido en el estándar original CSMA/CA, permitiendo velocidades de 5.5 y 11 Mbps.
802.11c	Es una combinación del 802.11 y el 802.11d. El estándar combinado 802.11c no ofrece ningún interés para el público general. Es solamente una versión modificada del estándar 802.11d que permite combinar el 802.11d con dispositivos compatibles 802.11 (en el nivel de enlace de datos).
802.11d	Es un complemento del estándar 802.11 que está pensado para permitir el uso internacional de las redes 802.11 locales. Permite que distintos dispositivos intercambien información en rangos de frecuencia según lo que se permite en el país de origen del dispositivo.
802.11e	Está destinado a mejorar la calidad del servicio en el nivel de la capa de enlace de datos. El objetivo del estándar es definir los requisitos de diferentes paquetes en cuanto al ancho de banda y al retardo de transmisión para permitir mejores transmisiones de audio y vídeo.
802.11f	Es una recomendación para proveedores de puntos de acceso que permite que los productos sean más compatibles. Utiliza el protocolo IAPP que le permite a un usuario itinerante cambiarse claramente de un punto de acceso a otro mientras está en movimiento sin importar qué marcas de puntos de acceso se usan en la infraestructura de la red.
802.11g	Trabaja en las mismas velocidades de IEEE 802.11a y otras velocidades. Mantiene la banda de 2.4 GHz y es compatible con 802.11b.
802.11h	Tiene por objeto resolver problemas derivados a la coexistencia del estándar 802.11 con el estándar europeo (HiperLAN 2, de ahí la h

	de 802.11h) y cumplir con las regulaciones europeas relacionadas con el uso de las frecuencias y el rendimiento energético.
802.11i	Está destinado a mejorar la seguridad en la transferencia de datos (al implementar el cifrado y la autenticación). Este estándar se basa en AES (estándar de cifrado avanzado) y puede cifrar transmisiones que se ejecutan en las tecnologías 802.11a, 802.11b y 802.11g.
802.11j	Es para la regulación japonesa lo que el 802.11h es para la regulación europea.
802.11k	Permite a los conmutadores y puntos de acceso inalámbricos calcular los recursos de radiofrecuencia de los clientes de una red WLAN, mejorando así su gestión. Está diseñado para ser implementado en software.
802.11n	La velocidad real de transmisión podría llegar a los 600 Mbps (lo que significa que las velocidades teóricas de transmisión serían aún mayores), y debería ser hasta 10 veces más rápida que una red bajo los estándares 802.11a y 802.11g, y unas 40 veces más rápida que una red bajo el estándar 802.11b. También se espera que el alcance de operación de las redes sea mayor con este nuevo estándar gracias a la tecnología MIMO (Multiple Input – Multiple Output), que permite utilizar varios canales a la vez para enviar y recibir datos gracias a la incorporación de varias antenas (3). Puede trabajar en dos bandas de frecuencias: 2,4 GHz (la que emplean 802.11b y 802.11g) y 5 GHz (la que usa 802.11a). Gracias a ello, 802.11n es compatible con dispositivos basados en todas las ediciones anteriores de Wi-Fi.
802.11p	Este estándar opera en el espectro de frecuencias de 5.9 GHz, especialmente indicado para automóviles. Será la base de las comunicaciones dedicadas de corto alcance (DSRC) en Norteamérica. La tecnología DSRC permitirá el intercambio de datos entre vehículos y entre automóviles e infraestructuras en carretera.
802.11r	También se conoce como Fast Basic Service Set Transition, y su principal característica es permitir a la red que establezca los

	protocolos de seguridad que identifican a un dispositivo en el nuevo punto de acceso antes de que abandone el actual y se pase a él.
802.11s	Define la interoperabilidad de fabricantes en cuanto a protocolos Mesh (son aquellas redes en las que se mezclan las dos topologías de las redes inalámbricas, la topología Ad-hoc y la topología infraestructura.). Bien es sabido que no existe un estándar, y que por eso cada fabricante tiene sus propios mecanismos de generación de mallas.
802.11v	Servirá para permitir la configuración remota de los dispositivos cliente. Esto permitirá una gestión de las estaciones de forma centralizada (similar a una red celular) o distribuida, a través de un mecanismo de capa 2. Esto incluye, por ejemplo, la capacidad de la red para supervisar, configurar y actualizar las estaciones cliente.
802.11w	Todavía no concluido. Permite mejorar la capa del control de acceso del medio de IEEE 802.11 para aumentar la seguridad de los protocolos de autenticación y codificación. Se intenta extender la protección que aporta el estándar 802.11i más allá de los datos hasta las tramas de gestión, responsables de las principales operaciones de una red.
802.11y	Permite operar en la banda de 3650 a 3700 MHz (excepto cuando pueda interferir con una estación terrestre de comunicaciones por satélite) en EEUU, aunque otras bandas en diferentes dominios reguladores también se están estudiando.

Tabla 1.1 Estándares 802.11[33]

También es importante mencionar la existencia de unos productos llamados 802.11G+ (802.11G Turbo), es una tecnología propietaria de la empresa Atheros que mejora el rendimiento de las redes IEEE 802.11g con una banda de 2.4 GHz, alcanza una velocidad de transferencia de 108 Mbps.

[33] http://standards.ieee.org/

1.2.2.3 802.11a[34]

Opera en la banda de 5 GHz designada por la UNII (Unlicensed National Information Infraestructure) y utiliza 12 canales no superpuestos (Capacidad total = 432 Mbps). Soporta tasa de transferencia de 6 a 54 Mbps. Utiliza como técnica de modulación OFDM.

1.2.2.4 802.11b[35]

Es una extensión del estándar IEEE 802.11 con un esquema DSSS. Opera en la banda de 2.4 GHz y utiliza tres canales no superpuestos (Capacidad total = 33 Mbps). Se dispone de tres tipos diferentes de modulación, que dependen de la tasa de velocidad máxima: BPSK → 1 Mbps, QPSK → 2 Mbps y CCK[36] → 5.5 y 11 Mbps.

1.2.2.5 802.11g[37]

Opera en la banda de 2.4 GHz Soporta tasa de transferencia de 54 Mbps. Utiliza como técnica de modulación OFDM (Orthogonal Frequency Division Multiplexing). Compatible con: 802.11, 802.11b y 802.11n.

1.2.2.6 Modos de Operación

El estándar 802.11 define dos modos de operación:

- El modo de infraestructura en el que los clientes de tecnología inalámbrica se conectan a un punto de acceso.

[34] "Materia Redes Inalámbricas de Área Local, Capítulo 2 EL ESTÁNDAR IEEE802.11 PARA WLAN, Capa Física IEEE 802.11a", Ing. Soraya Sinche, 2009.
[35] "Materia Redes Inalámbricas de Área Local, Capítulo 2 EL ESTÁNDAR IEEE802.11 PARA WLAN, Capa Física IEEE 802.11b", Ing. Soraya Sinche, 2009.
[36] CCK: Modulación por Código Complementario (*Complementary Code Keying*)
[37] "Materia Redes Inalámbricas de Área Local, Capítulo 2 EL ESTÁNDAR IEEE802.11 PARA WLAN, Capa Física IEEE 802.11g", Ing. Soraya Sinche, 2009.

- El modo ad-hoc en el que los clientes se conectan entre sí sin ningún Punto de Acceso (AP).

1.2.2.6.1 Modo de Infraestructura[38]

En el modo de infraestructura, cada estación de trabajo se conecta a un Punto de Acceso a través de un enlace inalámbrico, como se indica en la Figura 1.11

Figura 1.11 BSS con AP

La configuración formada por el Punto de Acceso y las estaciones ubicadas dentro del área de cobertura se llama conjunto de servicio básico o BSS. Cada BSS se identifica a través de un BSSID (identificador de BSS) que es un identificador de 6 bytes (48 bits). En el modo infraestructura el BSSID corresponde a la dirección MAC del Punto de Acceso.

Es posible vincular varios Puntos de Acceso juntos (o con más exactitud, varios BSS) con una conexión llamada sistema de distribución (o SD) para formar un conjunto de servicio extendido o ESS. El sistema de distribución también puede ser una red conectada, un cable entre dos Puntos de Acceso o incluso una red inalámbrica. En las Figuras 1.12 y 1.13 se indica un conjunto de servicio extendido (ESS), en donde el sistema de distribución (SD) es inalámbrico y alámbrico respectivamente.

[38] http://es.kioskea.net/contents/wifi/wifimodes.php3

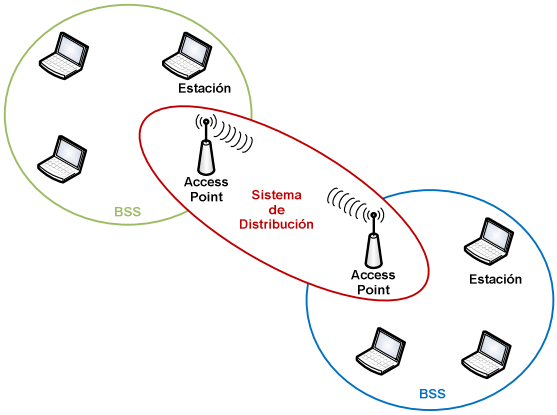


Figura 1.12 ESS con SD inalámbrico

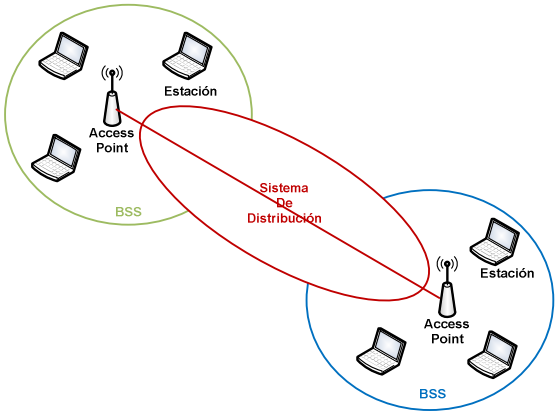


Figura 1.13 ESS con SD cableado

Un ESS se identifica a través de un ESSID (identificador del conjunto de servicio extendido), que es un identificador de 32 caracteres en formato ASCII que actúa como su nombre en la red. El ESSID, a menudo abreviado SSID, muestra el nombre de la red y de alguna manera representa una medida de seguridad de primer nivel ya que una estación debe conocer el SSID para conectarse a la red extendida.

Cuando un usuario va desde un BSS a otro mientras se mueve dentro del ESS, el adaptador de la red inalámbrica de su equipo puede cambiarse de punto de acceso, según la calidad de la señal que reciba desde distintos Puntos de Acceso. Los Puntos de Acceso se comunican entre sí a través de un sistema de distribución con el fin de intercambiar información sobre las estaciones y, si es necesario, para transmitir datos desde estaciones móviles. Esta característica que permite a las estaciones moverse de forma transparente de un Punto de Acceso al otro se denomina itinerancia.

1.2.2.6.2 Comunicación con el Access Point[39]

Cuando una estación se asocia a un BSS Wi-Fi, envía una solicitud de sondeo a cada canal. Esta solicitud contiene el ESSID que el BSS está configurado para usar y también el volumen de tráfico que su adaptador inalámbrico puede admitir.

[39] http://es.kioskea.net/contents/wifi/wifimodes.php3

Si no se establece ningún ESSID, la estación escucha a la red para encontrar un SSID.

Cada Punto de Acceso transmite una señal en intervalos regulares (diez veces por segundo aproximadamente). Esta señal, que se llama señalización, provee información de su BSSID, sus características y su ESSID, si corresponde. El ESSID se transmite automáticamente en forma predeterminada, pero se recomienda que si es posible se deshabilite esta opción.

Cuando se recibe una solicitud de sondeo, el Punto de Acceso verifica el ESSID y la solicitud del volumen de tráfico encontrado en la señalización. Si el ESSID dado concuerda con el del Punto de Acceso, éste envía una respuesta con datos de sincronización e información sobre su carga de tráfico. Así, la estación que recibe la respuesta puede verificar la calidad de la señal que envía el Punto de Acceso para determinar cuán lejos está.

En términos generales, mientras más cerca un Punto de Acceso esté, más grande será su tasa de transferencia de datos. Por lo tanto, una estación dentro del rango de muchos Puntos de Acceso (que tengan el mismo SSID) puede elegir el punto que ofrezca la mejor tasa de transferencia de datos.

1.2.2.6.3 Modo Ad-Hoc[40]

En el modo ad hoc los equipos inalámbricos se conectan entre sí para formar una red punto a punto, es decir, una red en la que cada equipo actúa como cliente y como Punto de Acceso simultáneamente, como se indica en la Figura 1.14

[40] http://es.kioskea.net/contents/wifi/wifimodes.php3

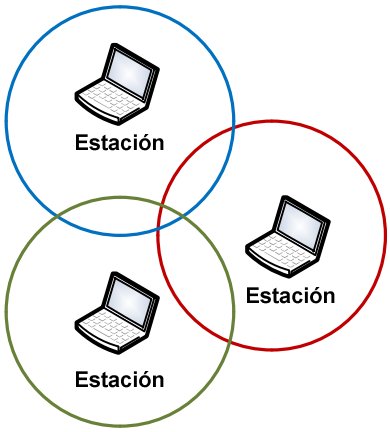


Figura 1.14 BSS sin AP

La configuración que forman las estaciones se llama conjunto de servicio básico independiente o IBSS. Un IBSS es una red inalámbrica que tiene al menos dos estaciones y no usa ningún punto de acceso. Por eso, el IBSS crea una red temporal que le permite a la gente que esté en el mismo lugar, intercambiar datos. Se identifica a través de un SSID de la misma manera en que lo hace un ESS en el modo infraestructura.

En una red ad hoc, el rango del BSS está determinado por el rango de cada estación. Esto significa que si dos estaciones de la red están fuera del rango de la otra, no podrán comunicarse, ni siquiera cuando puedan ver otras estaciones.

A diferencia del modo infraestructura, el modo ad hoc no tiene un sistema de distribución que pueda enviar tramas de datos desde una estación a la otra. Entonces, por definición, un IBSS es una red inalámbrica restringida.

1.2.3 SEGURIDAD[41]

Las ondas de radio tienen la posibilidad de propagarse en todas las direcciones dentro de un rango relativamente amplio. Es por esto que es muy difícil mantener las transmisiones de radio dentro de un área limitada, permitiendo así que las

[41] http://es.kioskea.net/contents/wifi/wifirisques.php3

ondas puedan pasar de un piso a otro en un edificio, con un alto grado de atenuación.

La consecuencia principal de esta propagación de ondas es que personas no autorizadas pueden escuchar la red, posiblemente más allá del confinamiento del edificio donde se ha establecido la red inalámbrica.

Se puede instalar una red inalámbrica muy fácilmente en una compañía sin que se entere el administrador de la red, un empleado sólo tiene que conectar un Punto de Acceso con un puerto de datos.

Existen muchos riesgos que surgen al no asegurar una red inalámbrica de manera adecuada, entre los principales se tiene:

- La intercepción de datos, es la práctica que consiste en escuchar las transmisiones de varios usuarios de una red inalámbrica. Una red inalámbrica es insegura de manera predeterminada, esto significa que está abierta a todos y cualquier persona dentro del área de cobertura del punto de acceso puede potencialmente escuchar las comunicaciones que se envían en la red. En el caso de una persona, la amenaza no es grande ya que los datos raramente son confidenciales, a menos que se trate de datos personales. Sin embargo, si se trata de una compañía, esto puede plantear un problema serio.
- El intento de acceder a la red local o a Internet (crackeo). La instalación de un Punto de Acceso en una red local permite que cualquier estación acceda a la red conectada y también a Internet, si la red local está conectada a ella. Es por esto que una red inalámbrica insegura les ofrece a los hackers la puerta de acceso perfecta a la red interna de una compañía u organización. Además de permitirle al hacker robar o destruir información de la red y de darle acceso a Internet gratuito, la red inalámbrica también puede inducirlo a llevar a cabo ataques cibernéticos. Como no existe manera de identificar al hacker en una red, puede que se responsabilice del ataque a la compañía que instaló la red inalámbrica.

- La interferencia de transmisión, las ondas radiales son muy sensibles a la interferencia. Por ello una señal se puede interferir fácilmente con una transmisión de radio que tenga una frecuencia cercana a la utilizada por la red inalámbrica. Hasta un simple horno microondas puede hacer que una red inalámbrica se vuelva inoperable si se está usando dentro del rango del Punto de Acceso.
- Los ataques de denegación de servicio. El método de acceso a la red del estándar 802.11 se basa en el protocolo CSMA/CA[42], que consiste en esperar hasta que la red esté libre antes de transmitir las tramas de datos. Una vez que se establece la conexión, una estación se debe vincular a un Punto de Acceso para poder enviarle paquetes. Debido a que los métodos para acceder a la red y asociarse a ella son conocidos, un hacker puede fácilmente enviar paquetes a una estación solicitándole que se desvincule de una red. El envío de información para afectar una red inalámbrica se conoce como ataque de denegación de servicio.

1.2.3.1 Medidas de Seguridad

1.2.3.1.1 Infraestructura Adecuada[43]

Lo primero que hay que hacer cuando se instala una red inalámbrica es ubicar el Punto de Acceso en un lugar razonable dependiendo del área de cobertura que se desee. Se debe tener en cuenta si el Punto de Acceso se va a instalar en exteriores o interiores, dependiendo de ello se tendrá un rango de cobertura. El uso de antenas con mayor ganancia aumentará considerablemente la cobertura. Se debe tener en cuenta además, el material con el que están construidos los edificios, ya que ciertos materiales reflejan las señales sin problema como la madera, lo cual puede extender la cobertura, en cambio otros materiales como el concreto con varilla, acero y cemento absorben y atenúan la potencia de la señal disminuyendo la cobertura.

[42] CSMA/CA: Método de acceso múltiple por detección de portadora con anulación de colisiones (*Carrier Sense Multiple Access/Collision Avoidance*).
[43] http://es.kioskea.net/contents/wifi/wifisecu.php3

1.2.3.1.2 Evitar el uso de valores predeterminados

Cuando se instala un Punto de Acceso por primera vez, se configura con ciertos valores predeterminados, inclusive la contraseña del administrador. Muchos administradores principiantes suponen que como la red ya está funcionando, no tiene sentido cambiar la configuración del Punto de Acceso. Sin embargo, las configuraciones predeterminadas brindan sólo un nivel de seguridad mínimo. Por esta razón, es necesario conectarse al dispositivo (casi siempre a través de una interfaz Web o al usar un puerto en particular en el punto de acceso) para cambiar la contraseña administrativa.

1.2.3.1.3 Filtrado SSID[44]

Además, para conectarse a un Punto de Acceso es necesario conocer el identificador de red (SSID). Por ello se recomienda cambiar el nombre predeterminado de la red y desactivar la transmisión del nombre en la red. Cambiar el identificador de red predeterminado es muy importante, ya que de lo contrario puede brindarles a los hackers información sobre la marca o el modelo del punto de acceso que se está usando.

1.2.3.1.4 Filtrado MAC[45]

Todo adaptador de red tiene su propia dirección física que se denomina dirección MAC. Esta dirección está representada por 12 dígitos en formato hexadecimal divida en grupos de dos dígitos separados por guiones.

Las interfaces de configuración de los Puntos de Acceso les permiten, por lo general, mantener una lista de control de acceso llamada ACL que se basa en las direcciones MAC de los dispositivos autorizados para conectarse a la red inalámbrica.

---

[44] "Materia Redes Inalámbricas de Área Local, Capítulo 3 SEGURIDAD EN REDES INALÁMBRICAS, Medidas de Seguridad en Redes Inalámbricas: Filtrado SSID", Ing. Soraya Sinche, 2009.

[45] "Materia Redes Inalámbricas de Área Local, Capítulo 3 SEGURIDAD EN REDES INALÁMBRICAS, Filtrado MAC" Ing. Soraya Sinche, 2009.

Esta precaución algo restrictiva le permite a la red limitar el acceso a un número dado de equipos. Sin embargo, esto no soluciona el problema de la seguridad en las transferencias de datos.

1.2.3.1.5 WEP[46]

Para solucionar los problemas de seguridad de transferencia en redes inalámbricas, el estándar 802.11 incluye el protocolo de seguridad WEP (Wired Equivalent Protocol). Este protocolo provee mecanismos de encripción y autenticación por medio de un mecanismo de clave compartida.

WEP es un protocolo de cifrado de trama de datos 802.11 que utiliza el algoritmo simétrico RC4 con claves de 64 bits o 128 bits. La clave de 64 bits se genera a partir de una clave estática de forma automática, aunque existe la posibilidad de introducir esta clave de forma manual. Esta clave secreta se debe declarar tanto en el punto de acceso como en los equipos cliente. A partir de la clave estática se generan 4 claves en función de si se van a utilizar claves de 64 o 128 bits, se generan claves de 40 y 104 bits respectivamente. Los 24 bits restantes se añaden a un vector de inicialización. De las 4 claves generadas se selecciona solo una de ellas para la encripción WEP.

En el caso de una clave de 40 bits, con un ataque de fuerza bruta (que prueba todas las claves posibles) un hacker puede encontrar la clave de sesión con rapidez. Por lo tanto, WEP no es suficiente para garantizar verdaderamente la privacidad de los datos.

Sin embargo, se recomienda utilizar al menos una clave WEP de 128 bits para garantizar un nivel de privacidad mínimo.

[46] "Materia Redes Inalámbricas de Área Local, Capítulo 3 SEGURIDAD EN REDES INALÁMBRICAS, WEP", Ing. Soraya Sinche, 2009.

1.2.3.1.6 IEEE 802.1X y EAP[47]

El estándar 802.1x es una solución de seguridad ratificada por el IEEE en junio de 2001 que puede autenticar (identificar) a un usuario que quiere acceder a la red (ya sea por cable o inalámbrica). Esto se hace a través del uso de un servidor de autenticación. El estándar 802.1x se basa en el protocolo EAP (Extensible Authentication Protocol). Este protocolo se usa para transportar la información de identificación del usuario.

EAP se basa en el uso de un controlador de acceso llamado autenticador, que le otorga o deniega a un usuario el acceso a la red. El usuario en este sistema se llama solicitante. El controlador de acceso es un firewall básico que actúa como intermediario entre el usuario y el servidor de autenticación, y que necesita muy pocos recursos para funcionar. Cuando se trata de una red inalámbrica, el punto de acceso actúa como autenticador.

El servidor NAS (Servicio de autenticación de red o Servicio de acceso a la red) redirige la petición al servidor RADIUS, el cual puede aprobar la identidad del usuario transmitida por el controlador de la red y otorgarle acceso según sus credenciales. Además, este tipo de servidor puede almacenar y hacer un seguimiento de la información relacionada con los usuarios. En el caso de un proveedor de servicio, por ejemplo, estas características le permiten al servidor facturarles en base a cuánto tiempo estuvieron conectados o cuántos datos transfirieron.

Generalmente el servidor de autenticación es un servidor RADIUS (Uses Remote Authentication Dial-In User Service), un servidor de autenticación estándar, pero puede utilizarse cualquier otro servicio de autenticación en su lugar.

A continuación encontrará un resumen sobre cómo funciona una red segura que usa el estándar 802.1x[48]:

[47] "Materia Redes Inalámbricas de Área Local, Capítulo 3 SEGURIDAD EN REDES INALÁMBRICAS, IEEE 802.1x y EAP", Ing. Soraya Sinche, 2009.

1. El Punto de Acceso, después de recibir la solicitud de conexión del usuario, envía una solicitud de autenticación.
2. El usuario envía una respuesta al controlador de acceso, quien enruta la respuesta al servidor de autenticación.
3. El servidor de autenticación envía un challenge (desafio) al controlador de acceso, quien lo transmite al usuario. El challenge es un método para establecer la identificación.
4. El usuario responde al challenge (desafío). Si la identidad del usuario es correcta, el servidor de autenticación envía la aprobación al controlador de acceso, quien le permite al usuario ingresar a la red o a parte de ella, según los derechos otorgados.
5. Si no se pudo verificar la identidad del usuario, el servidor de autenticación envía un mensaje de denegación y el controlador de acceso le deniega al usuario el acceso a la red.

1.2.3.1.7 WPA[49]

WPA (Wi-Fi Protected Access) es una solución de seguridad inalámbrica ofrecida por Wi-Fi Alliance para solucionar las carencias de WEP. WPA es una versión del protocolo 802.11i que depende de protocolos de autenticación y de un algoritmo de cifrado cerrado: TKIP (Temporal Key Integrity Protocol), el cual genera claves aleatorias.

El funcionamiento de WPA se basa en la implementación de un servidor de autenticación (en general un servidor RADIUS) que identifica a los usuarios en una red y establece sus privilegios de acceso. No obstante, redes pequeñas pueden usar una versión más simple de WPA, llamada WPA-PSK, al implementar la misma clave de cifrado en todos los dispositivos, con lo cual ya no se necesita el servidor RADIUS.

---

[48] http://es.kioskea.net/contents/wifi/wifi-802.1x.php3

[49] "Materia Redes Inalámbricas de Área Local, Capítulo 3 SEGURIDAD EN REDES INALÁMBRICAS, WPA (Wi-Fi Protected Access)", Ing. Soraya Sinche, 2009.

WPA en su primera construcción sólo admite redes en modo infraestructura, es decir que no se puede utilizar para asegurar redes punto a punto (modo ad-hoc).

1.2.3.1.8 802.11i[50]

El estándar 802.11i se ratificó el 24 de junio de 2004 para abordar el problema de la seguridad en redes inalámbricas. Se basa en el algoritmo de cifrado TKIP, como el WEP, pero también admite el AES (Advanced Encryption Standard), el cual es un algoritmo de cifrado que soporta claves de 128, 192 y 256 bits. Wi-Fi Alliance creó una nueva certificación, denominada WPA2, para dispositivos que admiten el estándar 802.11i (como computadores portátiles, PDA, tarjetas de red, etc.). A diferencia del WPA, el WPA2 puede asegurar redes inalámbricas en modo infraestructura como también redes en modo ad hoc.

IEEE 802.11i define dos modos de operación:

- **WPA-Personal**: Este modo permite la implementación de una infraestructura segura basada en WPA sin tener que utilizar un servidor de autenticación. WPA Personal se basa en el uso de una clave compartida, llamada PSK, que se almacena en el Punto de Acceso y en los dispositivos cliente. A diferencia del WEP, no se necesita ingresar una clave de longitud predefinida. El WPA le permite al usuario ingresar una frase de contraseña, después, un algoritmo la convierte en PSK.
- **WPA-Enterprise**: Este modo requiere de una infraestructura de autenticación 802.1x con un servidor de autenticación, generalmente un servidor RADIUS.

[50] "Materia Redes Inalámbricas de Área Local, Capítulo 3 SEGURIDAD EN REDES INALÁMBRICAS, IEEE 802.11i"; Ing. Soraya Sinche, 2009.

1.3 NETFILTER / IPTABLES[51]

1.3.1 INTRODUCCIÓN

Netfilter es un componente disponible en el núcleo Linux (desde la versión 2.4) que permite interceptar y manipular paquetes de red. Netfilter es también el nombre que recibe el proyecto que se encarga de ofrecer herramientas libres para cortafuegos basados en Linux.

Netfilter hace posible separar las operaciones sobre los paquetes en tres partes:

- Filtrado de Paquetes.
- Traducción de Direcciones (NAT).
- Seguimiento de conexiones.

El componente más popular construido sobre Netfilter es iptables, una herramienta que hace uso de la infraestructura que ofrece Netfilter, que permite:

- Filtrar Paquetes.
- Traducción de direcciones (NAT).
- Mantener registros de logs.
- Usar capacidades de seguimiento de conexiones NETFILTER para definir los denominados firewalls stateful (firewalls con estado).

Iptables reemplaza a herramientas (programas) muy usados para crear firewalls en Linux: ipchains (núcleo Linux 2.2) e ipfwadmin (núcleo Linux 2.0) que a su vez se basaba en ipfw de BSD. Tanto ipchains como ipfwadm alteran el código de red para poder manipular los paquetes.

Las posibles tareas sobre los paquetes se controlan mediante distintos conjuntos de reglas, en función de la situación o momento en la que se encuentre un

[51] http://www.netfilter.org/

paquete durante su procesamiento. Dichos conjuntos de reglas y demás datos residen en el espacio de memoria del kernel. Esto significa que cualquier regla que se establezca, se perderá cuando reinicie. Pare evitar esto, iptables provee otras utilidades que permiten guardar y recuperar reglas en memoria: iptables-save e iptables-restore.

En la mayoría de los sistemas Linux, iptables está instalado como /sbin/iptables. En la práctica un cortafuego (firewall) iptables consistirá en un script conteniendo los comandos iptables para configurar convenientemente las listas de reglas.

Típicamente este script residirá en el directorio /etc/init.d o en /etc/rc.d para que sea ejecutado cada vez que arranca el sistema.

Iptables es un software disponible en prácticamente todas las distribuciones de Linux actuales. Existe en la actualidad una versión de iptables para IPv6, llamada ip6tables al igual que la herramienta de administración.

1.3.2 ELEMENTOS

Iptables permite al administrador del sistema definir reglas acerca de qué hacer con los paquetes de red. Las reglas se agrupan en cadenas: cada cadena es una lista ordenada de reglas. Las cadenas se agrupan en tablas: cada tabla está asociada con un tipo diferente de procesamiento de paquetes.

1.3.2.1 Tablas[52]

Corresponden a los distintos tipos de procesamiento que se pueden aplicar sobre los paquetes. Es posible crear nuevas tablas mediante módulos de extensión. El administrador puede crear y eliminar cadenas definidas por usuarios dentro de cualquier tabla. Inicialmente, todas las cadenas están vacías y tienen una política de destino que permite que todos los paquetes pasen sin ser bloqueados o

[52] http://es.wikipedia.org/wiki/Netfilter/iptables

alterados. Hay tres tablas ya incorporadas, cada una de las cuales contiene ciertas cadenas predefinidas.

- **filter table**. Controla decisiones de filtrado de paquetes, bloquea o permite que un paquete continúe su camino.
  Cadenas: INPUT, OUTPUT, FORWARD.
- **nat table**. Esta tabla es la responsable de configurar las reglas de reescritura de direcciones o de puertos de los paquetes.
  Cadenas: PREROUTING, POSTROUTING, OUTPUT.
- **mangle table**. Controla los procesos de modificación del contenido y las opciones de los paquetes.
  Cadenas: INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING.

1.3.2.2 Cadenas[53]

Contienen las listas de reglas a aplicar sobre los paquetes.

- **INPUT chain**. Reglas a aplicar sobre los paquetes destinados a la propia máquina, justo antes de pasarlos a las aplicaciones. Usada para controlar las entradas al propio equipo (cortafuegos).
- **OUTPUT chain**. Reglas a aplicar sobre los paquetes originados en la propia máquina, justo después de recibirlas desde las aplicaciones. Usada para controlar las salidas de propio equipo (cortafuegos).
- **FORWARD chain**. Reglas a aplicar sobre los paquetes que atraviesan la máquina con destino a otras. Usadas en cortafuegos de borde.
- **PREROUTING chain**. Reglas a aplicar sobre paquetes justo antes de enviarlos a la red. Usada para: DNAT (destination NAT), redirección de puertos, balanceo de carga, proxy transparente.
- **POSTROUTING chain**. Reglas a aplicar sobre paquetes (propio o ajenos) recibidos de la red, antes de decidir a dónde encaminarlos (local o reenvío). Usada para: SNAT (source NAT), enmascaramiento.

[53] http://es.wikipedia.org/wiki/Netfilter/iptables

Se pueden crear cadenas definidas por el usuario, a las que se accederá desde reglas incluidas en alguna de las cadenas predeterminadas. Para ello se usa el comando:

```
iptables  -N  nombre_cadena
```

1.3.3 FUNCIONAMIENTO DE IPTABLES

En la Figura 1.15 se indica un diagrama de cómo están dispuestas las cadenas.

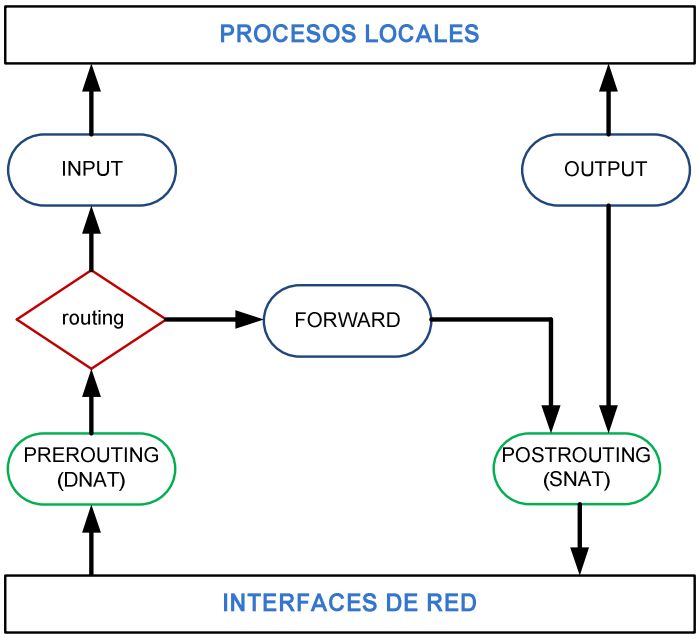


Figura 1.15 Diagrama de disposición de las cadenas[54]

Los círculos representan las 5 cadenas: INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING. Cuando un paquete alcanza un círculo del diagrama, se examina esa cadena para decidir la suerte del paquete. Si la cadena dice que hay que descartar (DROP) el paquete, se le elimina ahí mismo, pero si la cadena dice que hay que aceptarlo (ACCEPT), continúa su camino por el diagrama.

[54] http://www.pello.info/filez/IPTABLES_en_21_segundos.html

Dentro de cada cadena, las reglas se inspeccionan secuencialmente, el orden de las reglas es muy importante. Si el paquete no encaja con ninguna regla, se le aplica la política por defecto que se haya asignado a esa cadena, por defecto las cadenas predeterminadas están inicializadas con una política ACCEPT.

Cuando llega un paquete por la interfaz de red, pasan a través de la cadena PREROUTING antes de que se consulte la tabla de ruteo local (encaminamiento "routing"). Si está destinado al mismo dispositivo, el paquete entra en el diagrama hacia la cadena INPUT. Si pasa de aquí, cualquier proceso que esté esperando por el paquete, lo recibirá. En caso contrario, si el núcleo no tiene las capacidades de reenvío activadas, o no sabe hacia dónde reenviar el paquete, se descarta el paquete. Si está activado el reenvío, y el paquete está destinado a otra interfaz de red (si existe otra), entonces el paquete pasa directamente a la cadena FORWARD del diagrama. Si es aceptado, entonces saldrá del dispositivo. Antes de salir debe pasar por la cadena POSTROUTING. Finalmente, si un programa que se ejecuta en el dispositivo puede enviar paquetes de red. Estos paquetes pasan por la cadena OUTPUT de forma inmediata: si los acepta (ACCEPT), entonces el paquete continúa hacia afuera, sin antes pasar por la cadena POSTROUTING, dirigido a la interfaz a la que estuviera destinada.

1.3.4 ESQUEMA

A continuación se muestra la sintaxis detallada del comando iptables.

```
iptables -t tabla COMANDO CONDICIONES OBJETIVO
```

Si no se especifica una tabla, se usa por defecto *-t filter.*

1.3.4.1 Comandos más relevantes

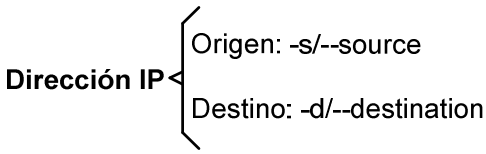
En la Tabla 1.2 se indica una descripción de los comandos básicos más usados en iptables:

Comando	Descripción
iptables -L/--list *cadena*	Lista las reglas actualmente en uso en una cadena.
iptables -F/--flush *cadena*	Vacía una cadena.
iptables -Z/--zero *cadena*	Reinicia los contadores de una cadena.
iptables -P/--policy *cadena* DROP/ACCEPT	Establece la política por defecto.
iptables -N/--new-chain *cadena*	Crea una nueva cadena.
iptables -X/--delete-chain *cadena*	Elimina una cadena.
iptables -A/--append *cadena*	Añade una regla (condiciones + objetivos) a una cadena
iptables -D/--delete *cadena*	Borra una regla de una cadena.
iptables -R/--replace *cadena*	Reemplaza una regla de una cadena.
iptables -I/--insert *cadena*	Inserta una regla de una cadena.
iptables --help	Accede a la página manual de iptables.

Tabla 1.2 Comandos Iptables

1.3.4.2 Condiciones[55]

Dirección IP: Para poder encaminar los paquetes a su destino final, cada interfaz de red necesita una dirección IP.



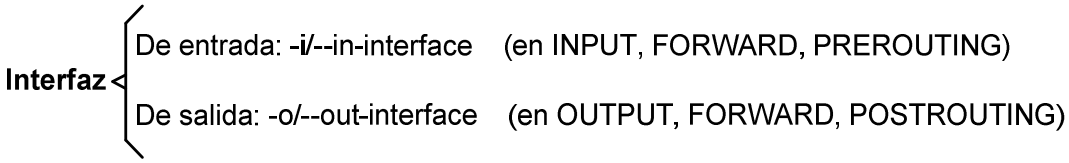
Las direcciones IP de origen (-s/--source/--src) y destino (-d/--destination/--dst) se pueden especificar de cuatro maneras:

- Una dirección IP (-s 192.168.10.25).
- Un nombre completo (-s www.google.com) (-s localhost).
- Un grupo de direcciones:

[55] http://www.netfilter.org/documentation/HOWTO/es/packet-filtering-HOWTO-7.html

- (-s 10.10.10.0/24)
- (-s 10.10.10.0/255.255.255.0).

Interfaz de Red: Las opciones –i/--in-interface y –o/--out-interface especifican el nombre de una interfaz con la que coincidir (eth0, eth1, ppp0,..., lo). Una interfaz es el dispositivo físico por el que entra (-i) o sale (-o) un paquete.



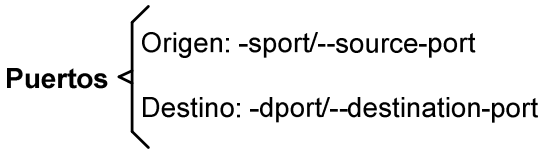
Se puede usar la orden ifconfig para obtener una lista de las interfaces que están up (esto es, funcionando en ese momento).

Protocolo: Se puede especificar el protocolo con el indicador -p/--protocol. El protocolo puede ser un número (si sabe los valores numéricos) o un nombre en el caso especial de TCP, UDP o ICMP. No importa si lo pone en mayúscula o minúscula; tcp valdrá lo mismo que TCP.

El nombre de protocolo puede ir prefijado de una “!”, para invertirlo, de manera que “-p ! TCP” especifica paquetes que no sean TCP.

Puerto: Un número de puerto especifica la aplicación a la que se dirigen los datos. Existen miles de puertos, codificados en 16 bits, es decir que se cuenta con 65536 posibilidades (0 → 65535). Es por eso que el IANA (Internet Assigned Numbers Authority) desarrollo una aplicación estándar para ayudar con las configuraciones de red:

- Los puertos del 0 al 1023 son los puertos conocidos. En términos generales, están reservados para procesos del sistema (daemons) o programas ejecutados por usuarios privilegiados. Sin embargo, un administrador de red puede conectar servicios con puertos de su elección.
- Los puertos del 1024 al 49151 son los puertos registrados.
- Los puertos del 49152 al 65535 son los puertos dinámicos y/o privados.



Los puertos origen (-sport/--source-port) y destino (-dport/--destination-port) se pueden especificar de 3 maneras:

- Un número de puerto (-sport 80).
- Un nombre de servicio (-sport http).
- Un rango de puertos (-sport 1:1023).

Control estado conexión -m state --state *ESTADOS*

Es el soporte básico para las reglas de filtrado con estado. Especifica la situación del paquete respecto a la conexión a la que pertenece:

- INVALID paquete que no es parte de una conexión existente e incapaz de crear una conexión nueva.
- ESTABLISHED paquete que pertenece a una conexión válida ya establecida.
- NEW paquete mediante el cual se está creando una nueva conexión.
- RELATED paquetes que inician una nueva conexión que está asociada con otra ya establecida.

En la Tabla 1.3 se indican otros comandos muy útiles para el uso de iptables.

Comando	Descripción
-m conntrack --cstate	Filtros con estados más detallados.
--tcp-flag	Verifica paquetes TCP que tienen marcadas o desmarcadas ciertas banderas del protocolo TCP.
-m –mac-source	Direcciones MAC
-m limit (--limit I –limit-burst)	Limita el número de paquetes/segundo.

Tabla 1.3 Otros comandos

1.3.4.3 Objetivos

El destino de una regla (objetivo) puede ser el nombre de una cadena definida por el usuario o uno de los destinos ya incorporados: ACCEPT (aceptar), DROP (descartar), REJECT (rechazar), MASQUERADE (enmascarar), entre otros.

En las Tablas 1.4, 1.5 y 1.6 se indican los objetivos (destinos de una regla) de cada una de las tablas que conforman iptables.

Objetivos	Descripción
-j ACCEPT	El paquete se acepta y se deja recorrer la cadena.
-j DROP	Se rechaza el paquete, sin informar al origen.
-j REJECT --reject-with	Se rechaza el paquete, informando al origen con el mensaje ICMP indicado.

Tabla 1.4 Objetivos predefinidos de la tabla filter

Comando	Descripción
-j SNAT --to-source	Realiza SNAT (source NAT) sobre los paquetes salientes (enmascaramiento de direcciones). Cambia dirección IP (opcional puerto) de origen del paquete. Solo disponible en POSTROUTING.
-j MASQUERADE	Idéntico que SNAT, pero usa la dirección IP del propio equipo. Útil en conexiones volátiles (ADSL, modem). Solo disponible en POSTROUTING.
-j DNAT --to-destination	Realiza DNAT (destination NAT) sobre los paquetes entrantes (redireccionamiento de puertos). Cambia dirección IP (opcional puerto) de destino del paquete. Solo disponible en PREROUTING y OUTPUT.

Tabla 1.5 Objetivos predefinidos de la tabla nat

Objetivos	Descripción
-j LOG	Crea entrada en el log del sistema (/var/log/syslog).
-j ULOG	Crea entrada en un log definido por el usuario.

Tabla 1.6 Objetivos de log

1.3.5 EJEMPLO DE USO DE IPTABLES[56]

Se desea implementar un firewall básico de borde con enmascaramiento de la red interna (NAT), cumpliendo los siguientes requisitos:

- Red interna del rango 192.168.10.0 / 24.
- Interfaces del firewall: eth0 (red externa) y eth1 (red interna).
- Servidor HTTP (puerto 80) y HTTPS (puerto 443) en 192.168.10.5 / 24.
- Servidor corporativo de correo saliente (SMTP, puerto 25) y entrante (POP3, puerto 110) en 192.168.10.6 / 24
- Todas las conexiones al firewall están prohibidas.
- Solo se permiten conexiones SSH (puerto 22) que provengan de equipos de la red interna. (Para labores de administración).
- No se permite la conexión para envío de e-mail (SMTP) con computadores del exterior.
- Todas los demás tipos de conexión están autorizados.

En la Figura 1.16 se indica el diagrama de red, del firewall a implementar con iptables, tomando en consideración los requisitos antes mencionados.

[56] http://www.pello.info/filez/firewall/iptables.html

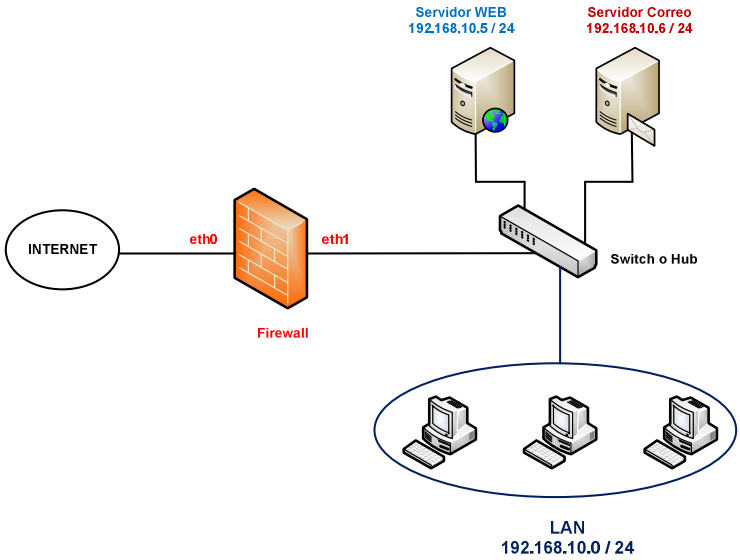


Figura 1.16 Diagrama de red, que permite ver un ejemplo de uso de iptables

A continuación se indica la configuración básica realizada para la implementación del firewall antes descrito.

```
# ! /bin/sh
# Ejemplo de SCRIPT DE IPTABLES para FIREWALL entre red local e internet.
#
# Vaciar y reiniciar las tablas actuales.
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

# Establecemos políticas por defecto
iptables -P INPUT DROP                  # descartar entradas al firewall
iptables -P OUTPUT ACCEPT               # aceptar salidas del firewall
iptables -P FORWARD ACCEPT              # aceptar reenvíos a través del firewall
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

# Permitir localhost
iptables -A INPUT -i lo -j ACCEPT

# Enmascaramiento de la red local
iptables -t nat -A POSTROUTING -j MASQUERADE

# Activamos el bit de forward, para que otras máquinas puedan salir a través del
# firewall
echo 1 > /proc/sys/net/ipv4/ip_forward

# Redireccionamiento de puertos 80, 443, 25, 110 a red interna
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT –to 192.168.10.5:80
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT –to 192.168.10.5:443
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 25 -j DNAT –to 192.168.10.6:25
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 110 -j DNAT –to 192.168.10.6:110
```

```
# Denegamos cualquier otra redirección
 iptables -t nat -A PREROUTING -i eth0 -d 192.168.10.0/24 -j DROP

# Permitimos SSH de la red interna
iptables -A INPUT -i eth1 -s 192.168.10.0/24 -p tcp --dport 22 -j ACCEPT

# Bloqueamos la salida SMTP
iptables -A FORWARD -i eth1 -o eth0 -s 192.168.10.0/24 -p tcp --dport 25 -j DROP
```

1.4 TRANSACCIÓN ELECTRÓNICA SEGURA (SET)[57]

1.4.1 INTRODUCCIÓN

SET (Secure Electronic Transactions) es un conjunto de especificaciones desarrolladas por VISA y MasterCard en Febrero de 1996, con el apoyo y asistencia de IBM, Microsoft, Netscape, RSA, Terisa y Verisign, que da paso a una forma segura de realizar transacciones electrónicas en redes de computadoras inseguras, en especial Internet, mediante el uso de tarjetas de crédito.

SET provee tres servicios:

- Un canal de comunicación seguro entre todas las partes involucradas en la transacción, ya que los datos viajan encriptados.
- Permite la identificación y autenticación del comerciante y el cliente, mediante el uso de certificados digitales X.509v3.
- Garantiza la confidencialidad de la información (datos de la tarjeta de crédito), ya que al estar el comprador identificado ante la entidad financiera por un certificado digital emitido por ella misma, no es preciso que la información de la tarjeta de crédito viaje, con lo que nunca llega a manos del comerciante ni puede ser interceptada por nadie.

[57] "Cryptography and Network Security: Principles and Practice", William Stallings, Second Edition, 1998

SET es una especificación compleja definida en tres libros publicados en Mayo de1997:

- Libro 1: *Business Description* (80 páginas).
- Libro 2: *Programmers Guide* (629 páginas).
- Libro 3: *Format Protocol Definition* (262 páginas).

1.4.2 REQUERIMIENTOS DEL NEGOCIO

El Libro 1 de la especificación de SET enlista los siguientes requerimientos del negocio para procesamiento de pago seguro con tarjeta de crédito a través de Internet y otras redes:

- Proveer confidencialidad de pago e información de órdenes de compra, mediante el uso de algoritmos de encripción como DES.
- Asegurar la integridad de la totalidad de los datos que se transmiten: firmas digitales RSA [58] y códigos hash SHA-1 [59] son usados para proveer integridad.
- Proveer autenticación de que el portador de una tarjeta es un usuario legítimo de una cuenta de tarjeta de crédito: SET utiliza certificados digitales X.509v3 y firmas digitales RSA para este propósito.
- Proveer autenticación de que el comerciante puede aceptar transacciones con tarjetas de crédito a través de su relación con una institución financiera: SET utiliza certificados digitales X.509v3 y firmas digitales RSA para este propósito.
- Asegurar el uso de las mejores prácticas de seguridad y de técnicas de diseño de sistemas para proteger los involucrados legítimos en la transacción de comercio electrónico.

[58] RSA: Sistema Criptográfico de Clave Pública descrito en 1977 por Ron Rivest, Adi Shamir y Len Adleman, del Instituto Tecnológico de Massachusetts (MIT); las letras RSA son las iniciales de sus apellidos.
[59] SHA-1: *Secure Hash Algorithm*

- Crear un protocolo que no dependa de mecanismos de seguridad de transporte ni que prevenga su uso: SET no interfiere con el uso de otros mecanismos de seguridad, tales como IPSec[60] y SSL/TLS[61].
- Facilitar y promover la interoperabilidad entre proveedores de software y redes: Los algoritmos y firmas digitales que usa SET son independientes a la plataforma hardware, sistema operativo y software Web.

1.4.3 PARTICIPANTES

El pago mediante tarjeta es un proceso complejo, en el cual se ven implicadas varias entidades, que se muestran en la Figura 1.17

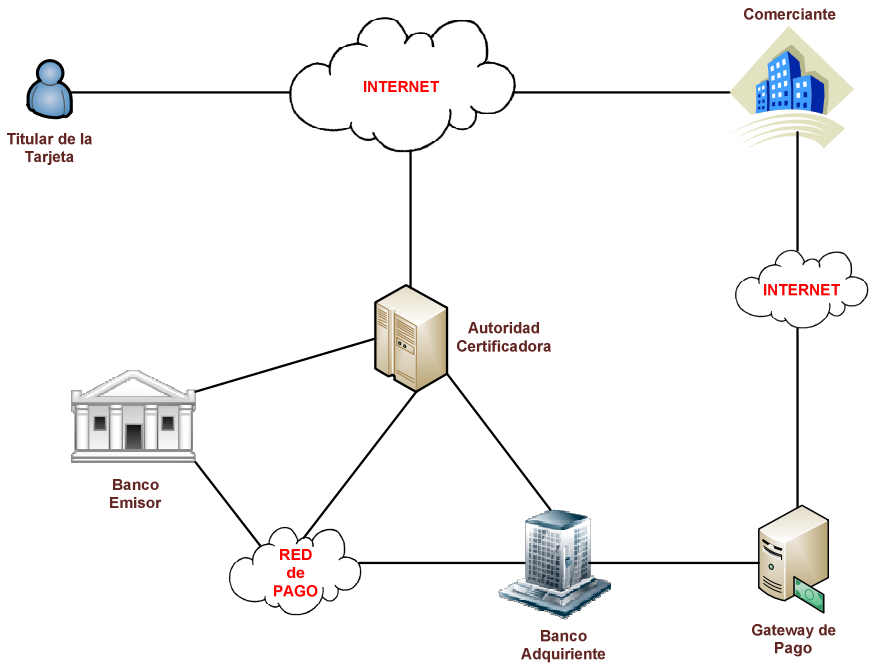


Figura 1.17 Participantes de SET

- El banco emisor: emite la tarjeta del cliente, extiende su crédito y es responsable de la facturación, recolección y servicio al consumidor.

[60] IPSec: *Internet Protocol security,* conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP) autenticando y/o cifrando cada paquete IP en un flujo de datos.
[61] SSL/TLS: *Secure Sockets Layer*/*Transport Layer Security*, protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet

- El banco adquiriente: establece una relación con el comerciante, procesando las transacciones con tarjeta y las autorizaciones de pago.
- El titular de la tarjeta: posee la tarjeta emitida por el banco emisor y realiza y paga las compras.
- El comerciante: vende productos, servicios o información y acepta el pago electrónico, que es gestionado por su entidad financiera (adquiriente).
- El *Gateway* de pago: mecanismo mediante el cual se procesan y autorizan las transacciones del comerciante. El gateway puede pertenecer a una entidad financiera (adquiriente) o a un operador de medio de pago, el cual procesa todas las transacciones de un conjunto de entidades.
- Autoridad de certificación: certifica las claves públicas del titular de la tarjeta, del comerciante y de los bancos.

1.4.4 FUNCIONAMIENTO[62]

Una transacción SET típica funciona de forma muy parecida a una transacción convencional con tarjeta de crédito y consta de los siguientes pasos:

1. Decisión de compra del cliente. El cliente está navegando por el sitio web del comerciante y decide comprar un artículo. Para ello rellenará algún formulario al efecto y posiblemente hará uso de alguna aplicación tipo carrito de la compra, para ir almacenando diversos artículos y pagarlos todos al final. El protocolo SET se inicia cuando el comprador pulsa el botón de Pagar.
2. El comerciante envía una descripción del pedido a la aplicación del cliente.
3. El cliente comprueba el pedido y transmite una orden de pago de vuelta al comerciante. El cliente crea dos mensajes que envía al comerciante. El primero, la información del pedido (OI), contiene los datos del pedido, mientras que el segundo contiene las instrucciones de pago del cliente (PI) (número de tarjeta de crédito, banco emisor, etc.) para el banco adquiriente. En este momento, el cliente genera un firma dual (DS), que permite juntar en un solo mensaje la información del pedido y las

[62] http://www.iec.csic.es/criptonomicon/comercio/set.html

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 envían 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 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).

1.4.5 DOBLE FIRMA (DS)[63]

SET hace uso del concepto de doble firma (DS) que permite juntar en un solo mensaje la información del pedido (OI) y las instrucciones de pago (PI).

En la Figura 1.18 se muestra el proceso de construcción de la doble firma (DS).

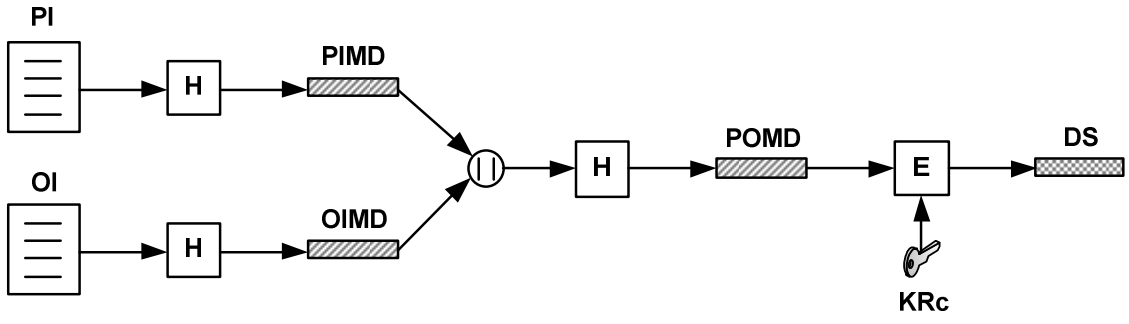


Figura 1.18 Construcción de la Doble Firma

1. El cliente obtiene el hash (usando SHA-1) de las instrucciones de pago (PI) y la información del pedido (OI), obteniendo así el PIMD (*Payment*

[63] "Cryptography and Network Security: Principles and Practice", William Stallings, Second Edition, 1998

Information Message Digest) y el OIMD (*Order Information Message Digest*).

2. El cliente combina (concatena) el PIMD y el OIMD, y de este resultado obtiene el hash (usando SHA-1), para producir el POMD (*Payment and Order Message Digest*).
3. El cliente encripta el POMD con su llave privada. La salida de este proceso es la doble firma DS (*Digital Signature*).

El cliente envía:

- OI, PIMD y DS al comerciante.
- E{PI}, E{DS}, E{OIMD} y E{Ks} al Banco. Como se muestra en la Figura 1.19

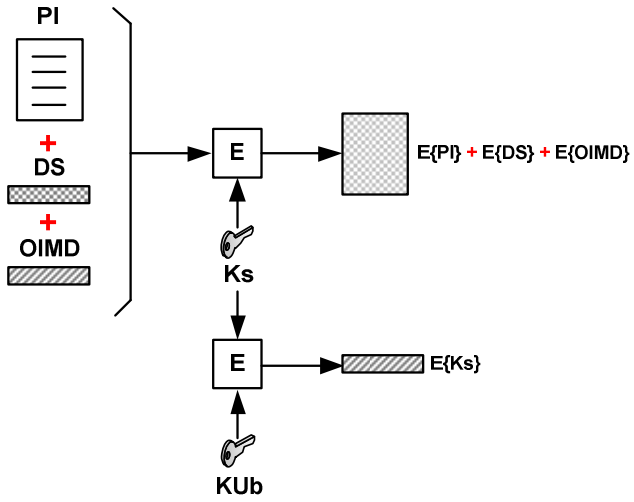


Figura 1.19 Información que se envía al Banco

Como se puede observar el comerciante puede acceder a la información del pedido (OI), pero no a las instrucciones de pago, mientras que el banco puede acceder a las instrucciones de pago (PI), pero no a la información del pedido.

El comerciante para validar la orden del cliente, sigue los siguientes pasos:

1. El comerciante calcula su OIMD, y usa este y el PIMD recibido del cliente para generar el POMD1 (Figura 1.20)

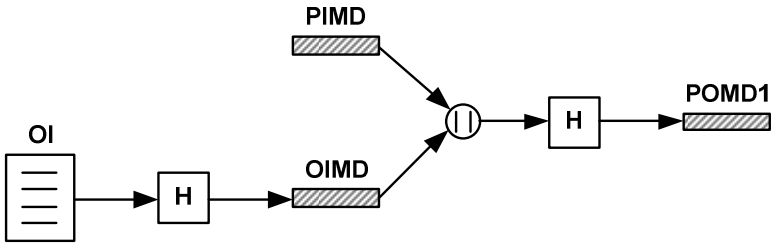


Figura 1.20 POMD1

2. El comerciante desencripta el DS recibido del cliente, con la llave pública del cliente (KUc), obteniendo así POMD2 (Figura 1.21)

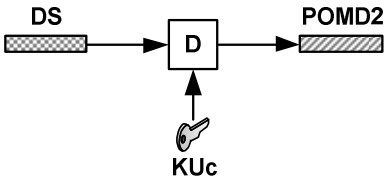


Figura 1.21 POMD2

3. El comerciante compara el POMD1 y el POMD2, si son iguales, valida la orden del cliente y envía la información que se muestra en la Figura 1.19 al Banco.

El Banco, para realizar la transacción del pago, sigue los siguientes pasos:

1. El banco desencripta el E{Ks} recibido del comerciante, con la llave privada del banco (KRb), obteniendo así la llave simétrica Ks (Figura 1.22)

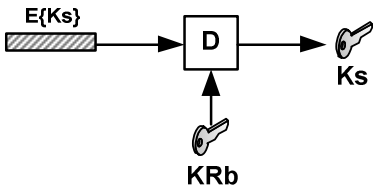


Figura 1.22 Llave simétrica Ks

2. Con la llave simétrica Ks, desencripta: E{PI}, E{DS} y E{OIMD} obteniendo así el PI, DS y OIMD (Figura 1.23)

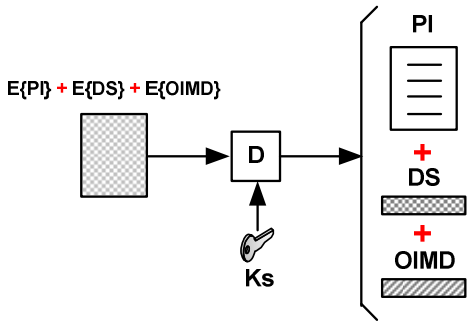


Figura 1.23 PI, DS y OIMD

3. Con el PI, DS y OIMD el banco valida la petición del cliente. El proceso es muy similar al realizado por el comerciante, compara el POMD1 y el POMD2, si son iguales, realiza las instrucciones de pago que se encuentran en PI (Figura 1.24)

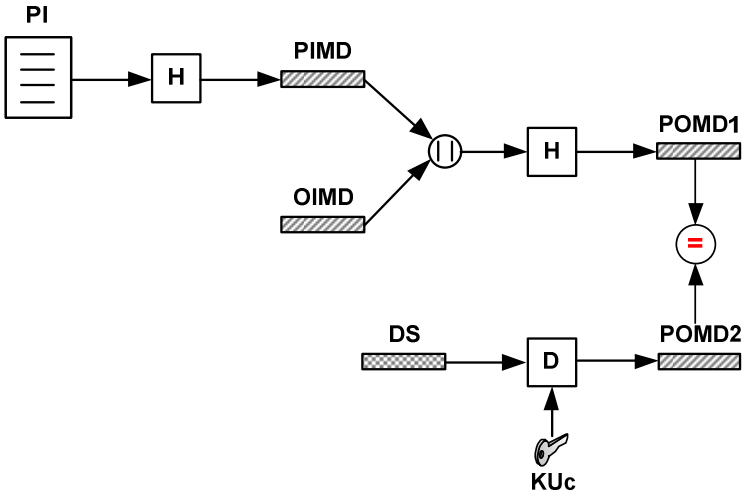


Figura 1.24 Validación del cliente por parte del banco

El protocolo definido por SET especifica el formato de los mensajes, las codificaciones y las operaciones criptográficas que deben usarse. No requiere un método particular de transporte, de manera que los mensajes SET pueden transportarse sobre HTTPS en aplicaciones web, sobre correo electrónico o cualquier otro método. Como los mensajes no necesitan transmitirse en tiempo presente, son posibles implantaciones de SET eficientes basadas en correo electrónico.

CAPÍTULO 2: DISEÑO E IMPLEMENTACIÓN DEL PORTAL CAUTIVO

2.1 INTRODUCCIÓN

Debido al alto crecimiento y evolución que se ha dado en la tecnología inalámbrica, tanto en dispositivos móviles como en dispositivos de red, el uso de portales cautivos es más frecuente permitiendo a un usuario de una red interactuar primero con una página web en la cual deben ingresar un nombre de usuario y una contraseña asignadas, para así poder navegar por Internet de forma normal. En la Figura 2.1 se muestra el funcionamiento del Portal Cautivo.

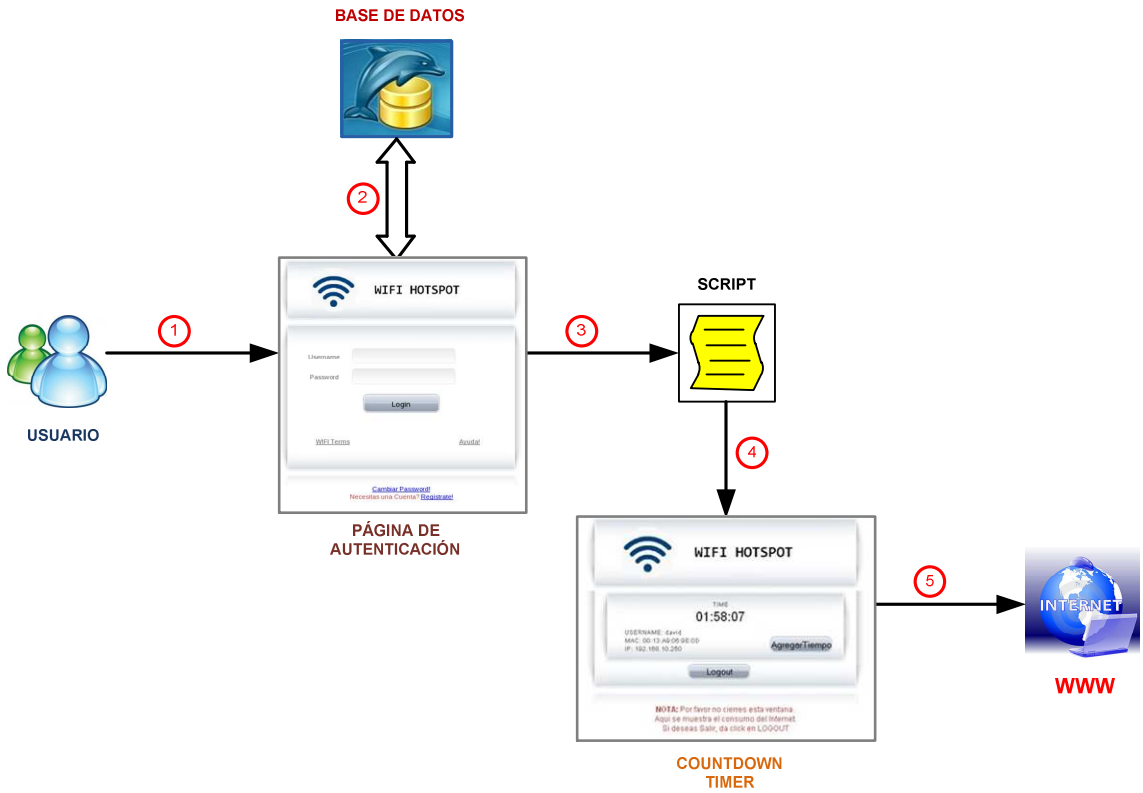


Figura 2.1 Funcionamiento del Portal Cautivo

1. Todo usuario que desea ingresar al Internet es redireccionado a la Página de Autenticación del Portal Cautivo, en donde debe ingresar un nombre de

usuario (username) y una contraseña (password) asignados, para así poder navegar por Internet de forma normal.

2. El Portal Cautivo comprueba que el nombre de usuario y la contraseña que el usuario ingreso existen en la Base de Datos.
3. Si el usuario existe en la Base de Datos, se generan las reglas necesarias que permiten el acceso del usuario al Internet.
4. Aparece una ventana (CountDown Timer), en la cual se muestra el tiempo de consumo del Internet, la dirección IP, la dirección MAC y el hostname del usuario.
5. El usuario es redireccionado a la página web que solicitó, verificando así que ya puede navegar por Internet de forma normal.

El Portal Cautivo es una computadora que, formando parte de una red, provee de varios servicios a otras computadoras denominadas clientes. No necesariamente es una computadora de última generación, un Portal Cautivo puede ser una computadora normal de escritorio (desktop) o hasta una computadora sumamente potente (server).

En la Figura 2.2 se muestran los elementos que debe tener un portal cautivo, entre ellos:

- Firewall, va ser el encargado de permitir, bloquear y redireccionar todo el tráfico de los clientes que estén conectados al Portal Cautivo.
- Servidor WEB, permite montar las páginas web del Portal Cautivo, entre ellas la página de autenticación, las páginas de ingreso de usuarios.
- Servidor de Base de Datos, es el encargado de almacenar la información de los usuarios, tickets y configuración del Portal cautivo.
- Servidor DHCP, permite administrador, supervisar y distribuir de forma centralizada las direcciones IP necesarias y, automáticamente, asignar una dirección IP a una computadora de la red.

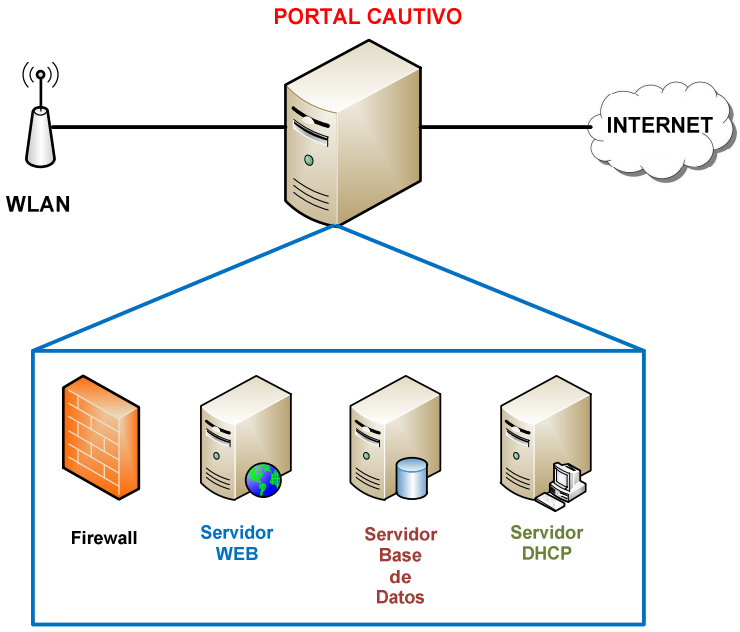


Figura 2.2 Elementos de un Portal Cautivo

Para el diseño del Portal Cautivo se emplea el "Proceso Unificado de Desarrollo de Software", y para la implementación se emplean varias herramientas de software libre (programas) que son descritos posteriormente.

Se seleccionó el "Proceso Unificado" sobre los demás modelos de desarrollo de software por proporcionar una guía detallada para el desarrollo de aplicaciones permitiendo así minimizar los riesgos antes de que ocurran.

2.2 PROCESO UNIFICADO[64]

El Proceso Unificado es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños del proyecto

[64] "El Proceso Unificado de Desarrollo de Software", Ivar Jacobson, Grady Booch, James Rumbaugh, 2000.

El Proceso Unificado de Desarrollo de Software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. La Figura 2.3 muestra el proceso de desarrollo de software.

Figura 2.3 Proceso de Desarrollo de Software

Tres son los aspectos que define al Proceso Unificado: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

2.2.1 DIRIGIDO POR CASOS DE USO

Un sistema software se crea para servir a sus usuarios por lo que, para construir un sistema exitoso, se debe conocer qué es lo que quieren y necesitan. El término "usuario" no se refiere solamente a los usuarios humanos sino también a otros sistemas, es decir, representa a algo o alguien que interactúa con el sistema a desarrollar.

En el Proceso Unificado, los casos de uso se utilizan para capturar los requisitos funcionales y para definir los objetivos de las iteraciones. En cada uno, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso.

2.2.2 CENTRADO EN LA AQUITECTURA

El concepto de arquitectura del software involucra los aspectos estáticos y dinámicos más significativos del sistema, y actúa como vista del diseño, dando una perspectiva completa y describiendo los elementos más importantes. La arquitectura surge de los propios casos de uso, sin embargo, también está influenciada por muchos otros factores, como la plataforma en la que se

ejecutará, el uso de estándares, la existencia de sistemas heredados o los requisitos no funcionales.

Puesto que la arquitectura y los casos de uso están relacionados, por una parte, los casos de uso deben, cuando son realizados, acomodarse en la arquitectura, y ésta debe ser lo bastante flexible para realizar todos los casos de uso, hoy y en el futuro. En la realidad, arquitectura y casos de uso deben evolucionar en paralelo.

2.2.3 ITERATIVO E INCREMENTAL

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas:

- Fase de Inicio, tiene por finalidad definir la visión, los objetivos y el alcance del proyecto, tanto desde el punto de vista funcional como del técnico, obteniéndose como uno de los principales resultados una lista de los casos de uso y una lista de los factores de riesgo del proyecto. El principal esfuerzo está radicado en el Modelamiento del Negocio y el Análisis de Requerimientos.
- Fase de Elaboración, tiene como principal finalidad completar el análisis de los casos de uso y definir la arquitectura del sistema, además se obtiene una aplicación que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- Fase de Construcción, está compuesta por un ciclo de varias iteraciones, en las cuales se van incorporando sucesivamente los casos de uso, de acuerdo a los factores de riesgo del proyecto. Este enfoque permite por ejemplo contar en forma temprana con versiones el sistema que satisfacen los principales casos de uso. Los cambios en los requerimientos no se incorporan hasta el inicio de la próxima iteración.

- Fase de Transición, se inicia con una versión “beta” del sistema y culmina con el sistema en fase de producción.

Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. La representación de las fases y su relación con las iteraciones se muestra en la Figura 2.4.

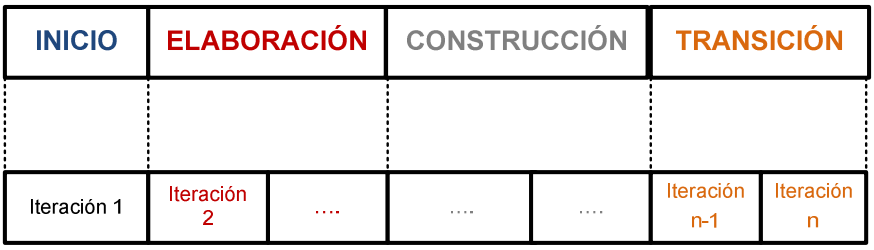


Figura 2.4 Relación entre fases e iteraciones del Proceso Unificado[65]

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas: Requisitos, Análisis, Diseño, Implementación y Prueba; conocidas también como Flujo de trabajo del Proceso Unificado. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto. En la Figura 2.5 se muestran las diferentes disciplinas del Proceso Unificado con cada uno de los modelos[66] que lo conforman.

[65] “El Proceso Unificado de Desarrollo de Software”, Ivar Jacobson, Grady Booch, James Rumbaugh, 2000.
[66] Modelo: Abstracción del sistema.

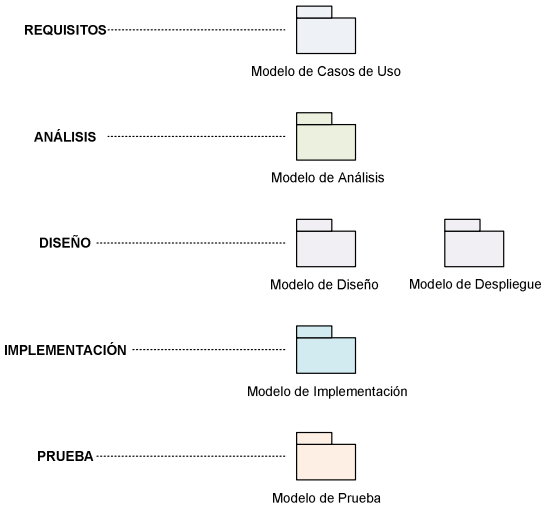


Figura 2.5 Disciplinas y Modelos del Proceso Unificado[67]

A continuación se describen cada uno de los modelos que conforman el proceso Unificado de Desarrollo del Software.

2.2.3.1 Modelo de Casos de Uso

Este modelo ayuda al cliente, a los usuarios y a los desarrolladores del sistema a llegar a un acuerdo sobre los requisitos que debe cumplir el sistema de software. El modelo de casos está compuesto por: actores[68], casos de uso y sus relaciones. En la Figura 2.6 se muestra el diagrama del modelo de casos de uso.

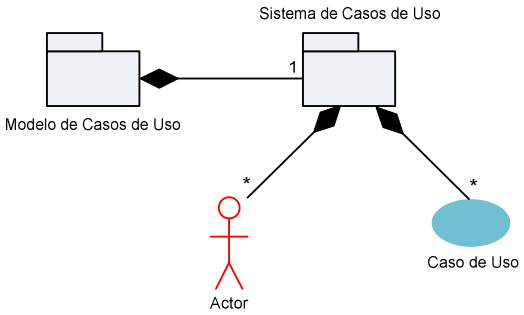


Figura 2.6 Diagrama del modelo de casos de uso[69]

[67] "El Proceso Unificado de Desarrollo de Software", Ivar Jacobson, Grady Booch, James Rumbaugh, 2000

[68] Actor: Usuario que utiliza el sistema para interactuar con los casos de usos. No todos los actores representan a personas, pueden ser otros sistemas o hardware externo que interactúa con el sistema.

[69] "El Proceso Unificado de Desarrollo de Software", Ivar Jacobson, Grady Booch, James Rumbaugh, 2000

2.2.3.2 Modelo de Análisis

El modelo de análisis se representa mediante un sistema de análisis que denota el paquete de más alto nivel del modelo. Dentro del modelo de análisis los casos de uso se describen mediante clases de análisis y sus objetos.

Las clases del análisis representan abstracciones de clases y posiblemente de subsistemas del diseño del sistema. Las clases del análisis se clasifican en tres tipos:

- Clase de Interfaz: Se utiliza para modelar la interacción (intercambio de información y peticiones) entre el sistema y los actores.
- Clase de Entidad: Se emplea para representar la información y el comportamiento asociado a un objeto suceso del mundo real (persona).
- Clase de Control: Se utiliza para representar coordinación, secuencia, transacciones, control de objetos, derivaciones y cálculos complejos que no pueden asociarse a una clase entidad.

Una realización del caso de uso – análisis es una colaboración dentro del modelo de análisis que describe cómo se lleva a cabo y se ejecuta un caso de uso determinado. El Paquete del Análisis proporciona un medio para organizar los artefactos del modelo del análisis en piezas manejables. En la Figura 2.7 se muestra el diagrama del modelo de análisis.

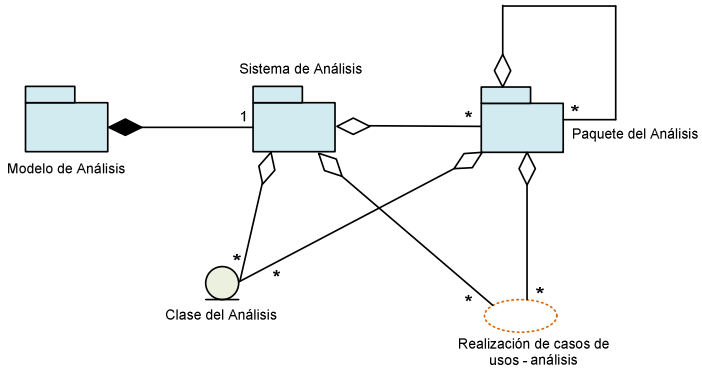


Figura 2.7 Diagrama del modelo de análisis[70]

[70] "El Proceso Unificado de Desarrollo de Software", Ivar Jacobson, Grady Booch, James Rumbaugh, 2000

2.2.3.3 Modelo de Diseño

Es un modelo de objetos que describe la realización física de los casos de uso, tomando en consideración las restricciones relacionadas al entorno de implementación, este modelo sirve como abstracción de la implementación del sistema. Las clases de diseño representan abstracciones del subsistema y componentes de la implementación del sistema, representan una sencilla correspondencia entre el diseño y la implementación.

El subsistema de diseño permite organizar los artefactos del modelo de diseño en piezas más manejables, puede ser conformado por clases de diseño, casos de uso – diseño, interfaces y otros subsistemas. La realización de casos de uso – diseño describe como se realiza un caso de uso en términos de interacción entre objetos de diseño. Las interfaces son utilizadas para especificar las operaciones que proporcionan las clases y los subsistemas. En la Figura 2.8 se muestra el diagrama del modelo de diseño.

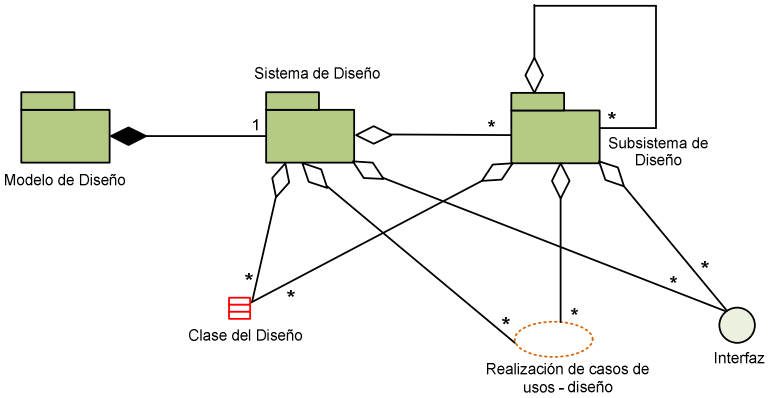


Figura 2.8 Diagrama del modelo de diseño[71]

2.2.3.4 Modelo de Despliegue

Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. En la Figura 2.9 se muestra el diagrama del modelo de despliegue.

[71] "El Proceso Unificado de Desarrollo de Software", Ivar Jacobson, Grady Booch, James Rumbaugh, 2000.

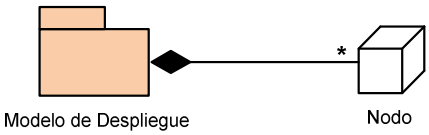


Figura 2.9 Diagrama del modelo de despliegue

2.2.3.5 Modelo de Implementación

Describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados.

El modelo de implementación se representa con un sistema de implementación que denota el subsistema de nivel superior del modelo. El utilizar otros subsistemas es por tanto una forma de organizar el modelo de implementación en trozos más manejables. El modelo de implementación contiene componentes e interfaces.

El componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Algunos estereotipos estándar de componentes son los siguientes:

- <<executable>> es un programa que puede ser ejecutado en un nodo.
- <<file>> es un fichero que contiene código fuente o datos.
- <<library>> es una librería estática o dinámica.
- <<table>> es una tabla de base de datos.
- <<document>> es un documento.

El subsistema de implementación proporciona una forma de organizar los artefactos del modelo de implementación en trozos más manejables. Se

manifiesta a través de un mecanismo de empaquetamiento concreto en un entorno de implementación determinado, tales como:

- Un paquete en Java.
- Un proyecto en Visual Basic.
- Un directorio de ficheros en un proyecto de C++.
- Un subsistema en un entorno de desarrollo integrado como Rational Apex[72].
- Un paquete en una herramienta de modelado visual como Rational Rose[73].

En la Figura 2.10 se muestra el diagrama del modelo de implementación.

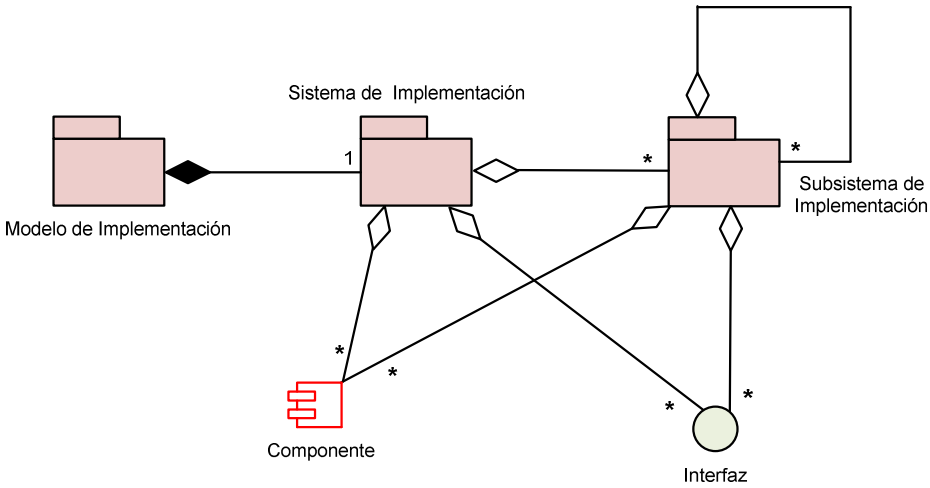


Figura 2.10 Diagrama del modelo de implementación[74]

2.2.3.6 Modelo de Pruebas

El modelo de pruebas describe principalmente cómo se prueban los componentes ejecutables en el modelo de implementación con pruebas de integración y de sistema. El modelo de pruebas es una colección de:

[72] Rational Apex: Entorno de desarrollo de software de IBM.
[73] Rational Rose: Herramienta software para el Modelado Visual mediante UML de sistemas software
[74] "El Proceso Unificado de Desarrollo de Software", Ivar Jacobson, Grady Booch, James Rumbaugh, 2000.

- Casos de prueba: especifican una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones baja las que ha de probarse.
- Procedimientos de prueba: especifican cómo realizar uno o varios casos de prueba o partes de estos.
- Componentes de prueba: automatizan uno o varios procedimientos de prueba o partes de ellos.

En la Figura 2.11 se muestra el diagrama del modelo de pruebas.

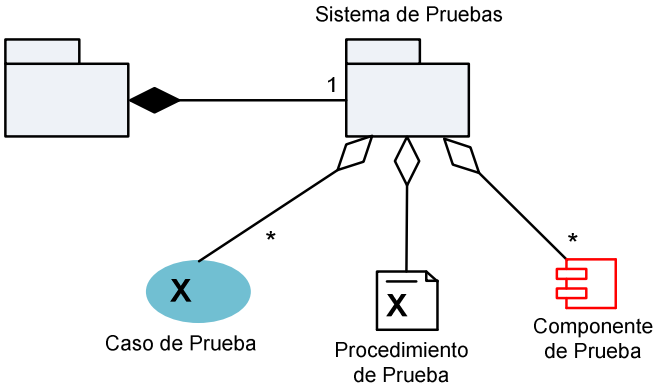


Figura 2.11 Diagrama del modelo de pruebas[75]

2.3 DISEÑO DEL PORTAL CAUTIVO

Al emplear el Proceso Unificado de Desarrollo de Software en el diseño del Portal Cautivo se deben desarrollar los pasos descritos anteriormente.

2.3.1 MODELO DE CASOS DE USO

EL objetivo principal del Portal Cautivo es permitir a un usuario de una red interactuar primero con una página web en la cual debe ingresar un nombre de usuario y una contraseña asignadas, para así poder navegar por Internet.

[75] "El Proceso Unificado de Desarrollo de Software", Ivar Jacobson, Grady Booch, James Rumbaugh, 2000

En la Tabla 2.1 se muestran los actores que van a participar en la descripción de los casos de uso.

ACTOR	DESCRIPCIÓN
Usuario	Persona que desea navegar por Internet.
Usuario Autenticado	Persona que ingresó su nombre de usuario y contraseña correcta en el Portal Cautivo.
Vendedor	Persona que vende los tickets de Internet.
Base de Datos	Programa que almacena la información de los tickets y usuarios.

Tabla 2.1 Actores de los casos de uso

En la Figura 2.12 se muestran los casos de uso que se van a implementar en el Portal Cautivo.

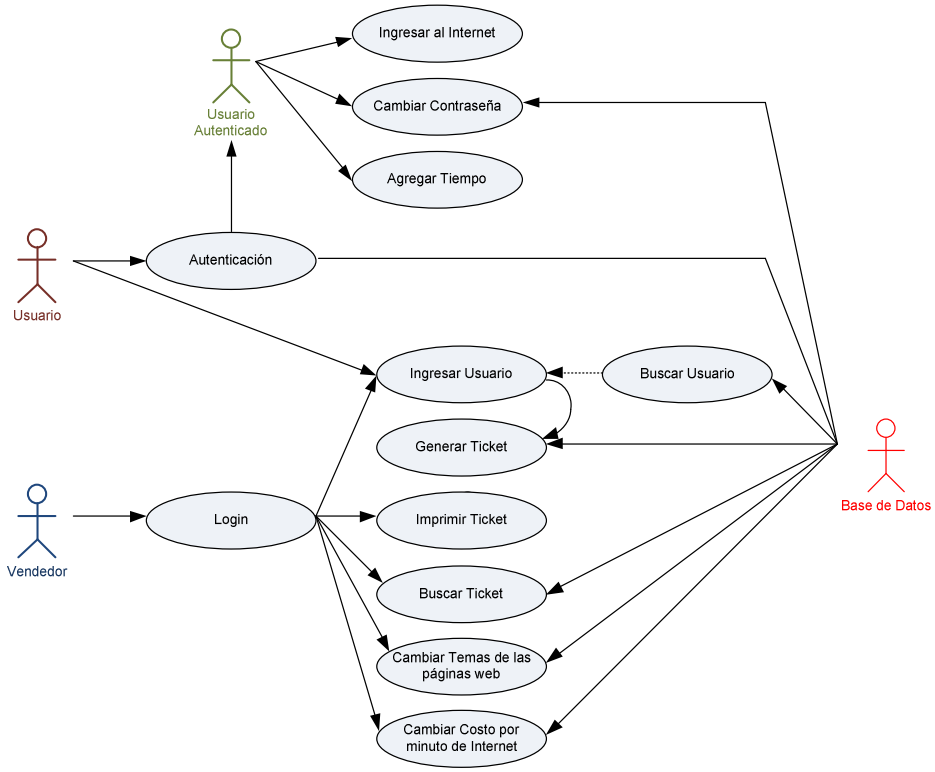


Figura 2.12 Casos de Uso del Portal Cautivo

La descripción de cada uno de los casos de uso se especifican en las tablas 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12 y 2.13

Nombre:	Autenticación
Descripción:	Permite el ingreso al Portal Cautivo a usuarios que posean el nombre de usuario y contraseña válidos.
Actores:	Usuario y Base de Datos
Precondición:	Haber ingresado a una página web.
Flujo Normal:	1 El usuario ingresa su nombre de usuario y contraseña en la página web de bienvenida. 2 El usuario presiona el botón Login. 3 El nombre de usuario y contraseña son correctos.
Flujo Alternativo:	1 En 3 el nombre de usuario y contraseña son incorrectos.
Poscondición:	El usuario se encuentra autenticado y puede ingresar al Internet.

Tabla 2.2 Descripción del caso de uso Autenticación

Nombre:	Ingresar al Internet
Descripción:	Permite el ingreso del usuario autenticado al Internet.
Actor:	Usuario Autenticado
Precondición:	Usuario previamente autenticado.
Flujo Normal:	1 Se abre una página web en la cual se muestra: un reloj que va decreciendo conforme pasa el tiempo, la dirección MAC, la dirección IP. 2 El usuario tiene habilitado el acceso al Internet.
Flujo Alternativo:	1 En 1 si se presiona el botón de Logout, el usuario cierra la sesión existente. 2 En 1 si se presiona el botón Agregar Tiempo, el usuario cierra la sesión existente y abre una página en la cual debe ingresar el tiempo que desea agregar. 3 Se deshabilita el acceso al Internet.
Poscondición:	El usuario se encuentra autenticado y puede: ingresar al Internet o deshabilitar el acceso.

Tabla 2.3 Descripción del caso de uso Ingresar al Internet

Nombre:	**Cambiar Contraseña**
Descripción:	Permite al usuario cambiar la contraseña.
Actores:	Usuario Autenticado y Base de Datos
Precondición:	Usuario previamente autenticado.
Flujo Normal:	1 Se presiona el enlace de cambiar password. 2 Se abre una página web en la cual se deben ingresar el nombre de usuario, la contraseña anterior y la nueva contraseña. 3 Si el nombre de usuario y contraseña anterior son correctos, se realiza el cambio del password.
Flujo Alternativo:	1 En 3 si el nombre de usuario o contraseña anterior son incorrectos, se abre una página indicando que los datos ingresados son incorrectos.
Poscondición:	El password del usuario ha cambiado.

Tabla 2.4 Descripción del caso de uso Cambiar Contraseña

Nombre:	**Agregar Tiempo**
Descripción:	Permite al usuario agregar tiempo de consumo de Internet.
Actores:	Usuario Autenticado y Base de Datos.
Precondición:	Usuario previamente autenticado.
Flujo Normal:	1 Se presiona el botón Agregar Tiempo. 2 Se abre una página web en la cual se debe seleccionar el tiempo de consumo de Internet en horas y minutos que se desea agregar. 3 Se presiona el botón Continuar. 4 Se ingresan los datos de la tarjeta de crédito. 5 Se verifica y valida la tarjeta de crédito. 6 Se agrega el tiempo en la Base de Datos.
Flujo Alternativo:	1 En 5 si existe un error en la tarjeta de crédito, se muestra una página web informando acerca del error.
Poscondición:	El password del usuario ha cambiado.

Tabla 2.5 Descripción del caso de uso Agregar Tiempo

Nombre:	Ingresar Usuario
Descripción:	Permite el ingreso de los datos personales del usuario.
Actores:	Usuario.
Precondición:	Usuario inexistente en la base de datos.
Flujo Normal:	1 Se ingresa un nombre de usuario y contraseña válidos. 2 Se presiona el botón Continuar. 3 Se selecciona de uso del Internet en horas y minutos. 4 Se presiona el botón Continuar. 5 Se ingresa los datos personales del usuario: nombres, apellidos, número de cédula, dirección, teléfono, email, ciudad. 6 Se ingresa los datos de la tarjeta de crédito, si el usuario está interactuando directamente con el Portal Cautivo. 7 Se presiona el botón Continuar. 8 Se verifica los datos ingresados y se valida la tarjeta de crédito. 9 Se ingresan los datos del usuario y del ticket en la base de datos. 10 Se almacena la información del usuario en la base de datos.
Flujo Alternativo:	1 En 1 si el nombre de usuario existe en la base de datos, se muestra un mensaje informando acerca del error. 2 En 8 si existe un error en el ingreso de los datos personales o de la tarjeta de crédito, se muestra una página web informando acerca del error. 3 En 1 el vendedor puede registrar al usuario. 4 El vendedor ingresa el tiempo de uso de Internet y la forma de pago, ya sea efectivo o tarjeta de crédito.
Poscondición:	Generar ticket.

Tabla 2.6 Descripción del caso de uso Ingresar Usuario

Nombre:	Buscar Usuario
Descripción:	Permite buscar un usuario en la base de datos.
Actores:	Usuario, Vendedor y Base de Datos
Precondición:	Usuario existente en la base de datos.
Flujo Normal:	1 Se ingresa el número de cédula del usuario. 2 Se presiona el botón Buscar. 3 Si el usuario existe en la base de datos, se muestra una página web con información personal del usuario.
Flujo Alternativo:	1 En 3 si el usuario no existe en la base de datos, se ingresa los datos personales del usuario.
Poscondición:	Registrar al Usuario.

Tabla 2.7 Descripción del caso de uso Buscar Usuario

Nombre:	Generar Ticket
Descripción:	Permite la generación del ticket.
Actores:	Usuario.
Precondición:	Registrar usuario.
Flujo Normal:	1 En el caso del vendedor, se genera un nombre de usuario y contraseña automáticamente. 2 En el caso del usuario, él ingresa un nombre de usuario y contraseña válidos. 3 Se almacena la información del ticket en la base de datos.
Poscondición:	Imprimir Ticket.

Tabla 2.8 Descripción del caso de uso Generar Ticket.

Nombre:	Imprimir Ticket
Descripción:	Permite al vendedor imprimir el ticket del usuario.
Actores:	Vendedor previamente autenticado.
Precondición:	Vendedor se encuentre autenticado.
Flujo Normal:	1 Se presiona el botón Imprimir. 2 Se imprime el ticket en formato pdf.

Tabla 2.9 Descripción del caso de uso Imprimir Ticket

Nombre:	Login
Descripción:	Permite al vendedor autenticarse.
Actores:	Vendedor y Base de Datos.
Precondición:	Haber ingresado a una página web.
Flujo Normal:	1 El vendedor ingresa su nombre de usuario y contraseña en la página web de bienvenida. 2 El vendedor presiona el botón Login. 3 El nombre de usuario y contraseña son correctos. 4 Se redirección a la página de bienvenida del vendedor.
Flujo Alternativo:	1 En 3 el nombre de usuario y contraseña son incorrectos.
Poscondición:	El vendedor se encuentra autenticado y puede realizar las distintas operaciones a él encomendadas.

Tabla 2.10 Descripción del caso de uso Login

Nombre:	Buscar Ticket
Descripción:	Permite al vendedor buscar un ticket.
Actores:	Vendedor previamente autenticado y Base de Datos.
Precondición:	Conocer el nombre del usuario.
Flujo Normal:	1 Se ingresa el nombre de usuario a buscar. 2 Se presiona el botón Buscar. 3 Se muestra una página web con información del usuario, los tiempos en los cuales ha ingresado al Internet.
Flujo Alternativo:	1 En 1 el nombre de usuario no se encuentra en la base de datos. 2 Se muestra un mensaje de error.

Tabla 2.11 Descripción del caso de uso Buscar Ticket

Nombre:	Cambiar temas de las páginas web
Descripción:	Permite al vendedor cambiar el tema de las páginas web de los usuarios.
Actores:	Vendedor previamente autenticado y Base de Datos.
Precondición:	Haber ingresado a configuración, temas.
Flujo Normal:	1 El vendedor selecciona el tema que desea que se muestre en las páginas del Portal Cautivo.
Poscondición:	Las páginas web de interacción entre el Portal Cautivo y el Usuario han cambiado.

Tabla 2.12 Descripción del caso de uso Cambiar temas de las páginas web

Nombre:	Cambiar costo por minuto del Internet
Descripción:	Permite al vendedor cambiar el costo por minuto del consumo de Internet.
Actores:	Vendedor previamente autenticado y Base de Datos.
Precondición:	Haber ingresado a configuración, temas.
Flujo Normal:	1 El vendedor selecciona el precio por minuto de consumo de Internet.
Flujo Alternativo:	1 En 1 se pueden adicionar impuestos a dicho precio.
Poscondición:	Los precios sean reales.

Tabla 2.13 Descripción del caso de uso Cambiar costo por minuto del Internet

2.3.2 DIAGRAMA DE FLUJO DEL PORTAL CAUTIVO

Una vez definidos los casos de usos del proceso unificado, se plantea el diagrama de flujo de la Figura 2.13. En donde:

1. Si un usuario abre el navegador, se abre a la página web de Login, en la cual el usuario puede: Ingresar el nombre de usuario y contraseña previamente solicitados o crear una cuenta.

2. Si el nombre de usuario y contraseña corresponden a los del vendedor, se abre la página web de bienvenida del vendedor, en la cual puede realizar las diferentes tareas descritas en los casos de uso.
3. Si no es el vendedor, se busca el nombre de usuario y contraseña en la base de datos, si los encuentra, se abre una página en la cual se muestra un reloj en decremento (countdown timer) y ciertos datos del usuario. Si no lo encuentra en la base de datos, se muestra una página de error.
4. Si el usuario no tiene un nombre de usuario y contraseña, procesa a registrarse en el portal Cautivo. Lo puede hacer de dos maneras: interactuando directamente con el portal cautivo o comprando un ticket de Internet al vendedor. Todos los pasos que se necesitan para obtener el ticket están descritos en los casos de usos previamente estudiados.

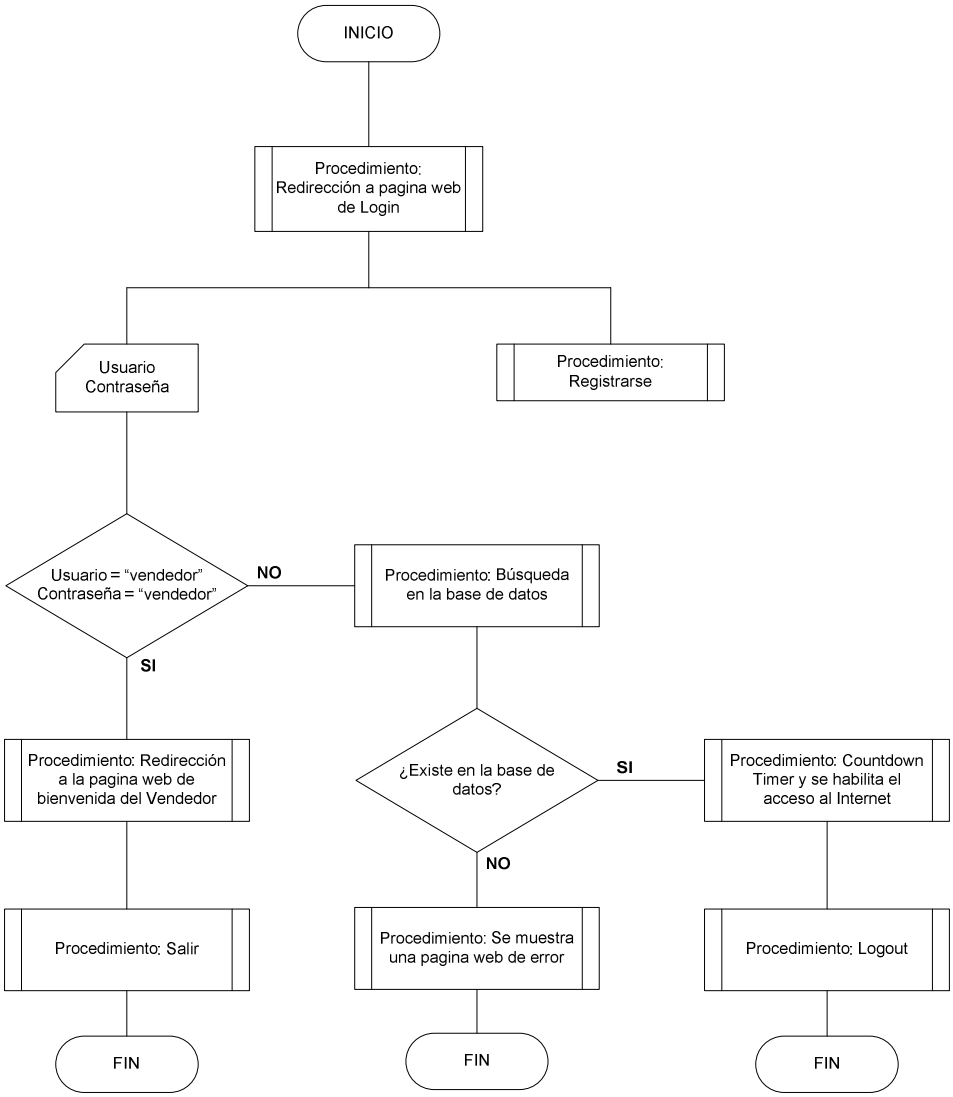


Figura 2.13 Diagrama de Flujo del Portal Cautivo

2.4 IMPLEMENTACIÓN DEL PORTAL CAUTIVO

La computadora del Portal Cautivo debe tener 2 tarjetas de red. Una tarjeta es usada como gateway (puerta de enlace) de la red. Si el tráfico es permitido, éste es redireccionado a la segunda tarjeta de red, la cual nos sirve de conexión a Internet. Entre la tarjeta de red y el Internet se puede usar Router, Bridge, Modems (xDSL[76], ISDN[77], Cable…) depende del proveedor de servicios de Internet (ISP).

En las Figuras 2.14 y 2.15 se indican dos tipos de topologías de red muy utilizadas a la hora de implementar un Portal cautivo.

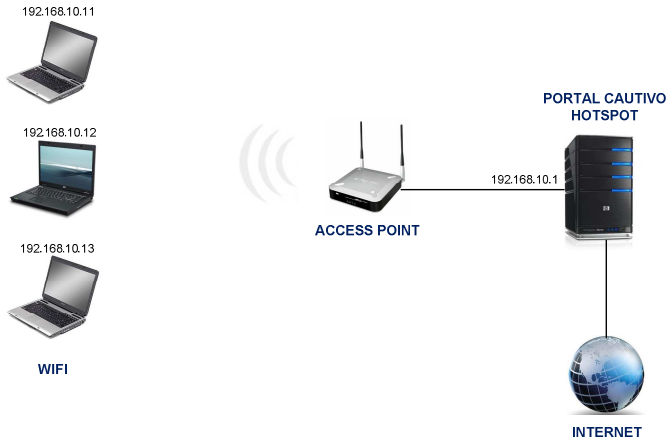


Figura 2.14 Topología de red Básico

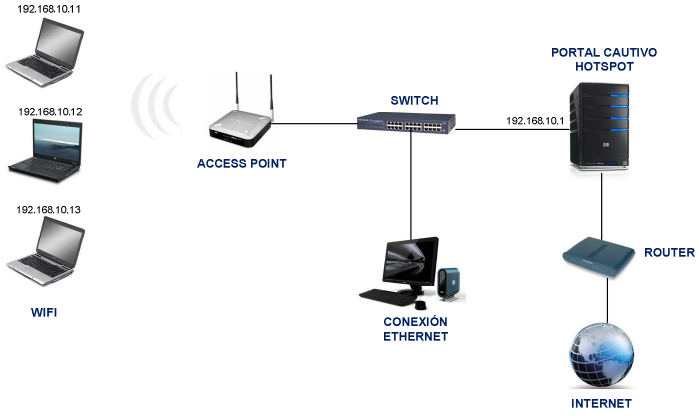


Figura 2.15 Topología de Red Completo

[76] xDSL: Protocolos de banda ancha de Internet (*Digital Subscriber Line*).
[77] ISDN: Red Digital de Servicios Integrados (*Integrated Services Digital Network*).

2.4.1 DIRECCIONAMIENTO

A cada usuario que esté conectado al Portal Cautivo, se le asigna una dirección IP automáticamente. Dependiendo de la cantidad de usuarios que tenga la red, se debe seleccionar un rango de direcciones IP adecuado.

Se puede seleccionar un rango de direcciones IP, ya sea por la clase de dirección IP (Tabla 2.14) o creando una subred, todo esto depende del número de usuarios por red.

Clase	Rango de Direcciones IP		Máscara	Número de Redes	Número de host/red
	Desde	Hasta			
A	10.0.0.0	10.255.255.255	255.0.0.0	1	16777214
B	172.16.0.0	172.31.255.255	255.255.0.0	16	65534
C	192.168.0.0	192.168.255.255	255.255.255.0	256	254

Tabla 2.14 Rango de Direcciones IP Privadas para Intranet (RFC1597)[78]

A la hora de escoger una dirección de red, es conveniente (no obligatorio) reservar un pequeño grupo de direcciones, las cuales van a ser usadas para fines de administración, por ejemplo: una dirección IP para el administrador de la red, para los vendedores de los tickets del Portal cautivo, para futuros servidores, entre otras.

En la Figura 2.16 se indica un ejemplo de asignación de direcciones IP, reservando un grupo de direcciones para administración.

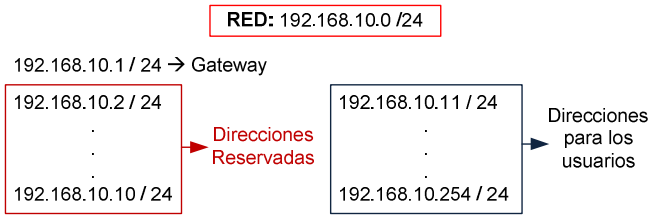


Figura 2.16 Ejemplo de Asignación de Direcciones IP, reservando un grupo de direcciones para administración

[78] http://www.faqs.org/rfcs/rfc1597.html

Generalmente se reserva un grupo de direcciones (10 o menos), al inicio o al final de todo el rango de direcciones.

Para la implementación del Portal Cautivo se toma en cuenta el Diagrama de Red de la Figura 2.17, en el cual se especifican: tipo de tarjetas de red (eth0, eth1) y asignación de direcciones de red (dirección IP del Gateway, direcciones a ser asignadas por DHCP).

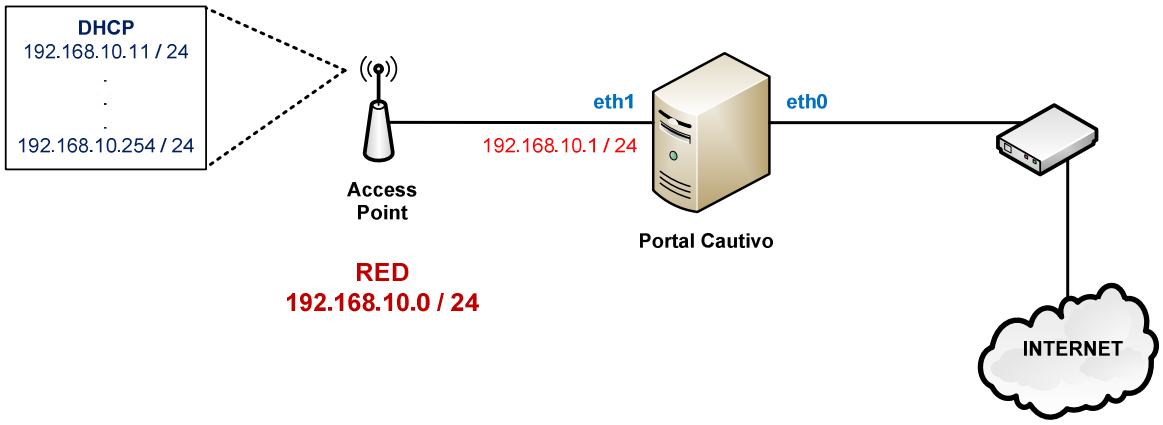


Figura 2.17 Diagrama de Red utilizado para el Portal Cautivo

2.4.2 ELEMENTOS DE UN PORTAL CAUTIVO

Como se mencionó anteriormente un Portal Cautivo es una agrupación de varios elementos. A continuación se presenta una descripción más detallada de los elementos a implementarse en el Portal Cautivo.

2.4.2.1 Firewall

Un firewall o cortafuegos es un dispositivo o programa de seguridad que está diseñado y configurado para permitir, limitar, cifrar, descifrar el tráfico que intente pasar a través de él, bloqueando todos los accesos que no cumplan con los criterios de seguridad configurados.

Los cortafuegos se utilizan con frecuencia para evitar que los usuarios de Internet no autorizados tengan acceso a redes privadas conectadas a Internet,

especialmente Intranets. Todos los mensajes que entren o salgan de la intranet pasan a través del cortafuego, que examina cada mensaje y bloquea aquellos que no cumplan los criterios de seguridad especificados.

En la Figura 2.18 se indica la representación gráfica de un firewall para brindar seguridad a una red LAN con DMZ[79].

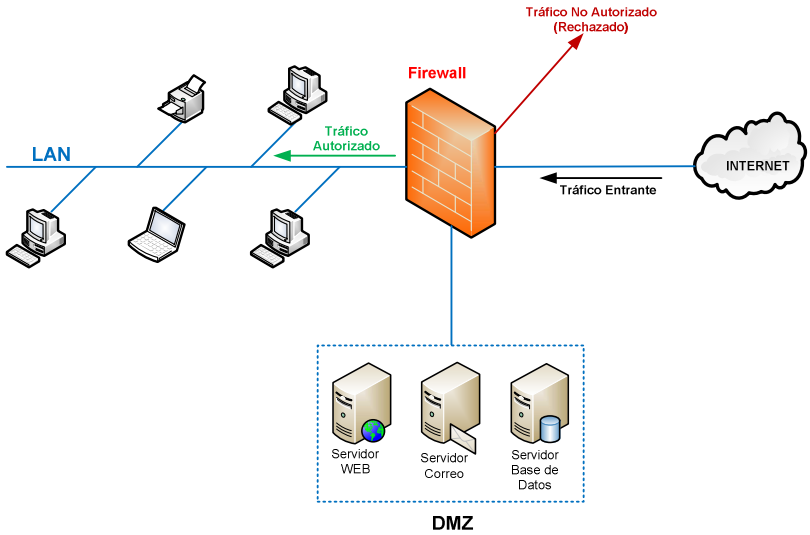


Figura 2.18 Representación del Firewall para brindar Seguridad a una red LAN con DMZ

También es frecuente conectar al cortafuego las denominadas zonas desmilitarizadas o DMZ, en las cuales se ubican todos los servidores de la organización que son accesibles desde Internet.

2.4.2.1.1 Alternativas de Software

El software por defecto que viene en casi todas las distribuciones Linux (desde la versión 2.4 del kernel) es el paquete iptables[80].
No existen otras opciones destacadas para la configuración de firewalls en Linux, pero existen paquetes adicionales que se pueden emplear como frontends de iptables, los cuales se ubican entre el usuario e iptables y buscan simplificar la

[79] DMZ: Zona Desmilitarizada (*DeMilitarized Zone*).
[80] http://www.netfilter.org/

configuración de los firewalls, entre estos paquetes se encuentran: Guarddog, Kmyfirewall, FWBuilder, Knetfilter, Firestarter, Shorewall, entre otros.

2.4.2.1.2 Selección de la Alternativa

La opción que se empleará para permitir, bloquear y redireccionar todo el tráfico de los clientes del Portal Cautivo es iptables, debido a que es la solución más ampliamente usada y documentada para los entornos GNU/Linux.

Es recomendable evitar el uso de frontends, debido a que nos son paquetes necesarios para nuestra aplicación, por lo que solo vendrían a consumir recursos del sistema.

2.4.2.1.3 Instalación de iptables

El servicio iptables suele venir incluido en todas las distribuciones Linux (versión del kernel superior a 2.4), caso contrario para instalar basta ubicar el paquete correspondiente iptables en el administrador de paquetes o emplear el comando yum install iptables; se puede también descargar una copia del paquete de la página oficial del proyecto http://www.netfilter.org/projects/iptables/index.html.

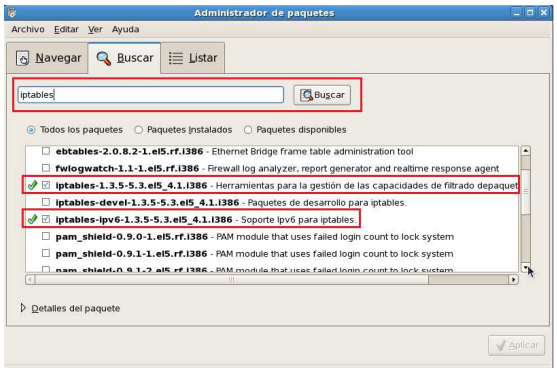


Figura 2.19 Verificación de que el paquete iptables está instalado, mediante el Administrador de Paquetes

Para verificar que está instalado iptables, se dirige a Aplicaciones → Agregar/Quitar Software → Buscar, y buscamos el paquete iptables. Como se

muestra en la Figura 2.19, el paquete iptables ya viene instalado por defecto en CentOS 5.5, en las dos versiones; tanto para ipv4 como para ipv6.

Otra forma de verificar que el paquete iptables está instalado, es dirigirse al directorio /etc/init.d. En la Figura 2.20 se muestran todos los servicios que tiene nuestro sistema operativo.

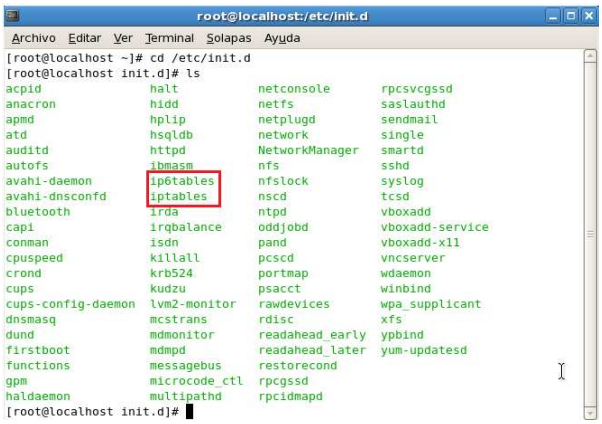


Figura 2.20 Verificación de que el paquete iptables está instalado, dirigiéndose al directorio /etc/init.d

Las herramientas mencionadas anteriormente permiten instalar un programa con todas sus dependencias, ya que previo a la instalación resuelve todas las dependencias que necesita, si las tiene continua con la instalación del programa caso contrario se descarga e instala primero las dependencias necesarias y luego el programa, evitando así descargarse e instalar una por una dichas dependencias.

2.4.2.1.4 Configuración

Para la configuración del firewall se procedieron a configurar las siguientes reglas de filtrado de paquetes (Figura 2.21):

```
## FLUSH DE REGLAS
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

## POLITICAS POR DEFECTO
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

## SE PERMITE EL LOCALHOST
iptables -A INPUT -i lo -j ACCEPT

## AL FIREWALL SE TIENE ACCESO DESDE LA RED LOCAL
iptables -A INPUT -s 192.168.10.0/24 -i eth1 -j ACCEPT

## ENMASCARAMIENTO
iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o eth0 -j MASQUERADE

## SE ACTIVA EL BIT DE FORWARING
## Con esto se permite hacer forward de paquetes
## Es decir que otras máquinas puedan salir a través del firewall
echo 1 > /proc/sys/net/ipv4/ip_forward

## SE CAPTURA EL TRAFICO HTTP Y SE REENVIA AL SERVER INTERNO
iptables -t nat -A PREROUTING -s 192.168.10.0/24 -p tcp --dport 80 -j DNAT --to 192.168.10.1
iptables -t nat -A PREROUTING -s 192.168.10.0/24 -p tcp --dport 443 -j DNAT --to 192.168.10.1

## SE DENIEGA FORWARD DE LA RED LOCAL
iptables -A FORWARD -s 192.168.10.0/24 -i eth1 -j DROP

## SE CREA CADENA USUARIOPERMITIDO
iptables -t nat -N USUARIOPERMITIDO
iptables -t nat -A USUARIOPERMITIDO -j RETURN
iptables -t nat -I PREROUTING 1 -j USUARIOPERMITIDO

## SE PERMITE SALIDA AL INTERNET A 192.168.10.2
## Tiene la información de las Tarjetas de Crédito
iptables -I FORWARD -s 192.168.10.2 -j ACCEPT
iptables -t nat -I USUARIOPERMITIDO -s 192.168.10.2 -j ACCEPT
```

Figura 2.21 Ejemplo de reglas de filtrado de paquetes.

Dichas reglas deben estar guardadas en memoria, caso contrario se perderán al momento de reiniciar el sistema operativo, para evitar esto se debe ejecutar el comando iptables-save.

2.4.2.2 Servidor DHCP

Protocolo de configuración dinámica de host DHCP (Dynamic Host Configuration Protocol) es un protocolo de red que permite la asignación dinámica y automática de todos los parámetros de configuración de una red, es decir, dirección IP, máscara de red, ruta de enlace predeterminada (gateway), dirección del DNS, entre otros.

Para brindar este servicio es necesario configurar un servidor DHCP que contenga una lista de direcciones IP y todos los parámetros requeridos para acceder a la red. Cuando un cliente se conecta al servidor se le asigna una de las direcciones disponibles por un tiempo determinado.

Este servicio facilita enormemente el proceso de configuración y administración de una red, ya que evita la necesidad de asignar direcciones IP a cada una de las computadoras que conforman la red, además que permite un control centralizado de las mismas.

En el caso del Portal Cautivo, cualquier cliente que se conecte a la red, se le asignará una dirección IP automáticamente dentro del rango ya establecido.

2.4.2.2.1 Alternativas de software

A continuación se presentan dos alternativas de software para la configuración del servidor DHCP en Linux. Las opciones escogidas se realizaron en función de su difusión y soporte.

DHCP SERVER ISC[81]: es la herramienta de código abierto más usada para la implementación de un servidor DHCP en LINUX. Tiene las siguientes características:

- Soporte para múltiples redes
- Soporta IPv6
- Actualización dinámica de DNS
- Asignación de direcciones por MAC.
- Soporta BOOTP[82].

DUAL DHCP DNS SERVER[83]: Permite asignar direcciones IP automáticamente (servidor DHCP) y resolver nombres de dominio (servidor DNS). Tiene las siguientes características:

- Combina el uso de DHCP y DNS.
- Código abierto, gratuito.
- Soporta hasta 32 redes.
- Asignación de direcciones por MAC.
- Soporte para DHCP Relay.
- Despliega el estado del servicio mediante una página HTML
- Solo IPv4.

2.4.2.2.2 Selección alternativa

De las dos alternativas antes mencionadas, se ha escogido la opción de DHCP Server ISC por cuanto constituye la versión más usada en servidores Linux y cuenta con la mayor cantidad de documentación de soporte para configuración, mantenimiento y solución de errores. Además de su larga trayectoria de uso muestra ser una alternativa robusta y de fácil configuración.

[81] http://www.isc.org/software/dhcp
[82] BOOTP: (*BOOTstrap Protocol*) Es un protocolo de red UDP utilizado por los clientes de red para obtener su dirección IP automáticamente.
[83] http://dhcp-dns-server.sourceforge.net/

2.4.2.2.3 Instalación de DHCP Server ISC.

La instalación del servicio DHCP se puede realizar añadiendo el paquete en el administrador de paquetes, empleando el comando yum install dhcpd.

En la Figura 2.22 se indica la instalación de DHCP SERVER ISC mediante el Administrador de Paquetes.

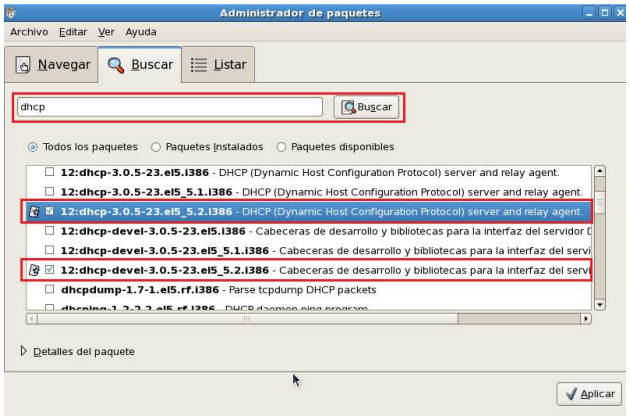


Figura 2.22 Instalación de DHCP SERVER ISC mediante el Administrador de paquetes

Si la instalación se realiza mediante el Administrador de paquetes, previamente se debe completar los siguientes cuadros de dialogo:

- Selección de paquetes, en la Figura 2.23 se indica que paquetes se van a instalar o desinstalar.

Figura 2.23 Selección de Paquetes

- Resolviendo dependencias, esta ventana no es obligatoria a mostrarse, depende del programa que se va a instalar; si necesita alguna

dependencia, la ventana muestra una lista de dependencias que necesita instalar, caso contrario pasa a la siguiente ventana.

- Descargando paquetes, en la Figura 2.24 muestra el progreso de la descarga del programa a instalar.

Figura 2.24 Descarga de Paquetes.

- Se ha completado la instalación del software, si la instalación se realizó con éxito se mostrará el cuadro de diálogo de la Figura 2.25

Figura 2.25 Finalización de la instalación

2.4.2.2.4 Configuración de DHCP Server ISC

La configuración de DHCP se basa en un fichero de texto, /etc/dhcpd.conf que el proceso servidor lee al inicio. La lectura del fichero de configuración sólo se realiza durante el inicio, nunca cuando ya está en ejecución, por tanto cualquier modificación requiere detener el servicio DHCP y volverlo a iniciar.

El dhcpd.conf es un fichero de texto, donde cada línea que comience por # indica un comentario y no se tiene en cuenta. Las distintas entradas de este fichero terminan en ";". Si una entrada de configuración necesita distintos parámetros se los puede agrupar mediante { y }. El fichero contendrá varias líneas de configuración de la forma: *parámetro valor* o *option parámetro valor*. El valor dependerá del parámetro que se necesite configurar, podrá ser un valor lógico (on

u off por ejemplo), una dirección, un nombre predefinido u otro valor según el caso.

En este fichero también se definen las subredes en las que actúa el servidor DHCP y que rangos de direcciones puede asignar.

En la Figura 2.26 se muestra el archivo de configuración dhcpd.conf del Portal Cautivo.

```
#
#
# ARCHIVO DE CONFIGURACION DEL SERVIDOR DHCP PARA EL PORTAL CAUTIVO
#
#
ddns-update-style interim;
ignore client-updates;

subnet 192.168.10.0 netmask 255.255.255.0 {

# --- default gateway
        option routers                  192.168.10.1;
        option subnet-mask              255.255.255.0;

        option domain-name              "wifihotspot.com";
        option domain-name-servers      192.168.10.1;

        range 192.168.10.5 192.168.10.250;
        default-lease-time 21600;
        max-lease-time 43200;

}
```

Figura 2.26 Archivo de configuración dhcpd.conf

Descripción del archivo de configuración:

- ddns-update-style interim; esta línea indica el método de actualización DNS automático con los valores de la IP asignados por DHCP.
- ignore client-updates; como su nombre lo indica ignora todas las actualizaciones que los clientes quieran realizar al servidor DHCP.
- subnet 192.168.10.0 netmask 255.255.255.0 { → Definición de la red. Dirección de red y máscara de red.

- option routers 192.168.10.1; Parámetro del DHCP indicando la puerta de enlace.
- option subnet-mask 255.255.255.0; Parámetro del DHCP indicando la máscara de subred.
- option domain-name wifihotspot.com; Parámetro del DHCP indicando el nombre del servidor DNS.
- option domain-name-servers 192.168.10.1; Parámetro del DHCP indicando la dirección IP del servidor DNS.
- range 192.168.10.5 192.168.10.250; Desde qué IP hasta qué IP asigna el servidor DHCP.
- default-lease-time 21600; Es el tiempo hasta el primer aviso que le presta la IP al cliente DHCP.
- max-lease-time 43200; Es el tiempo máximo de préstamo de IP al cliente DHCP.

Como se tiene dos tarjetas de red, es indispensable que el servicio dhcp solo funcione a través de la interfaz de red utilizada por la LAN. Para ello, se edita el fichero /etc/sysconfig/dhcpd y se agrega como argumento del parámetro DHCPDARGS el valor eth0, eth1, eth2, etc., o lo que corresponda. (Figura 2.27)

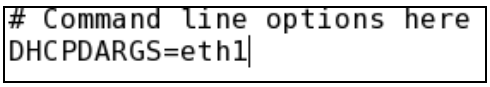


Figura 2.27 Fichero /etc/sysconfig/dhcpd

Para ejecutar por primera vez el servicio:

/etc/init.d/service dhcpd start

Para detener el servicio:

/etc/init.d/service dhcpd stop

Para añadir dhcpd al arranque del sistema:

/etc/init.d/chkconfig dhcpd on

Como el servidor DHCP puede pararse o reiniciarse, necesita mantener una lista de direcciones asignadas. El fichero /var/lib/dhcp/dhcpd.leases mantiene esta lista de asignaciones.

2.4.2.3 Servidor LAMP

LAMP se refiere a un conjunto de subsistemas de software necesarios para a configurar sitios web y servidores dinámicos de una manera sencilla, esto se consigue mediante la unión de las siguientes tecnologías:

- **L**inux, es un núcleo de sistema operativo libre tipo Unix.
- **A**pache[84], el Servidor HTTP Apache es un servidor web libre y de código abierto, el más popular en cuanto a uso, sirviendo de facto como plataforma de referencia para el diseño y valoración de otros servidores web.
- **M**ySQL[85], es un Sistema de Gestión de Bases de Datos (SGBD) relacional, que por lo tanto utiliza SQL, multihilo y multiusuario.
- **P**HP[86], Hypertext Preprocessor es un lenguaje de programación diseñado para producir sitios web dinámicos. PHP es utilizado en aplicaciones del lado del servidor, aunque puede ser usado también desde una interfaz de línea de comandos o como aplicación de escritorio.

En la Figura 2.28 se indican los diferentes logotipos que conforman LAMP.

Figura 2.28 Logotipos de los subsistemas que conforman LAMP

[84] http://www.apache.org/
[85] http://www.mysql.com/
[86] http://www.php.net/

2.4.2.3.1 Servidor HTTP Apache[87]

Es un servidor Web de código abierto para plataformas Unix, Microsoft, Macintosh entre otras que implementa el protocolo HTTP. Es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache tiene amplia aceptación en la red, desde 1996 es el servidor HTTP más usado. Algunos de los más grandes sitios web del mundo están ejecutándose sobre Apache.

2.4.2.3.1.1 Instalación

La instalación de apache web server se puede realizar añadiendo el paquete en el administrador de paquetes, empleando el comando yum install httpd o simplemente mediante la descarga del paquete en http://www.apache.org/. En la Figura 2.29 se indica la instalación de httpd mediante el administrador de paquetes.

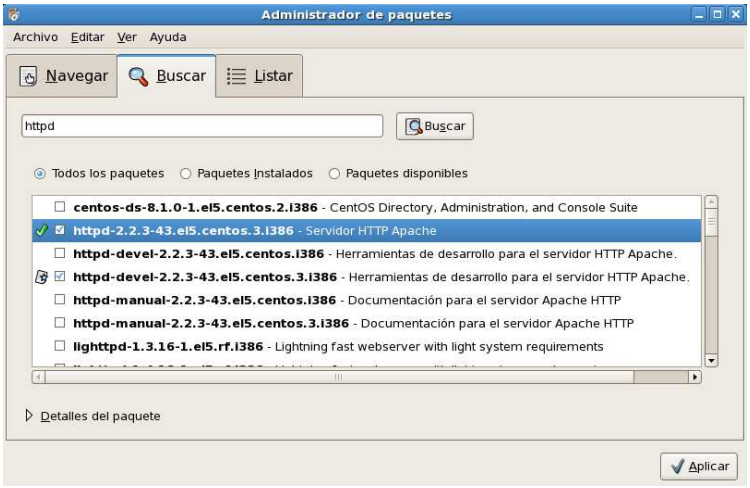


Figura 2.29 Instalación de httpd mediante el Administrador de Paquetes

[87] http://httpd.apache.org/ABOUT_APACHE.html

En las Figura 2.30 se indica los diferentes cuadros de dialogo que se muestran al instalar el paquete httpd.

Figura 2.30 Descarga e instalación de los paquetes necesarios para httpd

2.4.2.3.1.2 Configuración

Para la configuración del servidor HTTP Apache, se utilizará la herramienta Webmin (Ver Anexo C)

Previo a crear el host virtual es necesario copiar el directorio que contenga las páginas web a la ubicación /var/www/ y cambiar los permisos de todo el directorio mediante el comando chmod -R 755 portalcautivo (Figura 2.31)

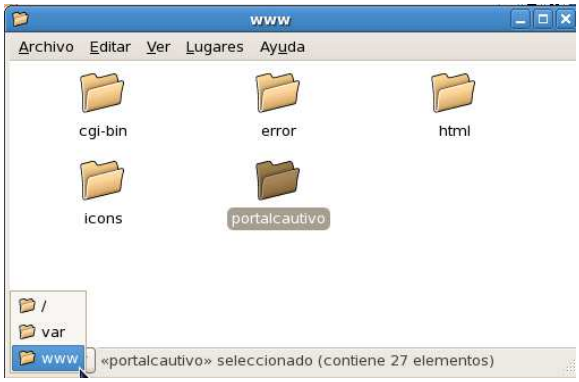


Figura 2.31 Ubicación del directorio que contiene todas las páginas web del Portal Cautivo

En la Figura 2.32 se muestra la pantalla de inicio de Apache Web Server, seleccionado en el Webmin.

Figura 2.32 Pantalla de inicio de Apache Webserver en el Webmin

Se selecciona Servers → Apache Webserver → Create virtual host, para entrar en la configuración de un nuevo virtual host, en el cual se debe ingresar: una dirección IP, el puerto que va a utilizar, el path en donde están las páginas web, el nombre del servidor (Figura 2.33)

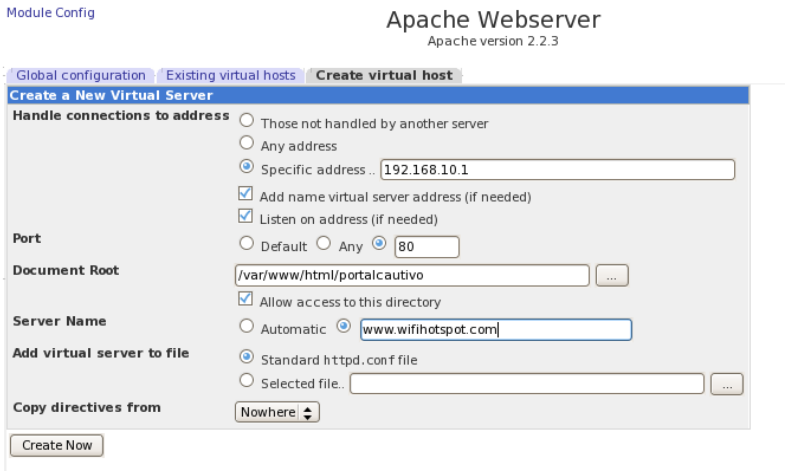


Figura 2.33 Creación de un Virtual Host

Para guardar los cambios efectuados se debe dar un clic en: create Now y Apply Changes. Con esto se completa la creación del virtual host.

2.4.2.3.2 Lenguaje de Programación PHP[88]

Es un lenguaje de programación principalmente en interpretación del lado del servidor (server-side scripting), diseñado originalmente para la creación de páginas web dinámicas. Actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

[88] http://www.php.net/

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con un nivel de aprendizaje muy simple. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones. PHP no es en sí un lenguaje de programación orientado a objetos, pero desde las últimas versiones se tiene un mejor soporte para la programación orientada a objetos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

El programa está liberado bajo la licencia GNU, fácil de usar y capaz de interpretar páginas dinámicas, tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX y Windows, y puede interactuar con los servidores web más populares como Apache.

2.4.2.3.2.1 Instalación

La instalación de PHP se puede realizar añadiendo el paquete en el administrador de paquetes, empleando el comando yum install php o simplemente mediante la descarga del paquete de la página del proyecto en http://www.php.net/. En la Figura 2.34 se indica la instalación de PHP mediante el administrador de paquetes.

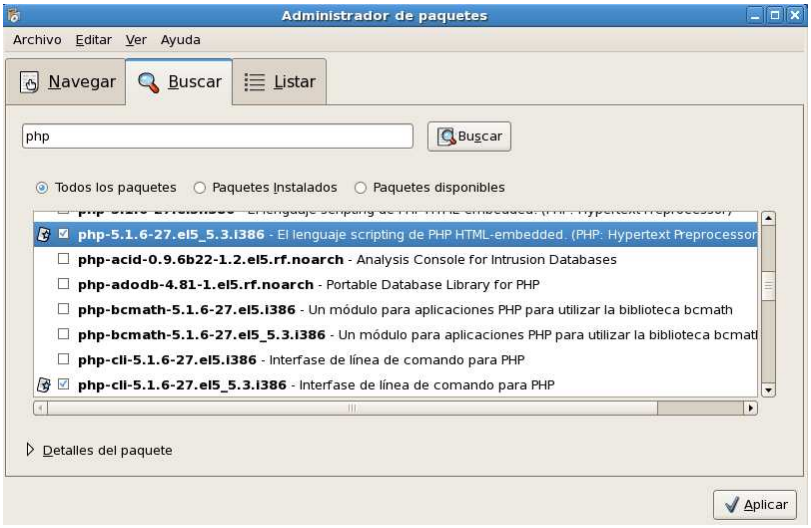


Figura 2.34 Instalación de PHP mediante el Administrador de paquetes

No solamente el paquete PHP hay que descargar e instalar, se debe descargar ciertos paquetes que proveen la conexión con la base de datos MySQL. En la Figura 2.35 se muestra dichos paquetes:

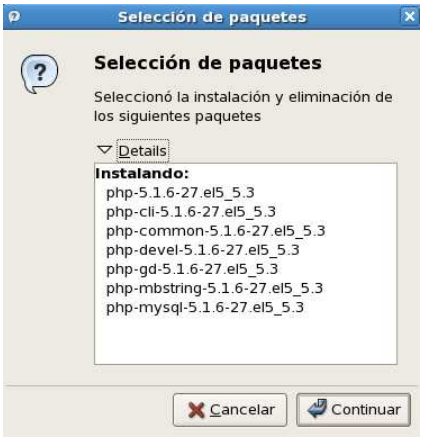


Figura 2.35 Paquetes adicionales (necesarios) para la instalación de PHP

2.4.2.3.2.2 Configuración

La configuración de PHP se la realiza en el fichero /etc/php.ini. Es un archivo de texto, en el cual se deben realizar dos cambios para el correcto funcionamiento del Portal Cautivo.

1. register_globals=On → Debido a que se utiliza variables de sesión para el registro y autenticación de los usuarios.
2. magic_quotes_gpc=On y magic_quotes_runtime=On → Permite añadir o quitar slashes (/) automáticamente. Esto sirve para la conexión con la base de datos MySQL.

2.4.2.3.3 Servidor de Base de Datos MySQL[89]

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Los servidores de bases de datos permiten manejar grandes y complejos volúmenes de datos, al tiempo que requieren compartir la información con un conjunto de clientes (que pueden ser tanto aplicaciones como usuarios). Un sistema gestor de bases de datos (SGBD) ofrece soluciones de forma fiable, rentable y de alto rendimiento, proporciona servicios de forma global y, en la medida de lo posible, independientemente de la plataforma. Esto hace que las empresas tiendan a presentar su información a través de la Web en forma de contenidos.

Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante detectar y corregir errores tanto de SQL como de programación.

2.4.2.3.3.1 Instalación

La instalación de MySQL se puede realizar añadiendo el paquete en el administrador de paquetes, empleando el comando yum install mysql mysql-server o simplemente mediante la descarga del paquete de la página del proyecto

[89] http://www.mysql.com/

en http://www.mysql.com/downloads/. En la Figura 2.36 se indica la instalación de mysql mediante el administrador de paquetes.

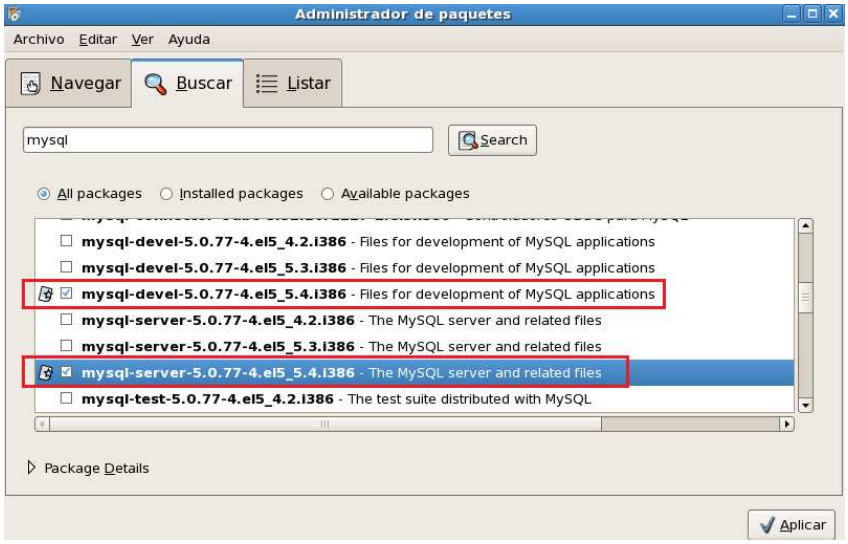


Figura 2.36 Instalación de mysql-server mediante el Administrador de paquetes

Es necesario también instalar el paquete phpmyadmin[90] que es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Se encuentra disponible bajo la licencia GPL[91].

En la Figura 2.37 se indica la instalación de phpmyadmin mediante el administrador de paquetes.

[90] http://www.phpmyadmin.net/home_page/index.php
[91] GPL: licencia orientada principalmente a proteger la libre distribución, modificación y uso de software.

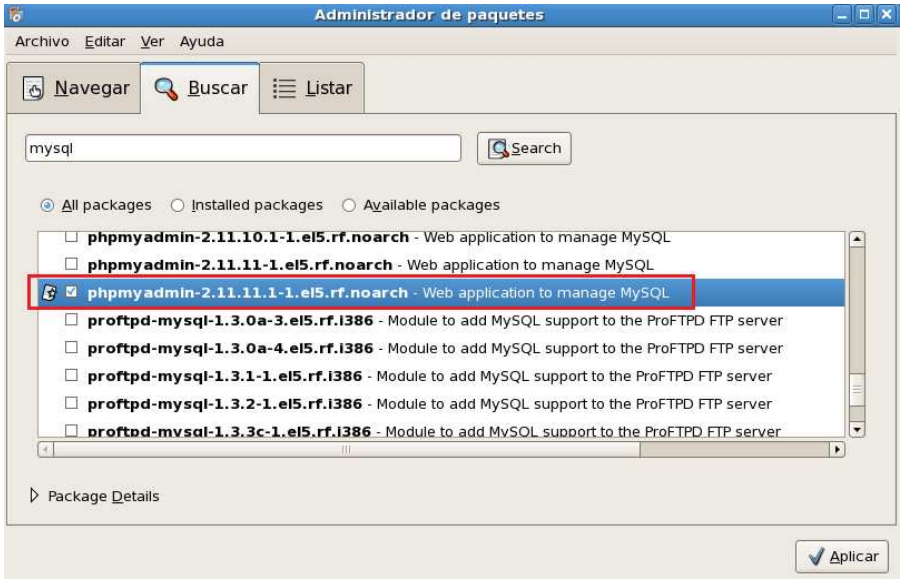


Figura 2.37 Instalación de phpmyadmin mediante el Administrador de paquetes

2.4.2.3.3.2 Configuración de MySQL

Lo primero que hay que configurar en MySQL (previo a la instalación) es asignar una contraseña al usuario root (administrador) debido a que por defecto no lo trae. En la Figura 2.38 se muestra el proceso para cambiar la contraseña del usuario root.

```
[root@server ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> update user set Password=PASSWORD('mysqladmin') where user='root';
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select Host, Password, User from user;
+-----------+-------------------+------+
| Host      | Password          | User |
+-----------+-------------------+------+
| localhost | 6f38971106105e69 | root |
| server    | 6f38971106105e69 | root |
| 127.0.0.1 | 6f38971106105e69 | root |
| localhost |                   |      |
| server    |                   |      |
+-----------+-------------------+------+
5 rows in set (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
Bye
[root@server ~]# service mysqld restart
Parando MySQL:                                             [  OK  ]
Iniciando MySQL:                                           [  OK  ]
[root@server ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

Figura 2.38 Configuración de un password para root en MySQL

Una vez colocado un password al usuario root, se edita el fichero /usr/share/phpmyadmin/config.inc.php, agregando una contraseña para el *blowfish secret*[92] (Figura 2.39)

Figura 2.39 Configuración config.inc.php

Iniciamos el servicio con el comando:

```
/etc/init.d/service mysqld start
```

En la Figura 2.40 y 2.41 se muestran las pantallas de acceso y de bienvenida a phpMyAdmin.

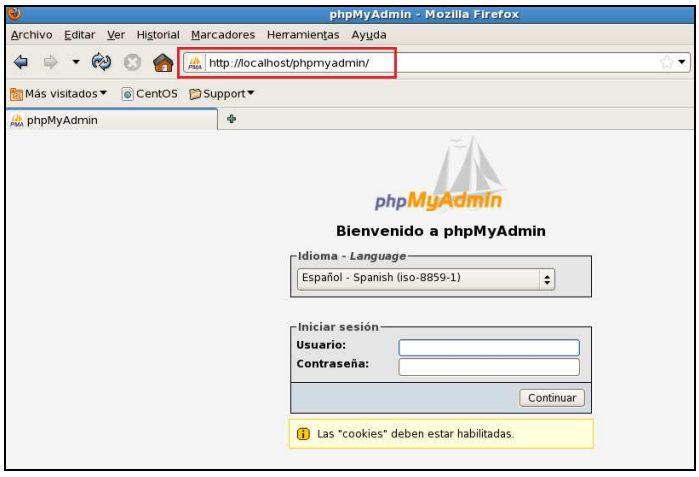


Figura 2.40 Acceso web a phpmyadmin.

[92] Blowfish Secret: La autenticación mediante cookies utiliza el algoritmo Blowfish para encriptar la contraseña. Si en la configuración del servidor se tiene habilitado el uso de autenticación mediante cookies, se debe colocar un *passphrase* (contraseña) que va ser usado por blowfish.

Figura 2.41 Pantalla de Bienvenida de phpMyAdmin

En la Figura 2.42 se indican los principales Directorios usados por MySQL.

/var/lib/mysql/ → Directorio donde se guardan las BDD. /etc/mysql/ → Directorio donde se encuentran los archivos de configuración.

Figura 2.42 Principales Directorios de MySQL

Una vez instalado y probado nuestro phpmyadmin pasamos a crear nuestra base de datos, de acuerdo al diagrama relacional que se muestra en la Figura 2.43

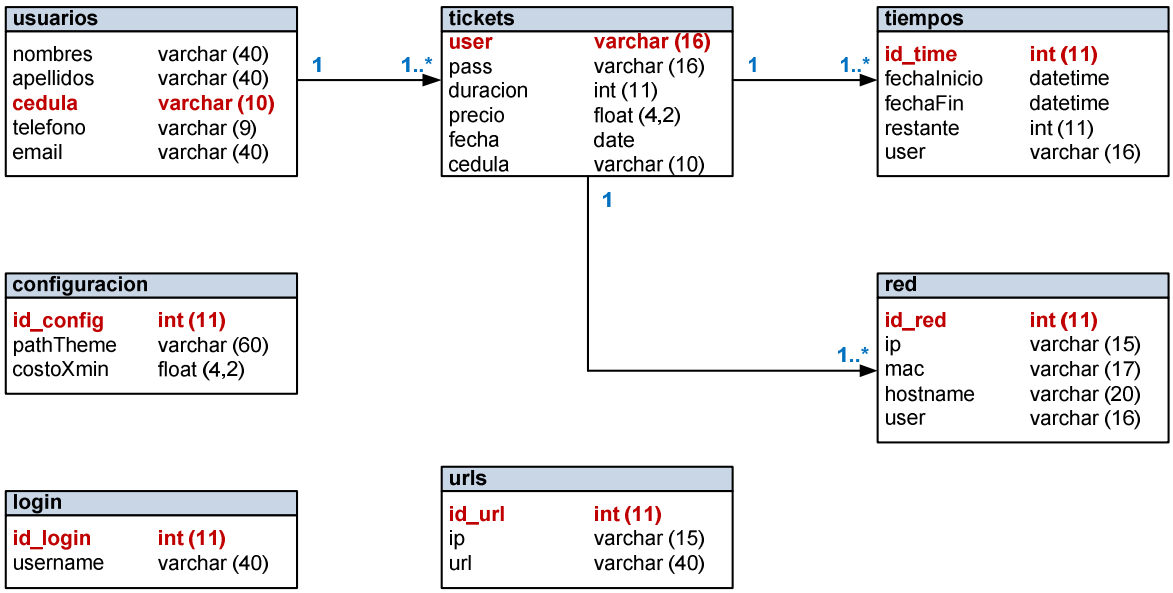


Figura 2.43 Tabla relacional de la estructura de la base de datos

A continuación se describe cada una de las tablas que conforman la base de datos.

- Tabla usuarios: Datos personales de los clientes del Portal Cautivo: nombres, apellidos, cédula de identidad, teléfono y email.
- Tabla tickets: Datos referentes al ticket: nombre de usuario, contraseña, duración, precio, fecha de compra y cédula (para establecer la relación de uno a muchos con la tabla usuarios).
- Tabla tiempos: Datos referentes al inicio y fin de una sesión válida (una vez ingresados el username y password correctos): id, fecha en que inició la sesión, fecha de finalización de la sesión, tiempo restante en segundos del tickets y username (para establecer relación de uno a muchos con la tabla tickets).
- Tabla red: Datos referentes a la configuración de red de los clientes: id, dirección IP, dirección MAC, hostname y username (para establecer relación de uno a muchos con la tabla tickets).
- Tabla configuración: Datos referentes a la configuración del Portal Cautivo: id, el path completo del Theme (tema) que van a usar las páginas Web, el costo por minuto de Internet.
- Tabla login: Datos referentes a un usuario autenticado.
- Tabla urls: Datos referentes a la dirección web que el cliente ingreso.

Se ingresa vía Web a http://localhost/phpmyadmin para crear la base de datos y sus distintas tablas. Se ingresa un nombre para la nueva base de datos (Figura 2.44)

Figura 2.44 Creación de una Base de Datos

A continuación se crea una nueva tabla en la base de datos, ingresando su nombre y el número de campos que tiene (Figura 2.45)

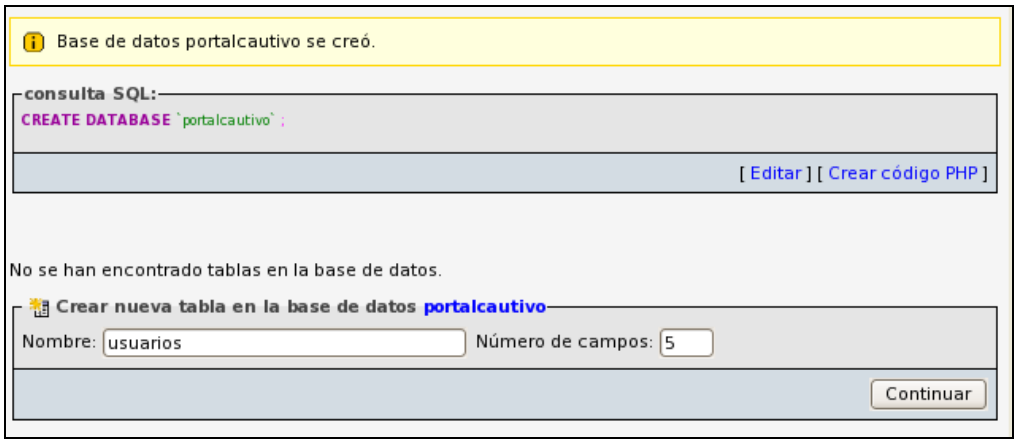


Figura 2.45 Creación de la tabla usuarios

Se ingresa los campos que contendrá la tabla usuario definiendo: el campo, el tipo de dato, la longitud (valores) y si el dato es clave primaria (Figura 2.46 y 2.47)

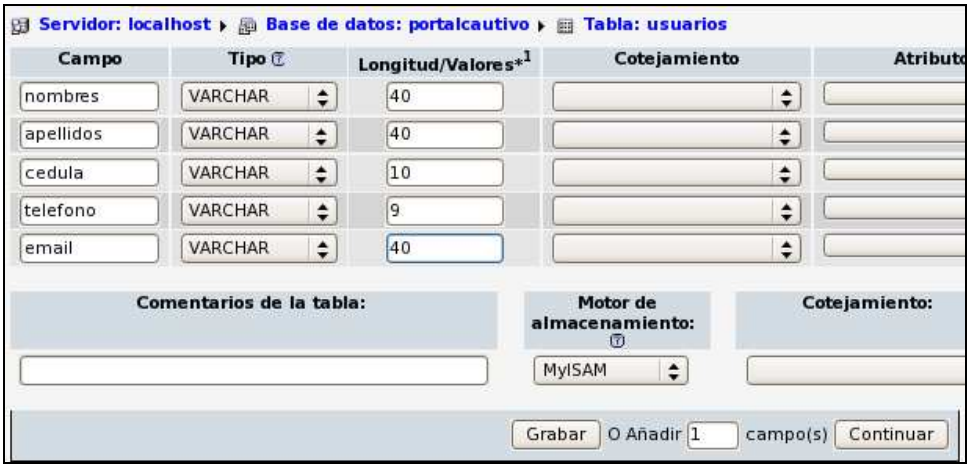


Figura 2.46 Campos de la tabla usuarios

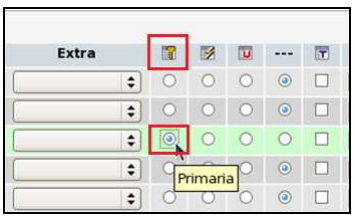


Figura 2.47 Selección de la clave primaria, en nuestro caso es el campo cedula

Ya se tiene la tabla usuarios creada. Los mismos pasos se siguen para crear las demás tablas (Figura 2.48)

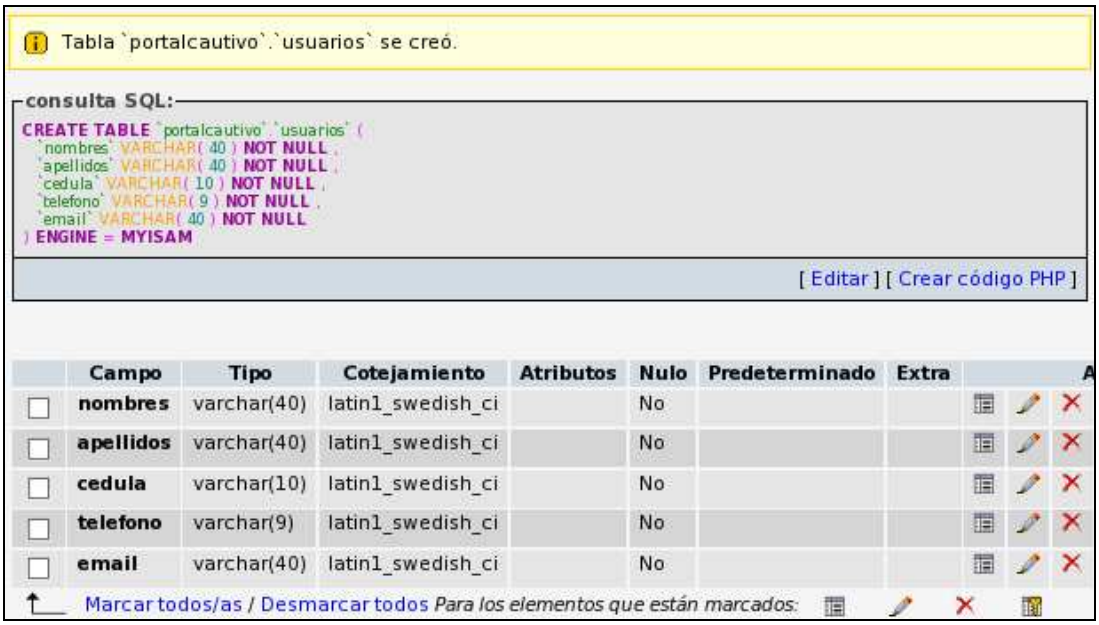


Figura 2.48 Tabla Usuarios

En la Figura 2.49 se indican todas las tablas que conforman la base de datos del Portal Cautivo.

Figura 2.49 Base de datos portalcautivo con sus respectivas tablas

2.4.2.3.4 HTTPS[93]

SSL (*Secure Socket Layer*) es un protocolo cliente-servidor que permite conexiones seguras (confidencialidad, autenticación e integridad) a cualquier protocolo basado en TCP/IP. Se usa normalmente con HTTP (formando HTTPS), para asegurar páginas web de comercio electrónico, entidades bancarias, etc.

El sistema HTTPS[94] utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

El puerto estándar para este protocolo es el 443.

2.4.2.3.4.1 Funcionamiento

El funcionamiento de SSL se detalla a continuación:

1. El cliente solicita una conexión al servidor SSL, que escucha por defecto en el puerto 443 TCP.
2. Cliente y servidor comprueban las versiones del protocolo soportadas.
3. Cliente y servidor acuerdan los algoritmos a utilizar.
4. El servidor envía su clave pública al cliente.
5. El cliente genera la "clave de sesión" (válida sólo para esa sesión) y la envía al servidor encriptada con la clave pública del servidor.
6. A partir de ese momento, todo el tráfico se encripta con cifrado simétrico con la "clave de sesión".

[93] http://www.estrellateyarde.org/discover/ssl
[94] HTTPS: *Hyper Text Transfer Protocol Secure*

7. Las claves públicas admiten certificados en los que se basa la autentificación.
8. Normalmente sólo se autentifica el servidor, mientras que el cliente se mantiene sin autentificar.
9. Se utiliza el valor hash de los datos transmitidos para garantizar la integridad de los mismos.

2.4.2.3.4.2 Servidor Web Apache con soporte SSL

Para preparar un servidor web que acepte conexiones HTTPS, se debe:

- Crear una Autoridad de Certificación.
- Crear un Certificado de clave pública para el servidor web. Este certificado debe estar firmado por la Autoridad de certificación para que el navegador web lo acepte. La autoridad certifica que el titular del certificado es quien dice ser.

2.4.2.3.4.3 Autoridad Certificadora (CA)[95]

Una autoridad certificadora como lo son Verisign, Thawte, beTRUSTed o ValiCert son empresas dedicadas a vender certificados de seguridad que la empresa que lo adquiere instala en su servidor web. Es decir que cualquier empresa que desea montar su aplicación Web bajo un sitio seguro con HTTPS, crea su certificado y lo manda a firmar con un CA, el CA verifica que la empresa es realmente quien dice ser. Después de verificar la autenticidad de la empresa en cuestión, el CA firma el certificado de seguridad de su cliente con alguno de sus certificados raíz bajo una fuerte encriptación y se lo regresa a la empresa, este lo instala en su servidor Web y cuando los clientes (navegadores) se conectan, estarán tanto el cliente como el servidor bajo un tráfico encriptado y seguro, todo avalado por el CA otorgante del certificado.

[95] http://www.linuxtotal.com.mx/index.php?cont=info_seyre_001

La enorme ventaja de este esquema es que todos los navegadores actuales (Internet Explorer, Firefox, Opera, Mozilla, etc) tienen incorporados los certificados raíz de todas las empresas CA conocidas del mundo, así que cuando el cliente se conecta al servidor no hay ninguna molestia para el cliente, todo es transparente para el usuario final.

Ahora bien, los CA como los mencionados, cobran por el servicio de firmar los certificados, sus precios comienzan en alrededor de 300 a 400 dólares anuales por certificado y pueden subir más dependiendo del tipo de encriptación que se solicite, es decir, para un sitio de comercio electrónico con un alto volumen de tráfico requerirá de certificados más seguros debido a que será más tentador para posibles hackers el tratar de violarlo.

El mismo servidor web puede convertirse en CA, el mismo emitir un certificado raíz de seguridad y a través de este generar certificados para sitios Web. Una de las desventajas que se tiene al implementar una CA en el mismo servidor Web es que en el navegador del cliente al no estar importado el certificado (integrado) en su lista de certificados seguros, pedirá al usuario cuando se conecte al sitio que acepte el certificado. Si el usuario es desconfiado y no lo acepta no se podrá conectar al servidor Web seguro.

Se requiere para la creación de una entidad certificadora (CA) dos cosas:

- Un par de claves, una pública y una privada (claves RSA o DSA, en otras palabras encriptación asimétrica). La clave es pública, como su nombre implica es expuesta a todo el mundo y la privada es solo conocida por el emisor, es decir, el servidor web.
- Un certificado de seguridad, que es una versión firmada o verificada de la clave pública RSA o DSA.

Se puede autofirmar la clave pública o enviarla a un tercero, a un CA reconocido para que la firme cobrando por el servicio. Una vez que se hace esto, se tiene en ambos casos como producto un certificado firmado que será el que el navegador

deberá importar en su lista de certificados de confianza para poder establecer la conexión.

2.4.2.3.4.4 Crear certificados SSL para Apache[96]

Para crear los certificados en Linux se usará la aplicación OpenSSL, que viene instalada en la mayoría de distribuciones Linux. Apache además deberá tener instalado el módulo ModSSL.

Todo el trabajo se hará dentro de un directorio de trabajo denominado CA y dentro de este a la vez se debe crear otros dos, llamados certificado y privado. El primero es donde se guardará una copia de cada certificado que se firme y en el otro directorio se guardará la llave privada.

```
#> mkdir CA
#> cd CA
#> mkdir certificado privado
```

Figura 2.50 Creación de los directorios CA, certificado y privado

Es muy importante no perder la llave privada que se generé, ya que con esta se puede firmar o renovar certificados, y mucho menos dársela a nadie, ya que toda la seguridad radica en la confidencialidad de la llave privada que se guardará en el directorio privado.

Lo siguiente será crear un par de archivos que en conjunto formarán la base de datos de los certificados autofirmados (Figura 2.51)

```
#> echo '01' > serial
#> > index.txt
```

Figura 2.51 Creación de los archivos serial e index.txt

[96] http://www.linuxtotal.com.mx/index.php?cont=info_seyre_001

El primer archivo "*serial*" simplemente contiene el siguiente número de serie de los certificados, ya que se va a crear el primero su número de serie será 01, después de crearlo se actualizará a 02 y así sucesivamente.

"*index.txt*" será la base de datos propiamente en base al número de serie.

Openssl tiene docenas de opciones y parámetros, mucha de la información que irá en el certificado es tomado del archivo de configuración, en vez de la línea de comandos. En la Figura 2.52 se muestra un archivo de configuración listo para ser usado. A este archivo se lo nombra como openssl.cnf y se lo guardará dentro del directorio CA. El archivo se divide en secciones indicadas entre [corchetes], y cada sección tiene sus propias variables. La idea principal del archivo de configuración es de simplificar el uso de los subcomados del comando openssl, que tiene tres subopciones principales: ca, req y x509, entonces, cuando se lee el archivo de configuración "*openssl.cnf*" y se usa la opción req por ejemplo, esta opción toma sus argumentos de la sección correspondiente del archivo de configuración.

Hay una directiva o variable importante que es distinguished_name (DN) o nombre distinguido en español, esta a su vez hace referencia a una sección que tiene los datos básicos de la autoridad certificadora (CA) y que también servirán para cuando se generen certificados. Más simple, el DN son los campos que identifican al propietario del certificado.

```
# Archivo de configuración para openssl
#
# ***** openssl.cnf ******

dir = .          # variable que establece el directorio de trabajo

# sección que permite convertirse en una CA
# solo se hace referencia a otra sección CA_default
[ ca ]
default_ca = CA_default


[ CA_default ]
serial = $dir/serial          # archivo que guarda el siguiente número de serie
database = $dir/index.txt          # archivo que guarda la base de datos de certificados
new_certs_dir = $dir/certificados          # directorio que guarda los certificados generados
```

```
certificate = $dir/cacert.pem          # nombre del archivo del certificado raíz
private_key = $dir/privado/cakey.pem          # llave privada del certificado raíz
default_md = sha1          # algoritmo de dispersión usado
preserve = no          # Indica si se preserva o no el orden de los campos del DN
                       # cuando se pasa a los certificados
nameopt = default_ca          # esta opción y la siguiente permiten mostrar detalles del certificado
certopt = default_ca
policy = policy_match          # indica el nombre de la sección donde se especifica que campos son
                               # obligatorios, opcionales y cuáles deben ser iguales al certificado raíz

# sección de políticas para la emisión de certificados
[ policy_match ]
countryName = match          # match, obligatorio
stateOrProvinceName = match
organizationName = match
organizationalUnitName = optional          # optional, campo opcional
commonName = supplied          # supplied, debe estar en la petición
emailAddress = optional

# sección que indica como los certificados deben ser creados
[ req ]
default_bits = 1024          # tamaño de la llave, si no se indica 512
default_keyfile = key.pem          # nombre de la llave privada
default_md = sha1          # algoritmo de dispersión a utilizar
string_mask = nombstr          # caracteres permitidos en la máscara de la llave
distinguished_name = req_distinguished_name          # sección para el nombre distinguido (DN)
req_extensions = v3_req          # sección con más extensiones que se añaden a la
                                 # petición del certificado

# sección del nombre distinguido, el valor es el prompt que se verá en pantalla.
# datos del propietario del certificado.
# esta sección define el contenido de datos de identificación que el certificado llevara.
[ req_distinguished_name ]
0.organizationName = Nombre de la organizacion
0.organizationName_default = Wifi HotSpot
organizationalUnitName = Departamento o division
emailAddress = Correo electronico
emailAddress_max = 40
localityName = Ciudad o distrito
localityName_default = Quito
stateOrProvinceName = Estado o provincia
stateOrProvinceName_default = Pichincha
countryName = Codigo del pais (dos letras)
countryName_default = EC
countryName_min = 2
countryName_max = 2
commonName = Nombre comun (hostname o IP)
commonName_max = 64

# si en la línea de comandos se indica la opción -x509,
# las siguientes extensiones también aplican
[ v3_ca ]
# indica que se trata de un certificado CA raíz con autoridad para
# firmar o revocar otros certificados
basicConstraints = CA:TRUE

# especifica bajo que método identificar a la llave publica que será certificada
subjectKeyIdentifier = hash
```

```
# especifica como identificar la llave publica
authorityKeyIdentifier = keyid:always,issuer:always

# extensiones de la opcion req
[ v3_req ]
basicConstraints        = CA:FALSE  # los certificados firmados no son CA
subjectKeyIdentifier    = hash
```

Figura 2.52 Archivo de configuración openssl.cnf

El archivo openssl.cnf debe ser guardado en el directorio CA. En el directorio CA se debe tener los directorios y archivos que se muestran en la Figura 2.53

```
#> ls -l
drwxr-xr-x  2 root root 4096 may 28  15:34 certificados
-rw-r--r--  1 root root    0 may 28 15:34 index.txt
-rwxr--r--  1 root root 4776 may 28  2011 openssl.cnf
drwxr-xr-x  2 root root 4096 may 28 15:34 privado
-rw-r--r--  1 root root    3 may 28 15:34 serial
#>
```

Figura 2.53 Contenido del directorio CA

Todo está listo para crear el certificado raíz, este certificado es el que convertirá al servidor web en una autoridad certificadora CA, así que cuando se emita el comando lo primero que nos pedirá es el "passphrase" o más llanamente, una contraseña pero en forma de una frase. Esta contraseña es de vital importancia ya que es con la que se valida la autoridad para después poder crear certificados autofirmados que son los que realmente se van a usar en el servidor web, debe ser preferentemente muy compleja, con mayúsculas, minúsculas, espacios, números y por supuesto símbolos, un buen ejemplo sería el que se muestra en la Figura 2.54

```
(=P0r la as+p3rA p3n%di3nt3 #haciA_la cum+bre=)
```

Figura 2.54 *Passphrase*

Puede parecer muy complicada para recordar y lo es, pero se tiene en cuenta que los algoritmos de cifrado son muy buenos y sumamente difíciles para romper mediante fuerza bruta, así que la verdadera debilidad es el uso de contraseñas débiles. Se recomienda como "passphrase" algo similar a lo anterior y al menos 20 caracteres.

A continuación se procede a crear 2 archivos:

- Un certificado raíz CA (cacert.pem)
- Una llave privada (privado/cakey.pem)

El comando para crear dichos archivos se muestra en la Figura 2.55

```
#> openssl req -new -x509 -extensions v3_ca -keyout privado/cakey.pem \
-out cacert.pem -days 3650 -config ./openssl.cnf

Generating a 1024 bit RSA private key
....++++++
.......++++++
writing new private key to 'privado/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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.
-----
Nombre de la organizacion [Wifi HotSpot]:
Departamento o division []:Redes
Correo electronico []:root@wifihotspot.com
Ciudad o distrito [Quito]:
Estado o provincia [Pichincha]:
Codigo del pais (dos letras) [EC]:
Nombre comun (hostname o IP) []:192.168.10.1
```

Figura 2.55 Creación del certificado raíz y la llave privada

Antes de analizar la salida, a continuación se muestran las opciones indicadas:

- req -new -x509 → crear un certificado nuevo autofirmado.
- -extensions v3_ca → crear un certificado raíz CA.
- -keyout → nombre y donde guardará la llave privada.
- -out → nombre del certificado raíz CA.
- -days 3650 → el certificado será válido por 3650 días (10 años).
- -config → archivo de configuración a utilizar.

Con respecto al resultado producido, lo primero que se indicó fue escribir y verificar la contraseña, después vienen los datos para identificar al propietario del

certificado CA, que como se puede apreciar los prompts y los datos por default provienen del archivo de configuración. Si no se especifica la opción -days entonces el certificado será válido por solo 30 días.

Hasta aquí ya se tiene un certificado raíz que válida al servidor como CA, los siguientes procedimientos son los que a continuación hay que realizar:

- Crear una llave privada (key.pem) y una solicitud de certificado[97] (wifi-cert.pem)
- Firmar la solicitud para generar un certificado autofirmado (certificado-wifi.pem)

Se vuele entonces a usar el comando openssl para lograr lo anterior (Figura 2.56) Solo que en la solicitud de firmado no es necesario especificar una contraseña, aunque si se generará una clave privada para la solicitud (Figura 2.57)

```
#> openssl req -new -nodes -out wifi-cert.pem -config ./openssl.cnf
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'key.pem'
-----
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.
-----
Nombre de la organizacion [WifiHotSpot]:
Departamento o division []:Redes
Correo electronico []:root@wifihotspot.com
Ciudad o distrito [Quito]:
Estado o provincia [Pichincha]:
Codigo del pais (dos letras) [EC]:
Nombre comun (hostname o IP) []:192.168.10.1
```

Figura 2.56 Creación de la llave privada y la solicitud de certificado

En donde la opción –nodes se especifica para indicar que no se usa una contraseña en la llave privada.

[97] CSR: Solicitud de Certificado (*Certificate Signing Request*)

```
#> openssl ca -out certificado-wifi.pem -config ./openssl.cnf -days 3650 \
-infiles wifi-cert.pem
Using configuration from ./openssl.cnf
Enter pass phrase for ./privado/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
organizationName      :PRINTABLE:'WifiHotSpot'
organizationalUnitName:PRINTABLE:'Redes'
localityName          :PRINTABLE:'Quito'
stateOrProvinceName   :PRINTABLE:'Pichincha'
countryName           :PRINTABLE:'EC'
commonName            :PRINTABLE:'192.168.10.1'
Certificate is to be certified until May 26 00:16:10 2021 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
#>
```

Figura 2.57 Creación del certificado autofirmado

La solicitud de firmado se especifica con la opción –infiles. El certificado será válido por 10 años –days.

Se tiene entonces dos elementos ya generados que se necesitarán en el Apache:

- key.pem → La llave privada.
- certificado-wifi.pem → Certificado Autofirmado.

Lo que hay que realizar es copiar los dos archivos en un directorio, de hecho podrían quedarse donde están, es lo de menos, pero por cuestión de orden y organización se van a copiar en /etc/httpd/conf que en la mayoría de distribuciones es el directorio de configuración del Apache (Figura 2.58)

```
#> cp key.pem certificado-wifi.pem /etc/httpd/conf/
```

Figura 2.58 Comando para copiar la llave privada y el certificado autofirmado

Lo siguiente es añadir en el servidor virtual de Apache, esto dentro del archivo de configuración httpd.conf, la ubicación de la llave privada y del certificado autofirmado, tal como se muestra en la Figura 2.59

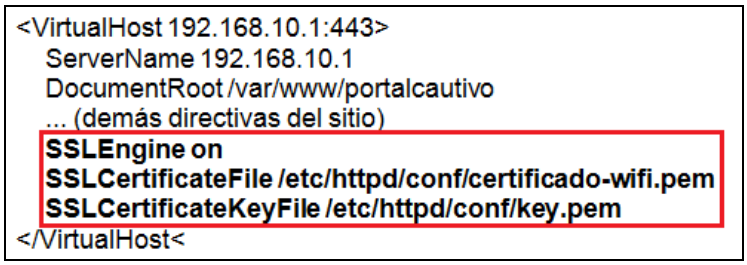


Figura 2.59 Habilitar el servicio SSL en Apache

También debe existir una línea que está fuera de las directivas del servidor virtual, que abre el puerto 443 a la escucha de paquetes:

Listen 443

Listo, ya se tiene habilitado HTTPS en el servidor web Apache.

CAPÍTULO 3: PRUEBAS Y RESULTADOS

3.1 DESCRIPCIÓN DEL ESCENARIO DE PRUEBAS

En la Figura 3.1 se presenta un diagrama de la red que se va a implementar, se muestran las direcciones IP a ser configuradas en el servidor, como el SSID a ser configurado en el Access Point.

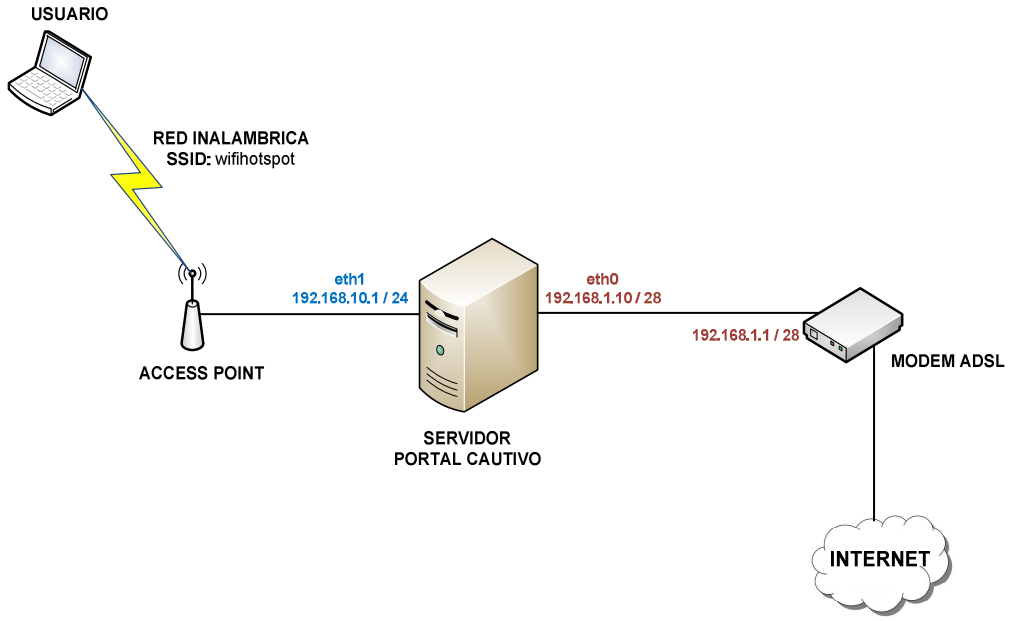


Figura 3.1 Diagrama de Red del Escenario de Pruebas

Para realizar las pruebas de funcionamiento del Portal Cautivo se ha implementado una pequeña red LAN dividida en dos segmentos de red: INTERNET e INALÁMBRICA. El direccionamiento IP de los segmentos de red empleados para el escenario de pruebas se muestra en la Tabla 3.1

Dirección de Red	Máscara	Nombre del segmento de red
192.168.1.0	255.255.255.240	INTERNET
192.168.10.0	255.255.255.0	INALÁMBRICA

Tabla 3.1 Direccionamiento IP de los segmentos de red del Escenario de Pruebas

A continuación se presenta una descripción de los segmentos de red y los elementos que se van incorporando.

3.1.1 SEGMENTO DE RED INTERNET

En este segmento se encuentra configurado el acceso a Internet del Portal Cautivo, el cual se realiza a través de un enlace por MÓDEM ADSL.

El modem que brinda acceso al Internet se encuentra configurado con la dirección 192.168.1.1/28 y el Portal Cautivo tiene configurado en su interfaz eth0 (la que se conecta a este segmento) la dirección 192.168.1.10/28 y como ruta por defecto (gateway) la dirección IP del modem.

3.1.2 SEGMENTO DE RED INALÁMBRICA

En este segmento de red se encuentra el Punto de Acceso Inalámbrico NanoStation2loco de la empresa UBIQUITI NETWORKS la cual diseña, desarrolla y comercializa equipos inalámbricos de banda ancha, ideales para operadores de red, proveedores de acceso inalámbrico a Internet, etc. Los productos de Ubiquiti Networks cumplen con los estándares de banda ancha de la industria, incluidos WiMAX y Wi-Fi.

En la Figura 3.2 se indica el Punto de Acceso Inalámbrico NanoStation2loco de la empresa UBIQUITI NETWORKS.

Figura 3.2 Ubiquiti NanoSation2loco

Como se indica en la Figura 3.3, el Access Point se configura para que trabaje en modo Access Point con un SSID: wifihotspot.

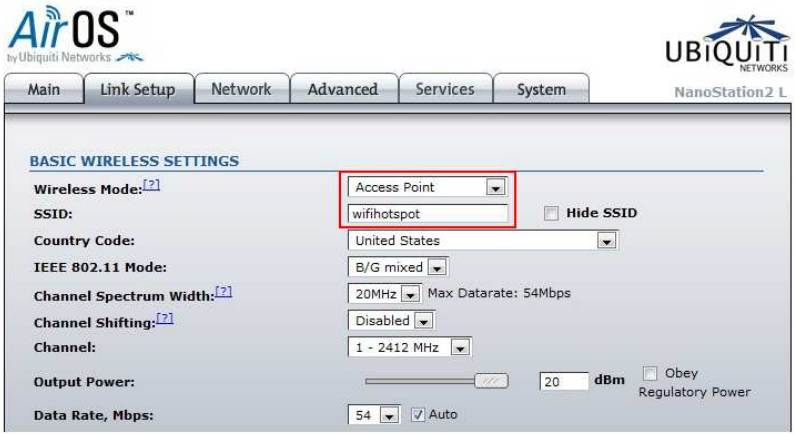


Figura 3.3 Configuración del Enlace Wireless del NanoSation2loco

En lo que se refiere a la configuración de la red, el Access Point está configurado para que trabaje en modo Bridge como se indica en la Figura 3.4, debido a que él no es el encargado de dar las direcciones IP automáticamente (DHCP), esto lo realiza el servidor DHCP configurado en el Portal Cautivo.

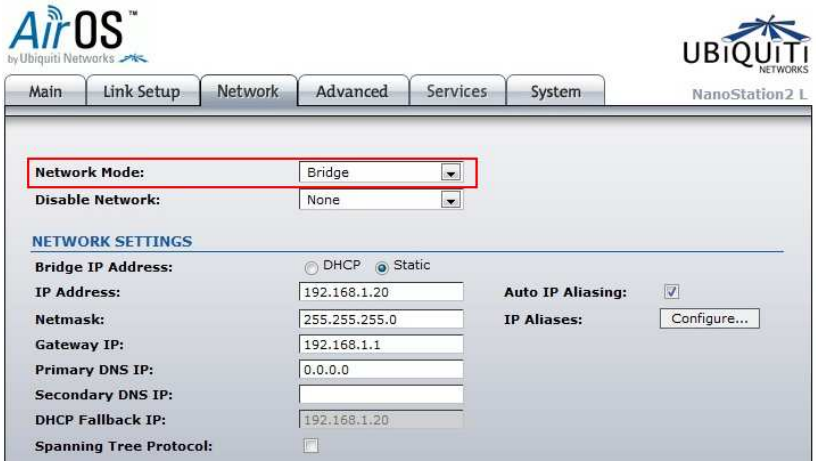


Figura 3.4 Configuración de Red del NanoSation2loco.

El Access Point se conectara a la interfaz eth1 del Portal Cautivo cuya dirección IP es 192.168.10.1/24 que sirve como la puerta de enlace para los usuarios que estén conectados a la red inalámbrica.

En el Portal Cautivo, se tiene instalado y configurado un servidor virtual que contiene la información de las tarjetas de crédito. Para ello es necesario tener instalado la herramienta VMWare[98].

3.2 FUNCIONAMIENTO Y PRUEBAS DEL SISTEMA

En esa sección se verifica la funcionalidad completa del Portal Cautivo, la forma de configuración de los dispositivos inalámbricos para conexión a la red y acceso al servicio, la interacción del usuario con las distintas páginas web que permiten: la autenticación del usuario, la compra y venta de un ticket de Internet.

3.2.1 CONFIGURACIÓN DE LOS DISPOSITIVOS INALÁMBRICOS

Primeramente, el cliente debe asociarse a una red inalámbrica, se localiza la misma, en este caso el SSID de la red es wifihotspot. Se debe verificar primero los niveles de señal para luego asociarse. El proceso de asociación tendrá 4 fases que se detallan a continuación.

FASE 1, se detecta la red a conectarse, verificando un nivel de señal adecuado (Figura 3.5)

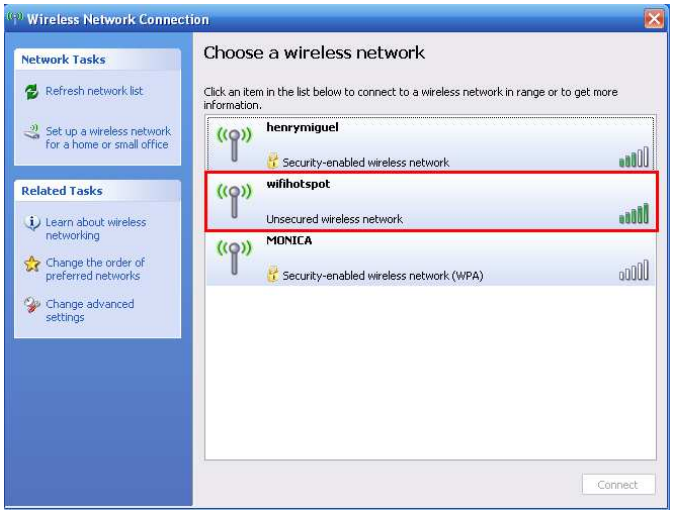


Figura 3.5 Detección de la red

[98] VMware (VM de *Virtual Machine*) es un sistema de virtualización por software. Un sistema virtual por software es un programa que simula un sistema físico (un computador, un hardware) con unas características de hardware determinadas.

FASE 2, se presiona el botón “conectar” y se verifica que se halla asociado al punto de acceso (Figura 3.6)

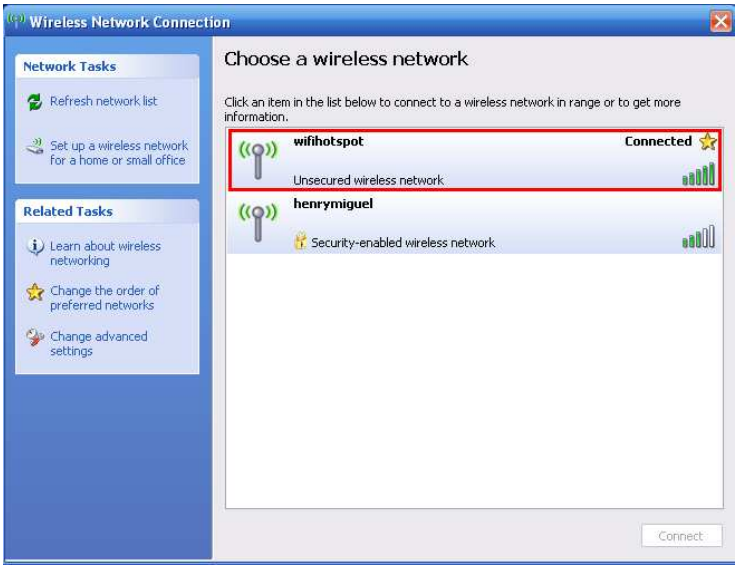


Figura 3.6 Asociación a la red

En esta fase, el servidor DHCP del Portal Cautivo dará una dirección IP libre dentro del rango de direcciones que tiene configurado (192.168.10.5 → 192.168.10.240) con sus respectivos parámetros: dirección IP, máscara, gateway, dirección IP del servidor DNS y dirección IP del servidor DHCP (Figura 3.7)

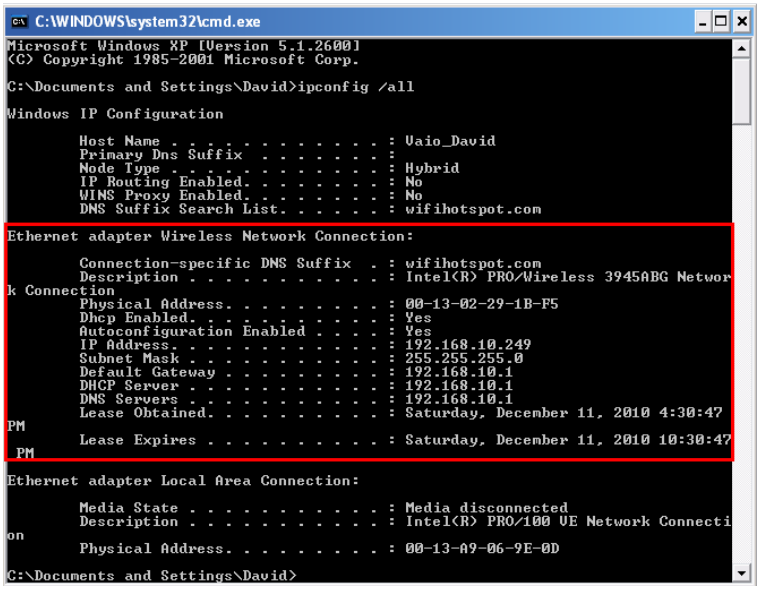


Figura 3.7 Parámetros de Red asignados

Para verificar la lista de direcciones asignadas por el servidor DHCP del Portal Cautivo, verificamos el fichero /var/lib/dhcp/dhcpd.leases. Otra forma es mediante el uso de la herramienta Webmin (Figura 3.8)

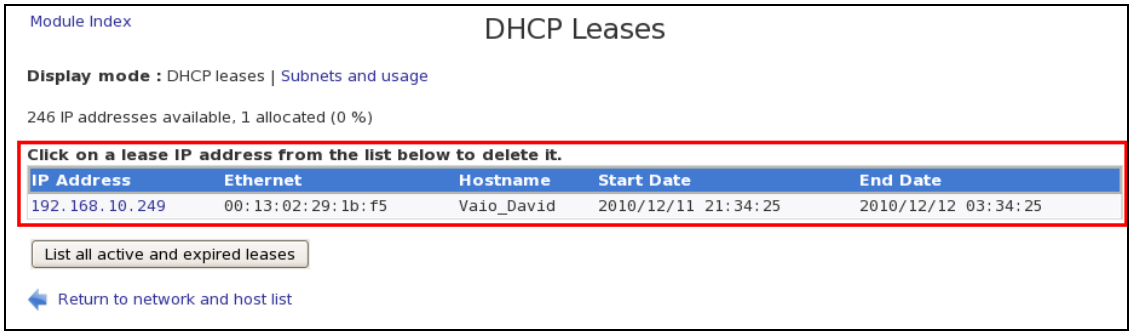


Figura 3.8 Lista de direcciones asignadas por el Servidor DHCP

FASE 3, como se aprecia en la Figura 3.7, el usuario está conectado a la red, pero no puede enviar datos hacia el exterior, hasta que se halla autentificado con el Portal Cautivo (Figura 3.9)

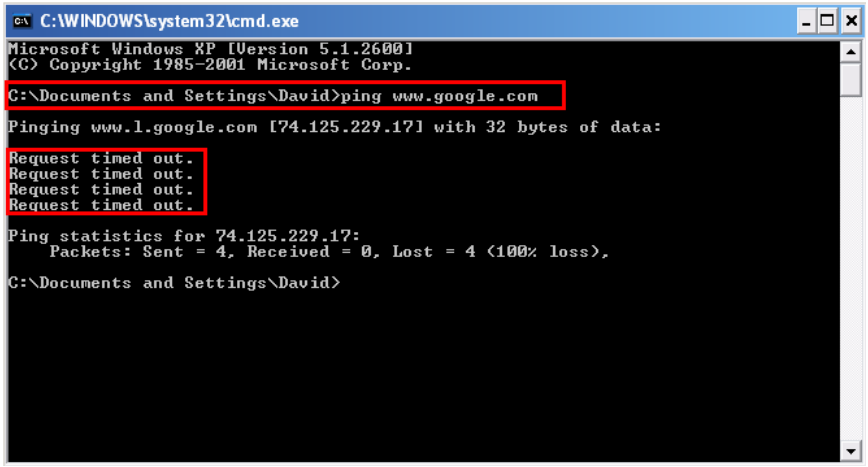


Figura 3.9 Tráfico Bloqueado, hasta que el usuario no se autentifique con el Portal Cautivo

FASE 4, si un usuario que no se ha autentificado con el Portal Cautivo, todo ingreso a una página web será redireccionado a la página de autenticación del Portal Cautivo, en donde debe ingresar el nombre de usuario y password correctos si es que los tiene, caso contrario deberá comprar un ticket de Internet. (Figura 3.10)

Figura 3.10 Página de Autenticación del Portal Cautivo

3.2.2 INTERACCIÓN DEL USUARIO CON LAS PÁGINAS WEB.

Un usuario para poder tener acceso al Internet, primero debe obtener (comprar) un ticket de Internet. Para ello en la página de autenticación del Portal Cautivo (Figura 3.10) se debe ir a la opción de registrarse. Se abren ciertas páginas web en las cuales el usuario deberá ir ingresando ciertos datos según las páginas lo requieran:

- Nombre de usuario y contraseña.
- Tiempo de Internet.
- Datos personales.

A continuación se describen cada una de las páginas web al momento de registrarse:

signUp1.php: En la cual debe ingresar un username y un password válidos (Figura 3.11)

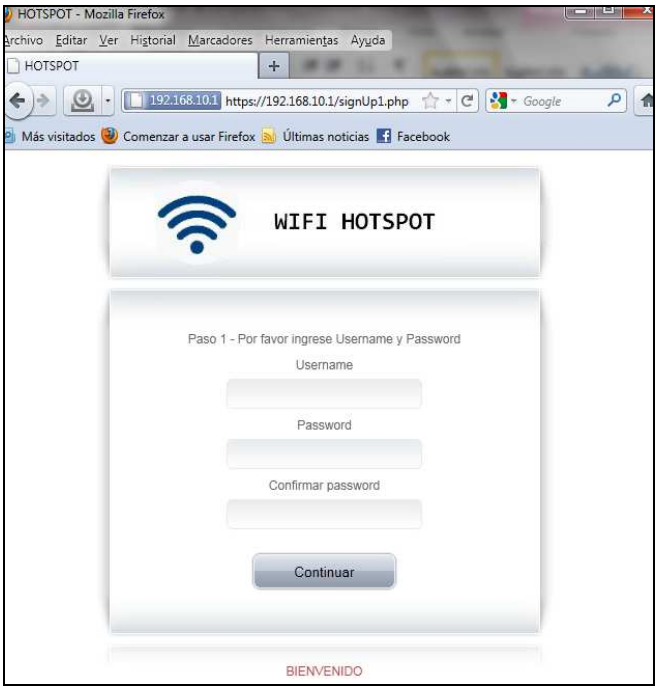


Figura 3.11 signUp1.php

Esta página verifica algunos parámetros antes de continuar, entre ellos:

- Se deben ingresar todos los campos (Figura 3.12)

Figura 3.12 Mensaje de Alerta al no introducir todos los campos

- La longitud del nombre de usuario y el password debe estar entre 4 y 16 caracteres, no se aceptan: espacios en blanco, ñ y tildes (Figura 3.13 y 3.14)

Figura 3.13 Mensaje de Alerta cuando no están entre el número de caracteres definidos

Figura 3.14 Mensaje de Alerta cuando se introduce un carácter erróneo

.

- Que el usuario y password no se encuentren registrados, caso contrario se mostrará el mensaje de la Figura 3.15

Figura 3.15 error.php

signUp2.php: En la cual se debe seleccionar el tiempo de Internet que se va a comprar en horas y minutos. Se muestra además el costo por minuto de Internet (Figura 3.16)

Figura 3.16 signUp2.php

Para poder continuar, se debe seleccionar un valor para los dos campos, caso contrario muestra el siguiente mensaje de alerta (Figura 3.17)

Figura 3.17 Mensaje de Alerta al no seleccionar un valor para los dos campos (horas y minutos)

signUp3.php: En la cual se deben ingresar los datos personales del usuario: nombres, apellidos, cédula de identidad, teléfono, email y los datos de la tarjeta de crédito que se va a usar para el pago del ticket, entre ellos: tipo, número, fecha de expiración y el código CVV (Valor de Validación de la tarjeta de crédito) (Figura 3.18)

Figura 3.18 signUp3.php

La página verifica algunos parámetros antes de continuar, entre ellos:

- Se deben ingresar todos los campos, caso contrario se muestra la página web de la Figura 3.19:

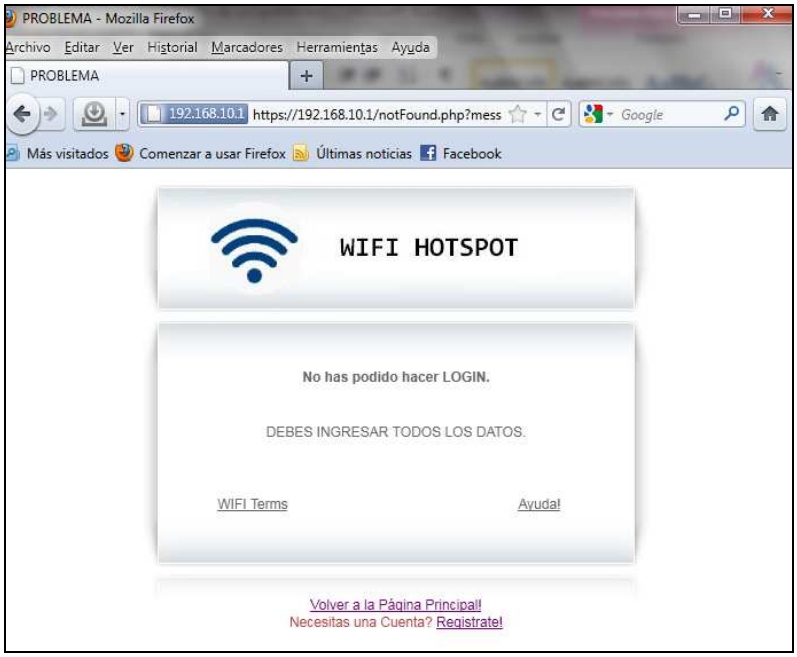


Figura 3.19 notFound.php

- Verifica que el número de cédula de identidad sea correcto (Figura 3.20)

Figura 3.20 notFound.php: Cédula Incorrecta.

- Verifica que el número de teléfono y el email sean correctos (Figura 3.21 y 3.22)

Figura 3.21 Mensaje de Alerta al ingresar un número de teléfono Incorrecto

Figura 3.22 Mensaje de Alerta al ingresar un email Incorrecto

- Por último verifica los parámetros de la tarjeta de crédito. Que el número de la tarjeta corresponda con el tipo de la tarjeta, que se ingresen solo números en los campos de Credit Number Card y CODE, que la fecha de caducidad escogida sea la correcta (Figura 3.23)

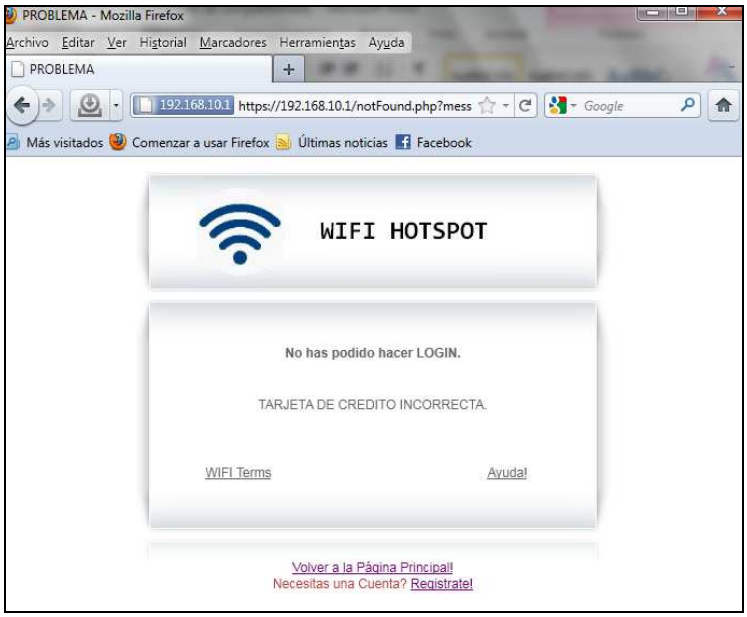


Figura 3.23 notFound.php: Tarjeta de Crédito Incorrecta

check2.php: Indica que el usuario ha ingresado exitosamente, con sus respectivos Detalles de la Cuenta y los Datos de la Tarjeta de Crédito (Figura 3.24)

Figura 3.24 check2.php

Una vez que el usuario se registró exitosamente con el Portal Cautivo, debe ingresar su nombre de usuario y password para poder tener acceso al Internet.

Una vez ingresados dichos datos en la página de autenticación del Portal Cautivo, se abrirá una página infoBox.php**,** en la cual se muestra un reloj que va decreciendo su valor conforme pasa el tiempo.

Además del reloj, se muestra información referente al dispositivo inalámbrico que esté usando el cliente: su dirección IP, su dirección MAC y el username (Figura 3.25)

Figura 3.25 infoBox.php

En la Figura 3.26 se muestran las reglas *iptables* que se crean en el Portal Cautivo para permitir acceso al usuario que se ha autenticado de forma exitosa en el sistema.

```
[root@server ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  192.168.10.0/24      0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  192.168.10.241       0.0.0.0/0
ACCEPT     all  --  192.168.10.2         0.0.0.0/0
DROP       all  --  192.168.10.0/24      0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

[root@server ~]# iptables -t nat -L -n
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
USUARIOPERMITIDO  all  --  0.0.0.0/0            0.0.0.0/0
DNAT       tcp  --  192.168.10.0/24      0.0.0.0/0           tcp dpt:80 to:192.168.10.1
DNAT       tcp  --  192.168.10.0/24      0.0.0.0/0           tcp dpt:443 to:192.168.10.1

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  192.168.10.0/24      0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain USUARIOPERMITIDO (1 references)
target     prot opt source               destination
ACCEPT     all  --  192.168.10.241       0.0.0.0/0
ACCEPT     all  --  192.168.10.2         0.0.0.0/0
RETURN     all  --  0.0.0.0/0            0.0.0.0/0
```

Figura 3.26 Reglas de iptables generadas para un usuario

Adicionalmente se pudo verificar que el usuario autenticado exitosamente, tenga acceso a Internet, lo cual se muestra en la Figura 3.27

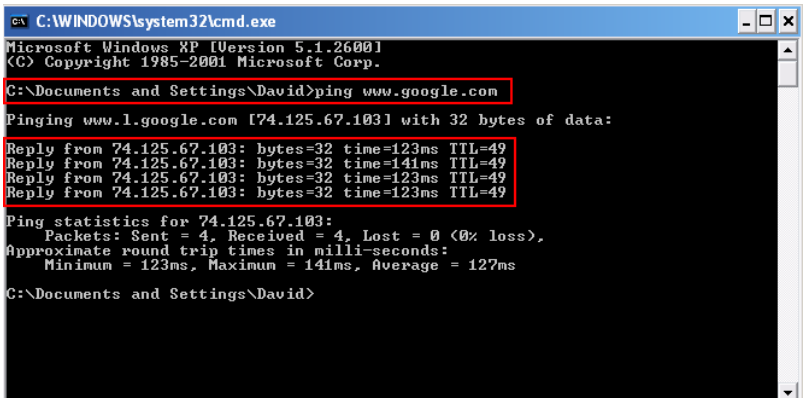


Figura 3.27 Acceso a Internet satisfactorio

Un usuario puede agregar más tiempo de Internet, en el cual debe seleccionar primero el tiempo de Internet (Figura 3.28), luego debe ingresar los datos de la tarjeta de crédito (Figura 3.29) y por último se muestra que el tiempo ha sido ingresado exitosamente (Figura 3.30)

Figura 3.28 signUpAdd1.php

Figura 3.29 signUpAdd2.php

Figura 3.30 checkAdd2.php

En la página de autenticación del Portal Cautivo, se muestran tres enlaces adicionales, que se detallan a continuación:

- Wifi-Terms: Se detallan los términos del servicio, que se deben cumplir entre el usuario y el Portal Cautivo (Figura 3.31)

Figura 3.31 terms.php

- Ayuda!: Se detallan los pasos que un usuario debe seguir para autenticarse con el Portal Cautivo (Figura 3.32)

Figura 3.32 help.php

- Cambiar Password!: Permite cambiar el password de un usuario en concreto. Se debe ingresar el username, el password y el nuevo password. (Figura 3.33)

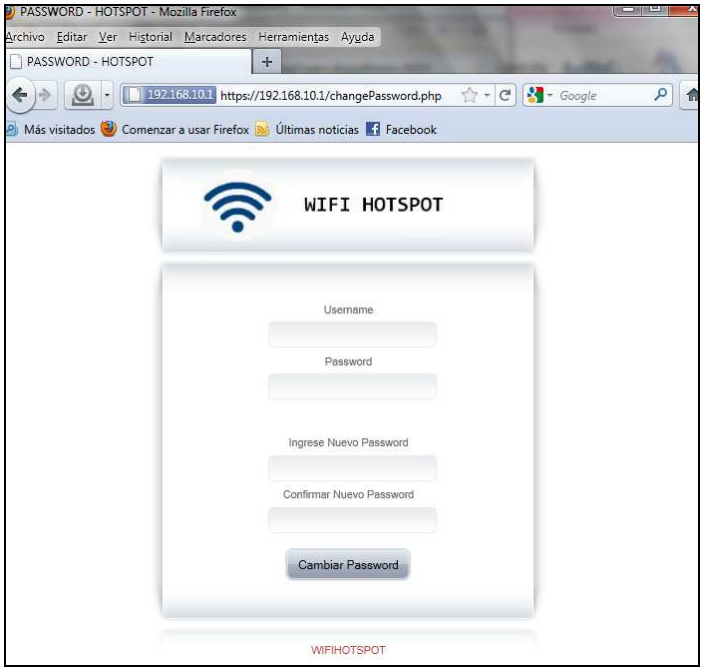


Figura 3.33 changePassword.php

3.2.3 INTERACCIÓN DEL VENDEDOR CON LAS PÁGINAS WEB

También existe la posibilidad de que un personal pueda vender los tickets de Internet. Para ello la persona que va a vender dichos tickets, ingresa su username y password asignados (por defecto el username = vendedor y el password = vendedor) en la página de autenticación del Portal Cautivo, el cual le llevará a la página inicio.php, la cual permite:

- Buscar Usuarios para obtener un historial de ingreso de cierto usuario.
- Generar y vender Tickets de Internet.
- Cambiar el diseño de las páginas que se muestran a los usuarios.
- Cambiar el precio por minuto del Internet. Se pueden añadir impuestos al precio si se desea.

En la Figura 3.34 se muestra la página web de bienvenida del vendedor (inicio.php).

Figura 3.34 inicio.php

A continuación se describen cada una de las páginas web que utiliza el vendedor:

user.php: Permite buscar un usuario en la base de datos del Portal Cautivo. (Figura 3.35)

Figura 3.35 user.php

Al momento de presionar en Buscar, se abre la página userRes.php, la cual muestra la información del usuario: username, dirección IP, su dirección MAC, duración del ticket, el precio, la fecha y hora en que ha ingresado a consumir el Internet (Figura 3.36)

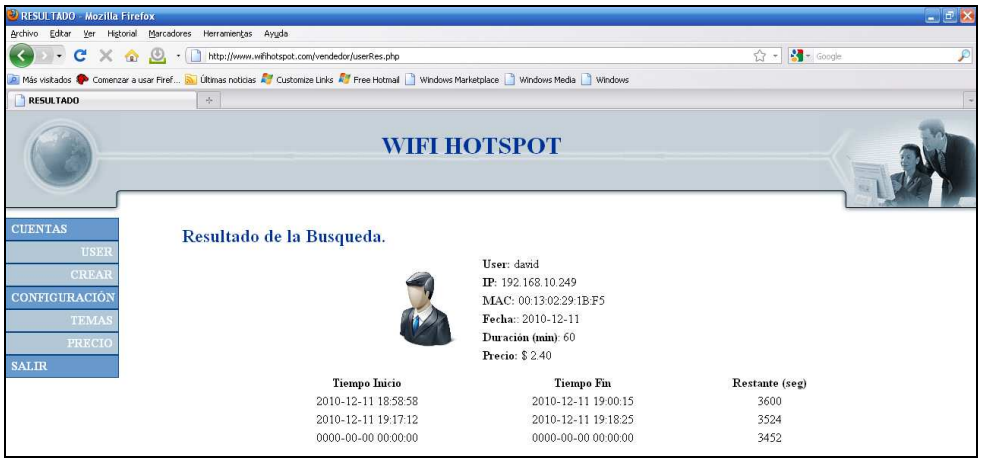


Figura 3.36 userRes.php

Si se ingresa un usuario que no existe en la base de datos del Portal Cautivo, muestra un mensaje en la página web de USUARIO NO ENCONTRADO.

crear.php: Permite crear y vender tickets de Internet. Para ello primero se ingresa el número de cédula del cliente para verificar si existe en la base de datos (Figura 3.37)

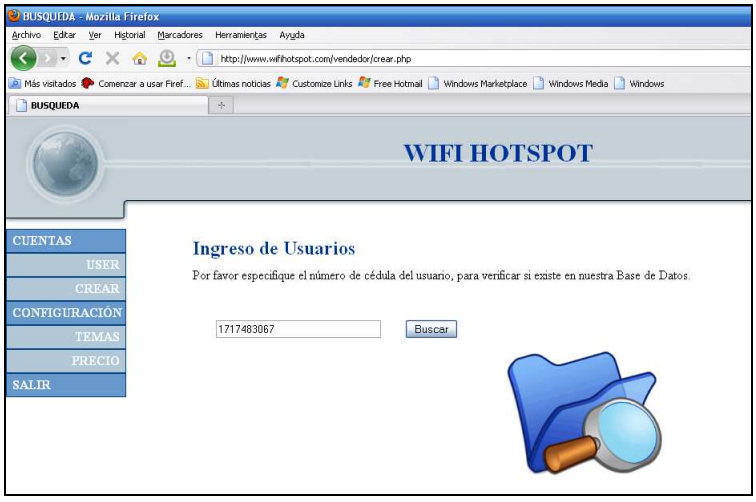


Figura 3.37 crear.php

- Si existe en la base, se abre la página inputTicket.php la cual muestra un Formulario de Ingreso con los datos personales del Cliente, en el cual debemos seleccionar el consumo de Internet (en horas y minutos) y la forma de pago (efectivo o tarjeta de crédito) (Figura 3.38)

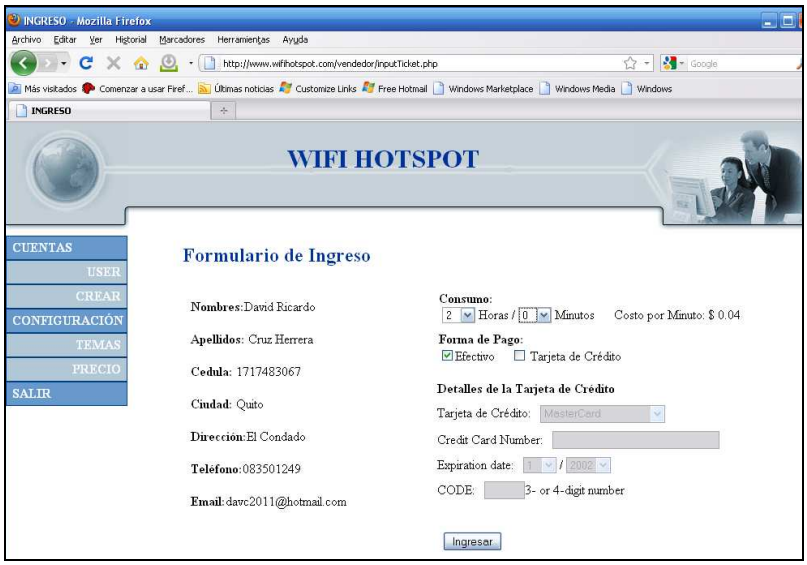


Figura 3.38 inputTicket.php

Al dar clic en ingresar se abre la página checkTicket.php, nos informa que se ha ingresado un usuario satisfactoriamente. El username y password son generados automáticamente (Figura 3.39)

Figura 3.39 checkTicket.php

Adicionalmente da una opción para imprimir el ticket en formato pdf (Figura 3.40)

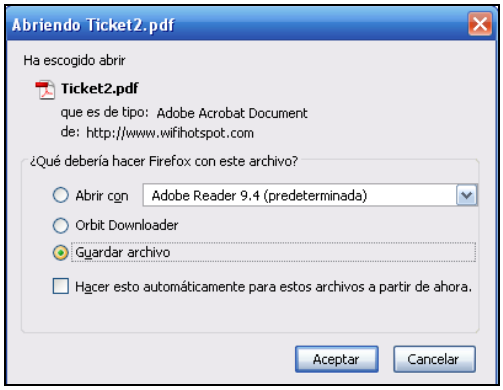


Figura 3.40 Impresión del Ticket en formato pdf

En la Figura 3.41 se indica el formato de impresión del ticket.

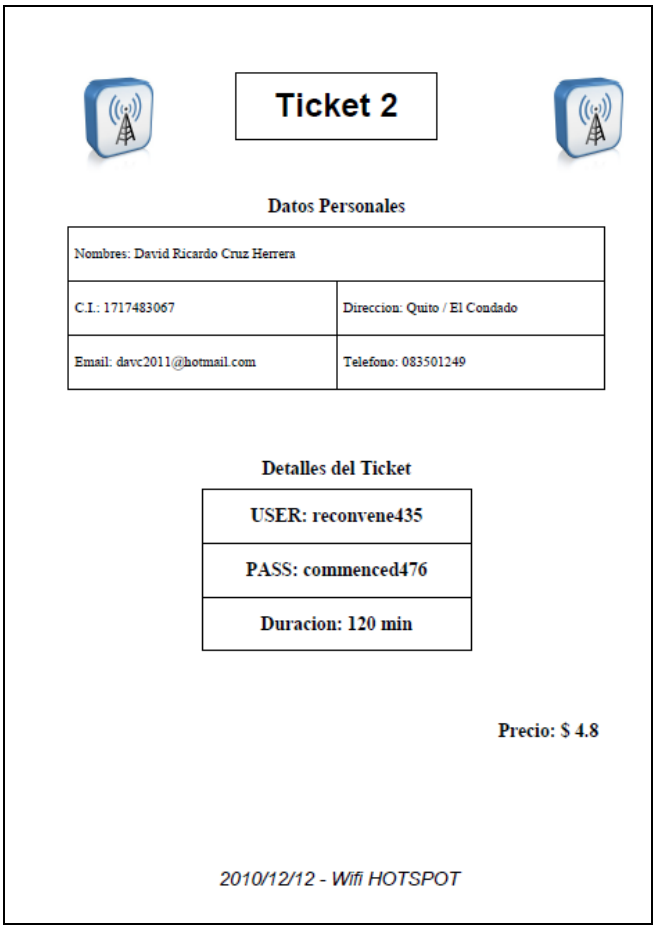


Figura 3.41 Formato del ticket.

- Si el usuario no existe en la base, se abre la página inputUser.php en la cual se debe completar todo el Formulario de Ingreso que consta de: datos personales del usuario, consumo de Internet (en horas y minutos) y la forma de pago (efectivo o tarjeta de crédito) (Figura 3.42)

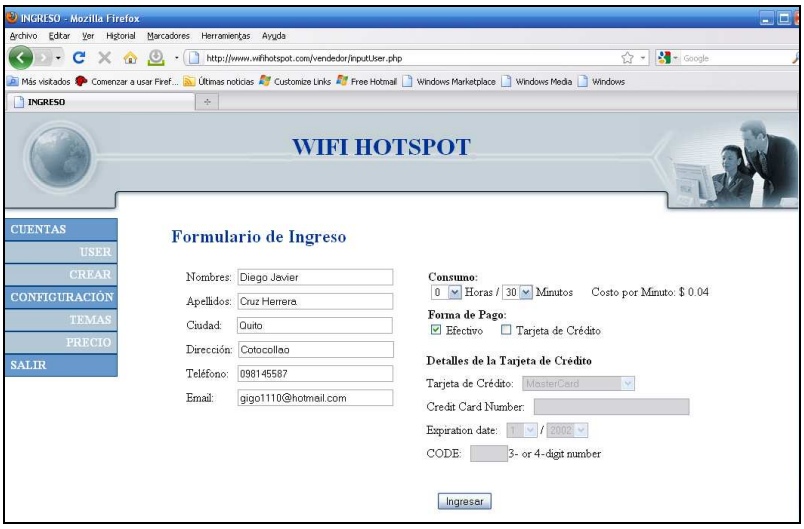


Figura 3.42 inputUser.php

Al dar clic en ingresar se abre a la página checkUser.php, nos informa que se ha ingresado un usuario satisfactoriamente. El username y password son generados automáticamente (Figura 3.43)

Figura 3.43 checkUser.php

Adicionalmente da una opción para imprimir el ticket en formato pdf (Figura 3.44)

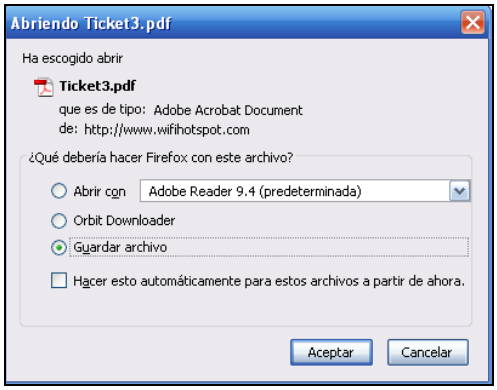


Figura 3.44 Impresión del Ticket en formato pdf

En la Figura 3.45 se indica el formato de impresión del ticket.

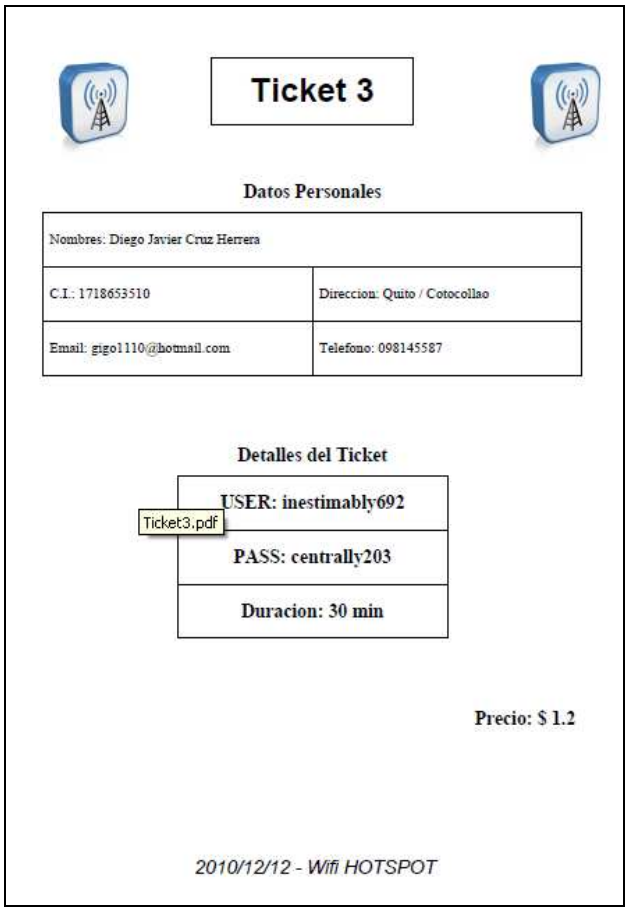


Figura 3.45 Formato del ticket.

themes.php: Permite cambiar los temas que se muestran en las páginas del Portal Cautivo. Si se desea crear propios temas, se debe modificar el archivo *style.css* localizado en la carpeta */www/portalcautivo/Themes/Default* (Figura 3.46)

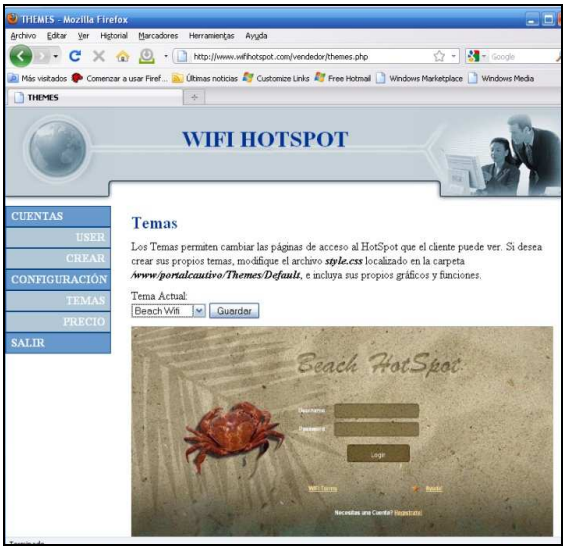


Figura 3.46 themes.php

precio.php: Permite modificar el precio por minuto que tendrán los tickets de Internet. Adicionalmente se puede definir impuestos sobre el precio por minuto si es que lo requiere (Figura 3.47)

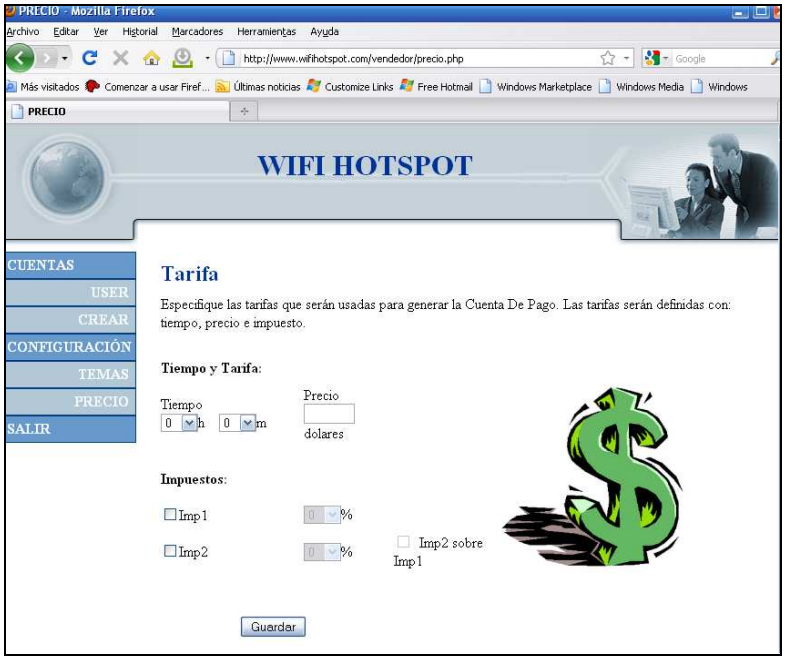


Figura 3.47 precio.php

salir.php: Permite cerrar la sesión del vendedor (Figura 3.48)

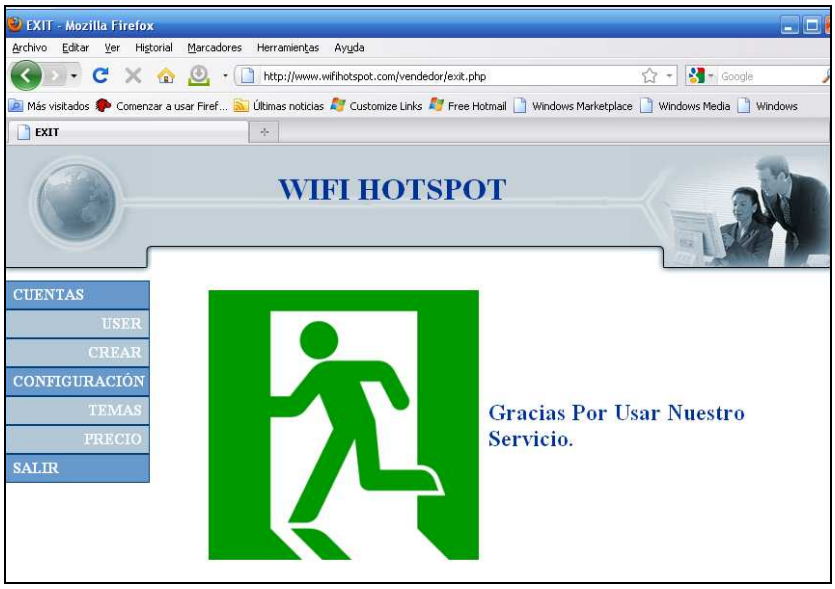


Figura 3.48 salir.php

3.3 VALIDACIÓN DE LA TARJETA DE CREDITO

El Portal Cautivo permite el ingreso de tarjetas de crédito para la venta de tickets de Internet, tanto en las páginas que se muestran a los clientes como en la página web del vendedor.

El Portal Cautivo solamente verifica que el número de la tarjeta de crédito sea el correcto para el tipo de tarjeta escogido. Adicionalmente verifica que la fecha de expiración de la tarjeta de crédito sea la correcta. Para ello se implementa una clase elaborada en PHP denominada *class.creditcard.php***,** va ser la encargada de verificar si la tarjeta de crédito es correcta.

La validación de la tarjeta de crédito se la realiza mediante el uso de un algoritmo denominado Algoritmo Mod 10. Además del algoritmo, las tarjetas de crédito tienen reglas en cuanto al formato del número. A continuación se presenta dichas reglas para las 6 más populares tarjetas de crédito:

- **Mastercard:** 16 dígitos de longitud y puede comenzar con los números del 51 al 55.
- **Visa:** 13 o 16 dígitos de longitud y debe comenzar con el número 4.
- **American Express:** 15 dígitos de longitud y debe comenzar con los números 34 o 37.
- **Diners Club:** 14 dígitos de longitud y debe comenzar con los números del 300 al 305, 36 o 38.
- **Discover:** 16 dígitos de longitud y debe comenzar con el número 6011.
- **JCB:** 15 o 16 dígitos de longitud y debe comenzar con los números 3, 1800 o 2131.

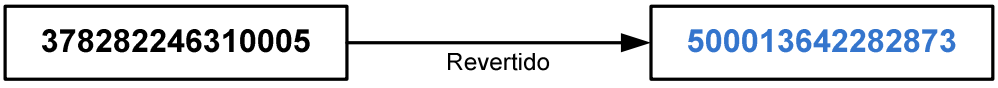
En la Figura 3.49 se indican los diferentes logotipos de las distintas tarjetas de crédito.

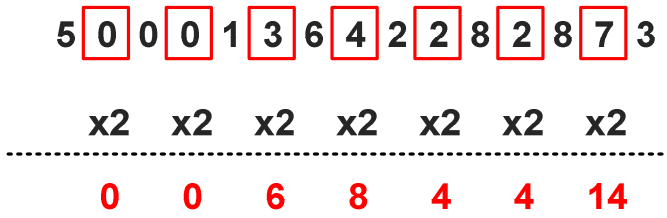
Figura 3.49 Logotipos de Tarjetas de Crédito

3.3.1 ALGORITMO MOD 10

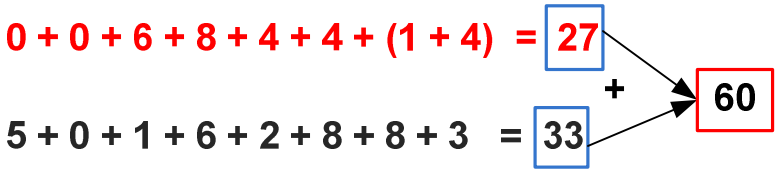
Son tres pasos que el Algoritmo Mod 10 utiliza para verificar si una tarjeta de crédito es válida o no. Usaremos el siguiente número de tarjeta de crédito válido: 378282246310005 para demostrar el uso del algoritmo.

Paso 1: El número es revertido, y el valor de cada segundo dígito es duplicado.





Paso 2: Los valores de los dígitos que resultaron de multiplicar por 2 son sumados con los valores de los dígitos restantes (los que no se duplicaron).



Paso 3: Al resultado obtenido se le aplica la operación módulo 10, si el resultado es 0 → el número de la tarjeta de crédito es válido, caso contrario es incorrecto.

60 MOD 10 = 0

Como se puede observar el resultado da 0, concluyendo que el número de tarjeta de crédito es correcto.

3.3.2 IMPLEMENTACIÓN DEL ALGORITMO MOD 10

La implementación del algoritmo MOD 10 se lleva a cabo con la clase elaborada en PHP denominada class.creditcard.php**,** la cual debe verificar ciertos parámetros para poder así validar la tarjeta de crédito, entre ellos:

- Que solo ingresen números en los campos de Credit Card Number y CODE (código CVV (Valor de Validación de la Tarjeta de Crédito)).
- Que el formato del número de la tarjeta de crédito corresponde con el tipo de la tarjeta, de acuerdo a las reglas mencionadas anteriormente.
- Que la fecha de caducidad escogida sea la correcta.

En las páginas web: signUp3.php (Figura 3.50), inputTicket.php (Figura 3.51) e inputUser.php (Figura 3.52) se debe incluir la clase class.creditcard.php, debido a que ellas contienen el detalle de la tarjeta de crédito

Figura 3.50 Ventana de signUp3.php

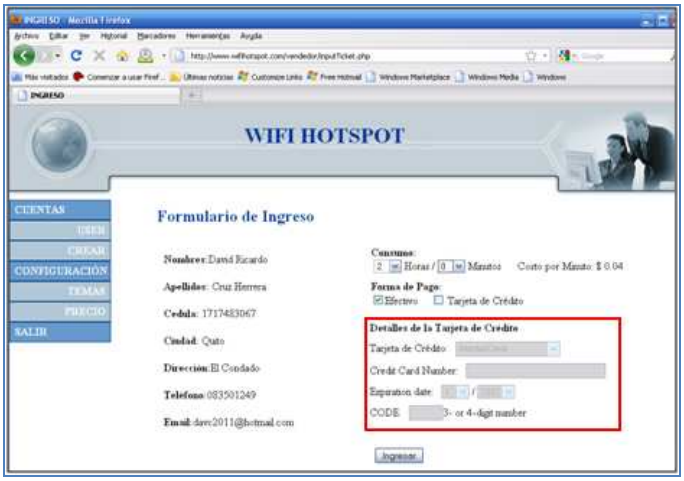


Figura 3.51 Ventana de inputTicket.php

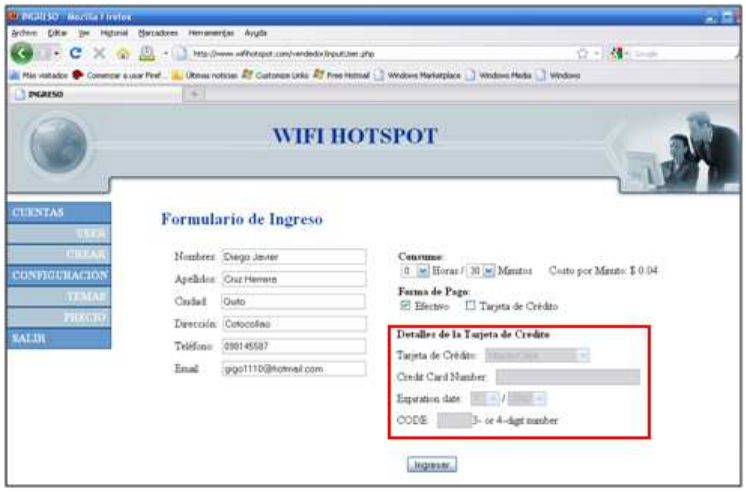


Figura 3.52 Ventana de inputUser.php

Si la tarjeta de crédito no es válida, se despliega un mensaje que se muestra en la Figura 3.53

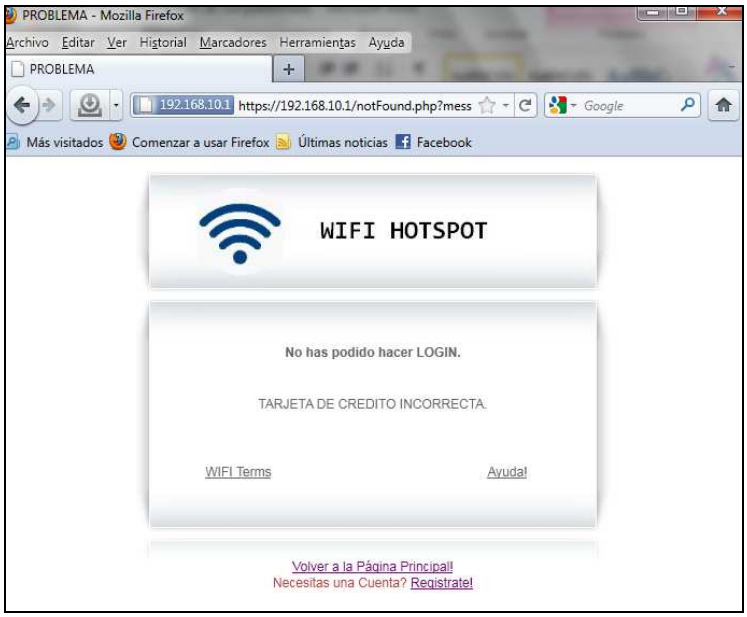


Figura 3.53 Tarjeta de Crédito Incorrecta.

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Un Portal Cautivo no necesita una configuración del lado del cliente, no requiere instalarse ningún programa en el PC del cliente; el Portal Cautivo asegura el enrutamiento de todos los clientes a la pantalla de inicio de sesión.

- El software de código abierto hoy en día tiene una gran aceptación y está ganando cada vez más espacio como una alternativa para quien no tiene la posibilidad de pagar altos costos por un software propietario.

- Linux es popular entre programadores y desarrolladores e implica un espíritu de colaboración, de esta forma si un usuario llegase a encontrar algún problema en cualquier aplicación, podrá editar directamente el código fuente o informar al desarrollador de la aplicación, para que en una futura versión se corrija el problema.

- CentOS en los últimos años ha sido el sistema operativo Linux dominante, sustituyendo rápidamente a la empresa Red Hat Enterprise como sistema operativo Linux estándar, debido a sus características, precio y seguridad. Los administradores de servidores Linux están más familiarizados con la distribución CentOS que cualquier otra distribución.

- Webmin es una aplicación de administración con interfaz web para Linux que permite efectuar tareas básicas de administración como la gestión de usuarios y tareas más avanzadas como la gestión de servidores web, bases de datos, servidores de correos.

- PHP es un lenguaje de programación del lado del servidor que puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Posee un gran

parecido con los lenguajes más comunes de programación como C y Perl, lo cual permite a la mayoría de programadores crear aplicaciones complejas con un nivel de aprendizaje muy simple.

- En las versiones actuales de PHP (versión 5 o superior) la variable $HTTP_POST_VARS es obsoleta. Para evitar este problema se plantean dos soluciones:

    - Se reemplaza la variable $HTTP_POST_VARS por $_POST.
    - Se edita el archivo de configuración (php.ini), colocando el valor de On en: regiter_long_arrays=On, que por defecto lo trae Off.

- La implementación presentada es un claro ejemplo de la gran cantidad de posibilidades que se tiene para implementar soluciones de seguridad en redes de comunicaciones con herramientas de software libre, como por ejemplo el uso de iptables para la implementación del Firewall, Apache y el módulo SSL para la implementación del servidor Web Seguro, MySQL para la implementación del servidor de Base de Datos, entre otros, y no únicamente para el sistema operativo Linux ya que existen versiones de software libre que también pueden ser empleadas en otros sistemas operativos.

4.2 RECOMENDACIONES

- El lugar de instalación del computador que va contener al Portal Cautivo debe poseer una apropiada instalación eléctrica, así como también debe poseer elementos de respaldo como sistemas de alimentación ininterrumpida (UPS: *Uninterrupted Power System*).

- A la hora de seleccionar el Access Point, es necesario que éste tenga la posibilidad de trabajar en modo Bridge (Puente) debido a que él no es el encargado de dar direcciones IP automáticamente, esto lo realiza el servidor DHCP configurado en el Portal cautivo.

- Es necesario activar JavaScript en los navegadores, ya que algunos lo traen desactivado por defecto, para el correcto funcionamiento de las páginas web del Portal Cautivo, especialmente la página web que contiene el tiempo de uso de Internet (representado por un reloj que va decreciendo).

- Lo primero que se debe configurar en el servidor de base de datos MySQL (previo a la instalación) es asignar una contraseña al usuario root (administrador) debido a que por defecto no lo trae.

- Se debe activar el servidor DHCP en la interfaz de red adecuada, debido a que si se activa en las dos interfaces puede causar una denegación del servicio.

BIBLIOGRAFÍA

LIBROS Y TÉSIS

- CORTEZ Felipe, Implementación de un Nodo con Portal Cautivo (Captive Portal) través de un Linksys WRT54g y m0n0wall. Segunda Edición. 2006.

- ANDREU Fernando, redes WLAN. Fundamentos y aplicaciones de seguridad. Primera Edición. Marcobombo, S.A. 2006.

- HUIDROBO José Manuel; ROLDÁN David, Comunicaciones en redes WLAN. Primera Edición. 2005.

- ASUNCIÓN Santamaría; HERNÁNDEZ Francisco J. López, Wireless LAN Standars And Applications. Artech House. 2001.

- JACOBSON Ivar; BOOCH Grady; RUMBAUGH James, El Proceso Unificado de Desarrollo de Software. Addison Wesley. 2000.

- ORTIZ Sampedro Carlos Eduardo, “DISEÑO E IMPLEMENTACION DE UNA APLICACIÓN PARA TELÉFONOS CELULARES QUE PERMITA LA CONFIGURACIÓN DE EQUIPOS CISCO UTILIZANDO UN ENLACE BLUETOOTH”. Tesis EPN. 2010. Ingeniería en Electrónica y Redes.

- INSUASTI Proaño Jorge Isaac, “DISEÑO E IMPLEMENTACIÓN DE DOS SOLUCIONES DE SEGURIDAD PARA UNA RED LAN INALÁMBRICA”. Tesis EPN. 2004. Ingeniería en Electrónica y Redes.

- SÁNCHEZ Robayo Jorge Omar, “DISEÑO E IMPLEMENTACIÓN DE UN PROGRAMA FILTRADOR DE PAQUETES PARA EL CONTROL DE ACCESO A INTERNET”. Tesis EPN. 2002. Ingeniería en Electrónica y Telecomunicaciones.

- PILALUISA Quinatoa Santiago Rogelio; REVELO Portilla Segundo Lizardo, “SISTEMA DE SOFTWARE PARAMETRIZABLE PARA LA ADMINISTRACIÓN DE SEGURIDADES DE ACCESO CONTROLADOS POR PERFILES DE USUARIOS PARA PROGRAMAS DE APLICACIÓN”. Tesis EPN. 2001. Ingeniería en Sistemas.

DIRECCIONES ELECTRÓNICAS

- http://www.antamediahotspot.com/ - HOTSPOT BILLING SOFTWARE.
- http://www.pfsense.org/ - PFSENSE
- https://www.hotspotsystem.com/es/main/index.html - Inicia Servicio de HOTSPOT.
- http://dev.wifidog.org/ - A Captive Portal Suite.
- http://www.chillispot.info/ - Chillispot - Open Source Wireless LAN Access Point Controller
- http://www.4ipnet.com/en/products_overview.php?ps=hotspot – Hotspot Gateway.
- http://www.netfilter.org/ - The netfilter.org project.
- http://php.netindex.php – PHP: HyperText Processor.
- http://www.mysql.com/ - MySQL: The world’s most popular open source database.
- http://www.apache.org – Welcome – The Apache Software Foundation.