UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO

"Cuna de héroes, crisol de pensadores"

FACULTAD DE INGENIERÍA ELÉCTRICA

Tesis:

DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA PROVEEDOR DE INTERNET WI-FI EN MODALIDAD DE PREPAGO CON VENTA AUTOMATIZADA

Presentado por:
Julio César Villar Mendoza

Para obtener el título de:
Ingeniero en Electrónica

Asesor:
M.C. Félix Jiménez Pérez

Morelia; Michoacán, noviembre del 2019.

Dedicatoria

- Al señor Jesucristo por haberme dado la fuerza para alcanzar este proyecto de vida.

- A mis padres que me han apoyado todo el tiempo de mi carrera, aunque ausentes físicamente pero presentes siempre conmigo.

Agradecimientos

- A la Deidad Celestial que siempre me ha acompañado en el camino de mi vida.

- A mis padres por todo su apoyo que me han dado y por creer en mí.

- A la Universidad Michoacana de San Nicolás de Hidalgo por abrirme sus puertas para que lograra mi sueño de ser un ingeniero en electrónica.

- A todos los maestros de la Facultad de Ingeniería Eléctrica quienes se preocupan y preparan por nuestra formación como profesionistas.

- A aquellos maestros que me han sabido escuchar y orientar: Dr Gilberto González Ávalos, Dr Roberto Tapia Sánchez.

- Finalmente al M. C. Félix Jiménez Pérez por su incomparable apoyo brindado para poder terminar la carrera.

Contenido

Dedicatoria I
Agradecimientos II
Contenido III
Lista de imágenes VI
Lista de tablas VIII
Lista de apéndices y anexos IX
Lista de abreviaturas y terminologías X
Resumen XII
Abstract XII
Palabras Clave XII
Capítulo 1. Introducción 1
1.1 Antecedentes 1
1.1.1 Surgimiento del Internet 1
1.1.2 Formas de difusión del Internet 2
1.1.3 Empresas nacionales que difunden el Internet vía Wi-Fi 3
1.2 Justificación 5
1.3 Planteamiento del problema 5
1.4 Objetivos 6
1.4.1 General 6
1.4.2 Particulares 7
1.5 Metodología de solución 7
1.6 Descripción de los capítulos 7
Capítulo 2. La estación PSII 8
2.1 Concepto y definición general 8
2.2 Estructura general de una estación PSII 8
2.3 Elementos que conforman una estación PSII 9
2.4 Seguridad de la estación PSII 10
2.4.1 Tipos de autenticación 10
2.5 El servidor NAS 11
2.6 El servidor RADIUS 12
2.7 El certificado de seguridad SSL 12
2.7.1 El protocolo SSL 12
2.7.2 El certificado de seguridad 12

2.8 El servidor WEB 13
2.9 La administración de las credenciales de usuario 13
2.10 El entorno de trabajo 14
Capítulo 3. Diseño y Construcción de la Estación PSII 15
3.1 Características de la estación 15
3.2 Estrategias de solución 15
3.3 Modelado general de la estación PSII 16
3.4 Construcción del módulo de acceso inalámbrico 16
3.4.1 La cubierta exterior o protección 16
3.4.1.1 Diseño del sistema de ventilación 17
3.4.1.2 Diseño del panel de control exterior 19
3.4.1.3 La tarjeta electrónica de encendido o controlador de arranque 21
3.4.1.4 Montaje y conexión de las tarjetas 28
3.4.2 Instalación y configuración del software 30
3.4.2.1 El sistema operativo 30
3.4.2.2 La base de datos de MySQL 31
3.4.2.3 Los planes de renta y su estructura en la base de datos 32
3.4.2.4 Las claves de usuario 36
3.4.2.5 La generación de claves de usuario 37
3.4.2.6 Activación de una clave de usuario en mysql 39
3.4.2.7 La activación automatizada de las claves de usuario 40
3.4.2.8 Activación remota de claves 42
3.4.2.9 Generación del certificado de seguridad SSL 43
3.4.2.10 Configuraciones de freeradius 46
3.4.2.11 Configuraciones del servidor web apache 46
3.4.2.12 Configuraciones del portal captivo 46
3.4.2.13 Configuraciones del controlador de red o Chillispot 48
3.5 Construcción del módulo de auto-servicio 48
3.5.1 Construcción del hardware del prototipo 49
3.5.1.1 Raspberry Pi Zero W 49
3.5.1.2 Pantalla LCD de 84x48 píxeles. 50
3.5.1.3 Botón de presión o push boton 50
3.5.1.4 Leds indicadores 51
3.5.1.5 Impresora térmica 52

3.5.1.6 Selector de monedas o coin selector 53
3.5.1.7 Diagrama electrónico de interconexiones del prototipo 54
3.5.2 Del software 56
3.5.2.1 Configuración del programa de venta 56
Capítulo 4. Pruebas y Resultados 60
4.1 Del módulo de acceso inalámbrico 60
4.1.1 El hardware 60
4.1.1.1 Del sistema de ventilación 60
4.1.1.2 De la exposición al flujo pluvial o lluvia 61
4.1.2 El software 61
4.1.2.1 Freeradius 61
4.1.2.2 Radtest 62
4.1.2.3 El controlador de red o chillispot 64
4.1.2.4 El servidor web apache 64
4.1.2.5 El portal captivo 67
4.1.2.6 Pruebas de ancho de banda 68
4.2 Resultados del prototipo del módulo de auto-servicio 69
4.2.1 Hardware 69
4.2.1.1 Prueba de la compra de un plan de renta 69
Capítulo 5. Conclusiones 74
5.1 Ventajas 74
5.2 Mejoras 75
5.3 Trabajos a futuro 76
5.4 Costos aproximados de los materiales y equipo 77
Apéndices y anexos 78
Referencias 163

Lista de imágenes

Fig. 1-1. Cables de transmisión a) coaxial, b) cable UTP y c) fibra óptica 3
Fig. 1-2. Dispositivos de transmisión inalámbrica de Internet..... 3
Fig. 1-3. Cobertura Wi-Fi de Wicoin en México 4
Fig. 1-4. Mapa de cobertura de Megacable en el país 4
Fig. 2-1. Esquema ilustrativo de una estación PSII..... 8
Fig. 2-2. Esquema general de la estación PSII 9
Fig. 2-3. Topología general para un servidor NAS..... 11
Fig. 2-4. Topología de red típica para chillispot 12
Fig. 3-1. Diagrama de bloques de la estación PSII..... 16
Fig. 3-2. Gabinete metálico o cubierta exterior 17
Fig. 3-3. Ranuras de entrada de aire exterior 18
Fig. 3-4. Salida del aire caliente..... 18
Fig. 3-5. Circuito de circulación del aire al interior 19
Fig. 3-6. Panel de control 20
Fig. 3-7. Cubierta exterior o gabinete acondicionado..... 20
Fig. 3-8. a) Diagrama electrónico b) Curva típica de carga..... 22
Fig. 3-9. Configuración como comparador del amplificador operacional..... 22
Fig. 3-10. Circuito RC 23
Fig. 3-11. Resultado de la simulación del circuito RC..... 23
Fig. 3-12. Diagrama electrónico de un divisor de tensión..... 24
Fig. 3-13. Divisor de tensión con valores definidos..... 25
Fig. 3-14. Circuito retardador de encendido..... 25
Fig. 3-15. Simulación de 15 segundos del circuito temporizador 26
Fig. 3-16. Diagrama electrónico completo del controlador de encendido 27
Fig. 3-17. Implementación en placa impresa del circuito controlador 28
Fig. 3-18. CPU, tarjetas de red y fuente de alimentación 29
Fig. 3-19. Chasis atornillado en el gabinete 29
Fig. 3-20. Tarjeta Wi-Fi y controlador de arranque montados. 30
Fig. 3-21. Instalación de Ubuntu10.04 server, lamp y openssh server en el CPU..... 31
Fig. 3-22. Tablas de la base de datos..... 32
Fig. 3-23. Tablas de los planes de renta configuradas 36
Fig. 3-24. Generación de un código alfanumérico con apg..... 38
Fig. 3-25. Clave “Jdnet0cs1” registrada en la base de datos..... 40
Fig. 3-26. Proceso automatizado para activar una clave 41
Fig. 3-27. Resultado del script activador de claves 42
Fig. 3-28. Activación de una clave desde el navegador chromium en ubuntu 10.04 43
Fig. 3-29. Creación del certificado raíz..... 44
Fig. 3-30. Creación del certificado con solicitud de firmado..... 45
Fig. 3-31. Firmado de la solicitud de certificado csr..... 45
Fig. 3-32. Pagina de presentación y logo de la estación PSII..... 47
Fig. 3-33. a) Visualización original b) Visualización personalizada..... 47
Fig. 3-34. Raspberry Pi Zero W 49
Fig. 3-35. Enumeración de los pines de la raspberry 49

Fig. 3-36. Vista frontal y posterior de la pantalla LCD-10168 50
Fig. 3-37. Diagrama de tiempos del protocolo SPI 50
Fig. 3-38. Botón de presión o push boton 51
Fig. 3-39. Led RGB 51
Fig. 3-40. Leds de estado del servicio dentro del botón de presión 52
Fig. 3-41. Impresora QUIAN ANJET80 QIT80170 52
Fig. 3-42. Selector de monedas 53
Fig. 3-43. Moneda de 5 pesos mexicanos 54
Fig. 3-44. Conexiones de los pines de la raspberry 54
Fig. 3-45. Aspecto general de las piezas interconectadas y funcionando 1. 55
Fig. 3-46. Aspecto general de las piezas interconectadas y funcionando 2. 55
Fig. 3-47. Aspecto general de las piezas interconectadas y funcionando 3. 56
Fig. 3-48. Lógica de funcionamiento del programa general. 57
Fig. 3-49. Lógica del programa para activar y cobrar una clave. 58
Fig. 4-1. Temperatura mayor a 42º sin ventilación. 61
Fig. 4-2. Freeradius preparado para procesar solicitudes de autenticación. 62
Fig. 4-3. Acceso aceptado para una clave vigente. 63
Fig. 4-4. Acceso rechazado para una clave expirada. 63
Fig. 4-5. Acceso rechazado para una clave no registrada o una clave mal configurada. 63
Fig. 4-6. Chillispot atendiendo la solicitud de conexión y autenticación. 64
Fig. 4-7. Certificado SSL funcionando en el servidor web 65
Fig. 4-8. Acceso al directorio de ejecución sin reglas de seguridad 66
Fig. 4-9. Acceso al directorio de ejecución con reglas de seguridad. 66
Fig. 4-10. Acceso con wget al directorio de ejecución con reglas de seguridad 66
Fig. 4-11. Re-direccionamiento al portal captivo antes de la autenticación. 67
Fig. 4-13. Velocidad de subida a una unidad de almacenamiento en red llamada sfv1. 69
Fig. 4-14. Pantalla de espera 70
Fig. 4-15. Pantalla de bienvenida 70
Fig. 4-16. Pantalla de selección de compra 71
Fig. 4-17. Pantalla de activación de la clave. 71
Fig. 4-18. Pantalla de clave activada 72
Fig. 4-19. Impresión de la clave activada en papel térmico 72
Fig. 4-20. Clave de usuario registrada en la base de datos. 73

Lista de tablas

Tabla 3.1. Planes de renta y su costo 33
Tabla 3.2. Modelo esquemático de la tabla radgroupcheck 33
Tabla 3.3. Modelo esquemático de la tabla radgroupreply 34
Tabla 3.4. Configuración de los valores de la tabla radgroupcheck 35
Tabla 3.5. Configuración de los valores de la tabla radgroupreply 35
Tabla 3.6. Modelo esquemático de radcheck 36
Tabla 3.7. Modelo esquemático de radreply 37
Tabla 3.8. Modelo esquemático de radusergroup 37
Tabla 3.9. Valores de radcheck 39
Tabla 3.10. Valores de radreply 39
Tabla 3.11. Valores de radusergroup 39
Tabla 3.12. Números asociados a los planes tarifarios 41

Lista de apéndices y anexos

Apéndice 1. Materiales empleados para la construcción del módulo de acceso inalámbrico ... 79
Apéndice 2. Materiales utilizados para la construcción del módulo de auto-servicio 81
Apéndice 3. Archivos modificados de freeradius..... 82
Apéndice 4. Configuración del archivo "/etc/apache2/default-ssl" 86
Apéndice 5. Configuración del archivo "/etc/apache2/default" 90
Apéndice 6. Código del portal captivo 91
Apéndice 7. Archivo de configuración del chillispot 101
Apéndice 8. Reglas del cortafuegos de la raspberry. 107
Apéndice 9. Asignación de los recursos de red a la interfaz USB de la impresora. 107
Apéndice 10. Código del script de arranque que inicia el programa de venta. 107
Anexo 1. Perforaciones hechas al gabinete del módulo de acceso inalámbrico. 108
Anexo 2. Medidas del panel de control del módulo de acceso inalámbrico. 109
Anexo 3. Circuito PCB de la tarjeta de arranque..... 110
Anexo 4. Contenido del archivo “registrar_planes.sql” 111
Anexo 5. Código generador de claves por volumen. 112
Anexo 6. Código del script activador de claves. 112
Anexo 7. Plantilla generadora de certificados SSL. 115
Anexo 8. Código html de la página de bienvenida del portal captivo. 120
Anexo 9. Código fuente del módulo de auto-servicio..... 131

Lista de abreviaturas y terminologías

CPU: Unidad de procesamiento central, por sus siglás en inglés.
APG: Software generador de claves aleatorias.
CA: Certificado SSL de autoridad.
DD-RWT: Sistema operativo usado en algunos módems y routers.
DHCP: Servidor de asignación de recursos de red.
IPTABLES: Cortafuegos o firewall de Linux.
NAS: Servidor de acceso a la red (Network Access Server).
OPENWRT: Sistema operativo usado en algunos módems y routers.
PHP: Lenguaje de programación web.
PORTAL CAPTIVO: Página web que captura todo el tráfico de datos antes de la autenticación.
PSI: Proveedor de servicios de Internet.
PSII: Proveedor de servicios de Internet inalámbrico.
RADIUS: Remote Authentication Dial-In User Service.
SPI: Serial Peripheral Interface.
SSL: Secure Socktes Layer.
TMEF: Tiempo medio entre fallos.
WISP: Wireless Internet Service Provider.
TERMINAL: Es el intérprete de comandos en Linux, también es conocida como consola o shell.
PROTOCOLO: Es un conjunto de reglas que dos equipos emplean para poder comunicarse entre sí.
PHP: Lenguaje de programación usado en la creación de páginas web dinámicas y es código que se ejecuta en el equipo servidor web.
BSS: Basic Service Set ó Conjunto Básico de Servicios, se refiere a una topología de redes inalámbricas Wi-Fi en donde el punto de acceso o access point gobierna la red.
EAP-TLS, EAP PEAP, EAP-TTLS: Protocolos de autenticación usado en redes.
WEP, WPA, WPA2, WPA-PSK: Protocolos de autenticación usados en redes inalámbricas.
IP: Protocolo de Internet.
TMEF: Tiempo medio entre fallos.
Mbps: Mega bits por segundo, indica la tasa de transferencia de datos.
Kbps: Kilo bits por segundo, indica la tasa de transferencia de datos.
DNS: Servidor de nombres de dominios.
LTSPICEIV: Software para simulación de circuitos electrónicos.

LIVEWIRE – PROFESIONAL EDITION: Software para simulación de circuitos electrónicos.
RGB: Led que contiene los tres colores básicos en un solo encápsulado: rojo (R), verde (G) y azul (B) por sus iniciales en inglés.
USB: En este trabajo se refiere a un dispositivo de comunicación serial que usa el protocolo USB para comunicarse con el CPU.
RJ45: Conector físico usado en conexiones de red cableadas con cables UTP.
WGET: Software para descargar contenido web a través de línea de comandos o desde la terminal.

Resumen

Esta tesis presenta el diseño del prototipo de una estación PSII (Proveedor de servicios de Internet inalámbrico) auto-atendida, cuyo objetivo es proporcionar servicio de Internet bajo la modalidad de renta por tiempo, por prepago y con sistema de venta automatizado o autoatendido. El sistema está orientado a personas o público en general que deseen comercializar el servicio de Internet administrando, a través de un procedimiento sencillo, el control complejo que implica la administración de una estación PSII; tales como la asignación de los recursos de red, ancho de banda, expiración del periodo de vigencia, etc. El prototipo de la estación PSII incluye el diseño y construcción de un módulo de autoservicio que tiene como propósito activar, cobrar y expedir de manera impresa, sin la intervención de un administrador, claves de acceso al usuario para poder acceder al servicio.

Abstract

This thesis presents the prototype design of a self-attended PSII (Wireless Internet Service Provider) station, whose objective is to provide Internet service under the modality of rent by time, by prepaid and with automated or self-service sales system. The system is aimed at people or the general public who wish to market the Internet service by administering, through a simple procedure, the complex control involved in the administration of a PSII station; such as the allocation of network resources, bandwidth, expiration of the period of validity, etc. The PSII station prototype includes the design and construction of a self-service module whose purpose is to activate, collect and issue in print, without the intervention of an administrator, access keys to the user to access the service.

Palabras Clave

Portal captivo, WISP, proveedor de servicios de Internet inalámbrico, sistema WISP autoatendido, mysql.

Capítulo 1. Introducción

La invención del Internet en el siglo XX revolucionó a escala global la manera de comunicarse entre personas, la comunicación entre ordenadores y hasta el comercio. Este capítulo presenta a grandes rasgos, la difusión, la manera en cómo se inventó el Internet y su desarrollo hasta llegar a ser lo que es actualmente: Una compleja red de comunicaciones entre computadoras.

1.1 Antecedentes

1.1.1 Surgimiento del Internet

El ser humano desde su aparición, creó o estableció una manera muy propia para comunicarse con sus semejantes, esto con el fin de interactuar entre sí y transmitir un mensaje. Remitiéndose un poco a la historia de la humanidad del ser humano, la forma de comunicación que se empleaba era a través de señales de humo, sonido de cuernos de animales, trompetas y campanas como señales de alerta, para los habitantes de una determinada comunidad o sociedad; con el paso del tiempo fue evolucionado y perfeccionando su sistema de comunicación valiéndose posteriormente de las cartas y telegramas cuando el destino de la información a transmitir estaba en lugares lejanos.

Pero desde tiempos remotos la humanidad ha estado inmersa en constantes guerras y conflictos sociales, esto ha propiciado a que el hombre desarrolle nuevos sistemas de comunicación para lograr la coordinación entre las distintas áreas de sus ejércitos para obtener la victoria o la supervivencia. Al inventarse el teléfono evoluciona la manera de comunicarse, ya que se crean códigos para identificar al destinatario, al remitente y viceversa, se reduce el tiempo de entrega -recepción del mensaje entre otros, poco a poco se va tejiendo una red de comunicación más sofisticada además de eficiente. Más tarde, al inventarse las computadoras el cambio en las comunicaciones se ve revolucionado por lo que ahora es posible no solo transmitir digitalmente audio sino que también imágenes, documentos, fotografías, etcétera; entonces se vuelve una compleja red de comunicaciones que es necesario inventar protocolos de redes de computadoras para poder intercambiar la información, posteriormente a esta compleja red se le denominaría Internet.

Aunque no se puede hablar con precisión de una fecha exacta sobre el surgimiento del Internet pero sí se puede hablar de un periodo tal como lo afirma el Dr. José Martínez De Velasco Farinós y lo plantea de la siguiente manera:

> El doctor Licklider, del Instituto Tecnológico de Massachussets (MIT), fue la primera persona que expuso la posibilidad de que hubiera una red de ordenadores a escala mundial. En unas notas, memoranda, escritas en agosto de 1962, proponía el establecimiento de una "red galáctica" formada por un conjunto de ordenadores interconectados entre sí y con la posibilidad de acceder rápidamente a datos y programas desde cualquier lugar.
> De este modo, nació en 1983 INTERNET como red de interconexión entre ARPANET, MILNET y CSNET, a las que se irán añadiendo posteriormente otras redes de EE.UU. y de otros países. [**1**]

De esta forma podemos darnos cuenta que el Internet es el resultado de los avances tecnológicos militares que surgió a partir de una necesidad de salvaguardar las comunicaciones en el ejército de EEUU, sin embargo, con el transcurrir del tiempo fue evolucionando hasta llegar a ser lo que es actualmente, al grado de convertirse en un servicio rentable desde el punto de vista comercial. [**1**]

1.1.2 Formas de difusión del Internet

El desarrollo del Internet como medio de comunicación entre computadoras dio lugar a la explotación comercial del mismo, de manera que en los tiempos actuales existen empresas en nuestro país cuya actividad comercial principal es la de lucrar con la distribución del servicio. Sin embargo, para que éstas puedan hacer llegar el servicio al domicilio del suscriptor es necesario instalar la infraestructura necesaria que permite la transmisión de la información entre el proveedor de Internet y el cliente y viceversa.
Existen dos maneras de transmitir la información entre computadoras: la cableada y la inalámbrica. En la transmisión cableada se pueden encontrar la transmisión por cable trenzado (Ethernet y telefónico), cable coaxial, fibra óptica. La Fig. 1-1 muestra las imágenes de los cables más usados para conexiones cableadas.

Fig. 1-1. Cables de transmisión a) coaxial, b) cable UTP y c) fibra óptica [2] [3] [4]

Mientras que para el caso inalámbrico se puede mencionar a los enlaces por satélite, Wi-Fi, acceso por banda ancha (telefonía celular), entre otros. Algunos dispositivos usuales son mostrados en la Fig. 1-2 (de izquierda a derecha, antena satelital, módem USB Wi-Fi y módem de banda ancha):

Fig. 1-2. Dispositivos de transmisión inalámbrica de Internet [5] [6] [7]

1.1.3 Empresas nacionales que difunden el Internet vía Wi-Fi

En nuestro país existen empresas que difunden el Internet de manera inalámbrica, algunas de ellas son:

WiCoin

Empresa que provee de Internet por medio de hotspots instaladados en casetas telefónicas públicas propias de la empresa. Tiene cobertura en Baja California Norte, Coahuila, Nayarit, Zacatecas, Jalisco, San Luis Potosí, Guanajuato, Michoacán, México, Guerrero, Veracruz, Puebla, Distrito Federal, Oaxaca y Chiapas. El mapa de la Fig. 1-3 muestra la cobertura a nivel nacional de wicoin.

Fig. 1-3. Cobertura Wi-Fi de Wicoin en México [**8**]

Megacable

Megacable es una empresa de televisión por cable, de telefonía y de Internet, actualmente presta el servicio de Internet inalámbrico por Wi-Fi sólo a sus subscriptores y sin costo alguno. La cobertura es a nivel nacional tal como lo muestra el mapa de la Fig. 1-4.

Fig. 1-4. Mapa de cobertura de Megacable en el país [**9**]

Xcien

De acuerdo con información obtenida en su página web “http://xcien.com”, Xcien es una empresa dedicada a la instalación de sistemas WISP y sólo opera en Nuevo León, Jalisco, Querétaro, Guanajuato y Tamaulipas.

1.2 Justificación

La sociedad actual no puede ser estática, es decir, tiene que estar acorde con los avances de la ciencia y la tecnología, debido a que los mismos tiempos lo exigen, por lo que una de las razones principales al elegir el presente tema de investigación denominado "Diseño y construcción de un sistema proveedor de Internet Wi-Fi en modalidad de prepago con venta automatizada" se debe principalmente porque el número de personas que puedan gozar del servicio de Internet como el que se pretende proporcionar se incremente, ya que a pesar que existen algunos proveedores de Internet (PSI) cableado tales como Infinitum de Telmex, Megacable, TotalPlay, Axtel, entre otros, existen lugares donde la infraestructura de las empresas PSI no tienen cobertura hasta el domicilio del subscriptor lo que imposibilita su acceso.

Tomando en cuenta esta necesidad y que el servicio de Internet actualmente es indispensable, el presente trabajo tiene como finalidad presentar una alternativa para facilitar a la población donde se carece de este servicio y su infraestructura, la implementación de un PSI inalámbrico (PSII de aquí en adelante) para dar cobertura hasta el hogar del subscriptor, porque a pesar de que ya existen comercialmente en el mercado, difícilmente se encuentra un modelo de venta automatizado, es decir, un sistema que controle automáticamente la vigencia del servicio al usuario y al mismo tiempo efectúe el servicio de cobranza, como el modelo que se presenta. Así mismo, se pretende que la implementación de este proyecto sea el prototipo de una máquina PSII que pueda ser adquirida por las empresas PSI como forma de expansión comercial o bien como un equipo para los negocios dedicados al comercio del Internet.

1.3 Planteamiento del problema

Actualmente se ofertan en el mercado equipos Wi-Fi que pueden funcionar, bajo ciertos requisitos, como PSII por medio de un firmware (conjunto de aplicaciones pequeñas que hacen funcionar a un dispositivo embebido en particular) genérico pre-cargado especializado tal como DD-RWT (de licencia comercial) u OPENWRT, los cuales no son propios de los fabricantes de los dispositivos.

DD-RWT y OPENWRT tienen la característica de controlar el acceso a Internet a través de un programa controlador incluido en el firmware, el cual necesita consultar las credenciales de acceso a la red a un sistema de autorización

dedicado para ésta tarea. Dentro del sistema de autorización se encuentra un conjunto de aplicaciones que corroboran las credenciales de acceso y que por sus tamaños y/o capacidad de procesamiento no caben o no se pueden incorporar en el firmware ya que los recursos tal como la memoria de almacenamiento o procesamiento de un dispositivo embebido (dispositivo dedicado a una o pocas tareas en particular) como lo es el módem Wi-Fi, están limitados, en consecuencia están alojados en sistemas o dispositivos externos al mismo. Como se puede ver, aún cuando ya existen dispositivos pre-configurados para funcionar como PSII éstos necesitan trabajar asociadamente con otros sistemas para operar plenamente debido a las limitantes anteriormente descritas.

Pero entonces, ¿de qué manera es posible implementarse un PSII completo dada las limitaciones de los dispositivos que hay en el mercado? La solución propuesta es conjuntar en un solo equipo todos los sistemas necesarios para su funcionamiento, así como el acondicionamiento físico del hardware para operar a temperatura ambiente a la interperie y la creación de un dispositivo interactivo especializado en el registro, cobro y liberación de las claves de acceso.

En este proyecto se automatizan las tareas que implica administrar manualmente un sistema PSII, desde el proceso de cobranza hasta la administración de los recursos de red tal como el control del tiempo concesionado, el ancho de banda, entre otros, puesto que la ingeniería tiene como objetivo encontrar soluciones a problemas no triviales que faciliten y sean de beneficio a nuestra sociedad.

1.4 Objetivos

1.4.1 General

Implementar una estación PSII con sistema de venta auto-atendido que permita a las personas de diversos contextos (rurales, semiurbanos y urbanos), acceder al servicio de internet en lugares donde la cobertura cableada de las operadoras PSI locales no cubren o son inexistentes.

1.4.2 Particulares

- Diseñar un sistema de venta totalmente automatizado con el cual pueda interactuar el cliente.
- Presentar el modelo de un sistema prototipo que pueda ser accesible económicamente para las empresas PSI, comercializadores del servicio de Internet y personas interesadas en la misma.

1.5 Metodología de solución

La metodología a seguir consiste en la documentación, estudio y análisis sobre el funcionamiento de un PSII a través de sitios de Internet, bibliografías e investigar y analizar a los que ya operan en el territorio nacional. En materia de hardware: diseñar y simular cuando sea necesario la implementación de partes o piezas electrónicas.

1.6 Descripción de los capítulos

Este trabajo está estructurado en cinco capítulos, los cuales se describen a continuación:

El capítulo I describe brevemente los antecedentes, el surgimiento del internet, sus formas de distribución y las empresas nacionales que lo comercializan.

El capítulo II corresponde al marco teórico y los conceptos básicos. En la misma sección se incluye la justificación, el objetivo general y los objetivos particulares.

El capítulo III trata sobre el diseño y construcción del software y hardware, que le permitirán a la estación PSII operar correctamente. Es decir, se explica el proceso que implicó dicho diseño.

El capítulo IV está enfocado a describir los resultados de las pruebas experimentales en tiempo real y los resultados obtenidos del funcionamiento de la estación PSII ante una carga de usuarios conectados simultáneamente.

El capítulo V trata sobre las conclusiones, observaciones, así como dar una explicación de efectos no deseados encontrados durante el proceso de pruebas. Por otro lado, también se mencionan las mejoras que se desean realizar a futuro al sistema prototipo.

Capítulo 2. La estación PSII

El presente capítulo aborda sobre los conceptos, la estructura y fundamentos generales de la estación PSII.

2.1 Concepto y definición general

Se define una estación PSII como un proveedor de Internet cuyo propósito es la difusión de la señal de Internet por alguna forma inalámbrica como el Wi-Fi, satelital y por banda ancha. La Fig. 2-1 muestra la perspectiva general de un PSII.

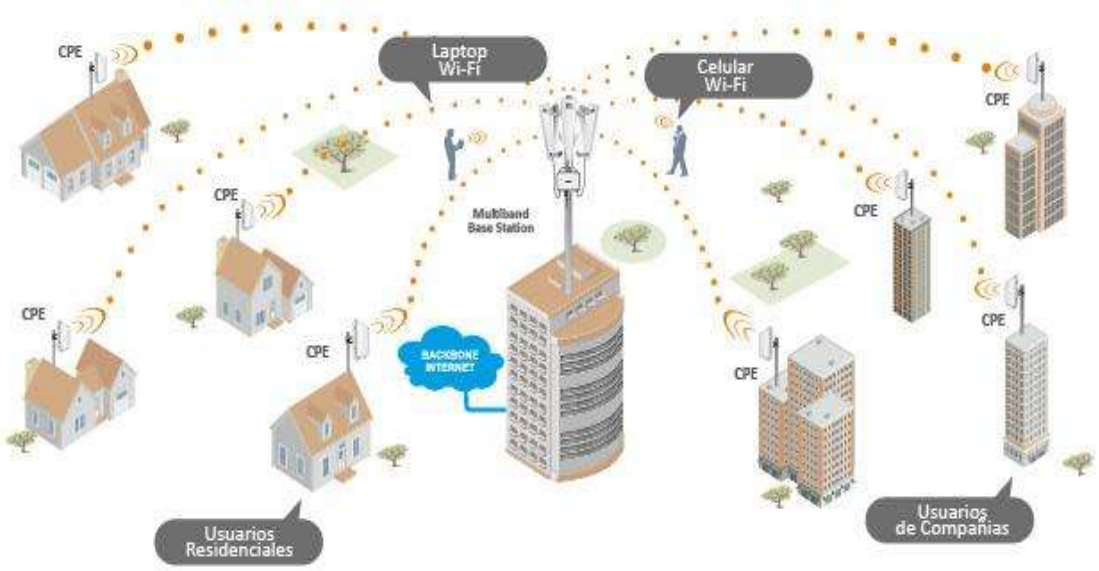


Fig. 2-1. Esquema ilustrativo de una estación PSII. [**10**]

2.2 Estructura general de una estación PSII

Un PSI está formado por conmutadores, servidores de acceso a la red, enrutadores, etcétera. El presente PSII está formado por una agrupación de equipos similares los cuales tienen la tarea de establecer la conexión con un PSI externo, permitir la conexión con el cliente y autenticarlo para determinar si es un usuario registrado. Esta descripción se modela mejor con la Fig. 2-2.

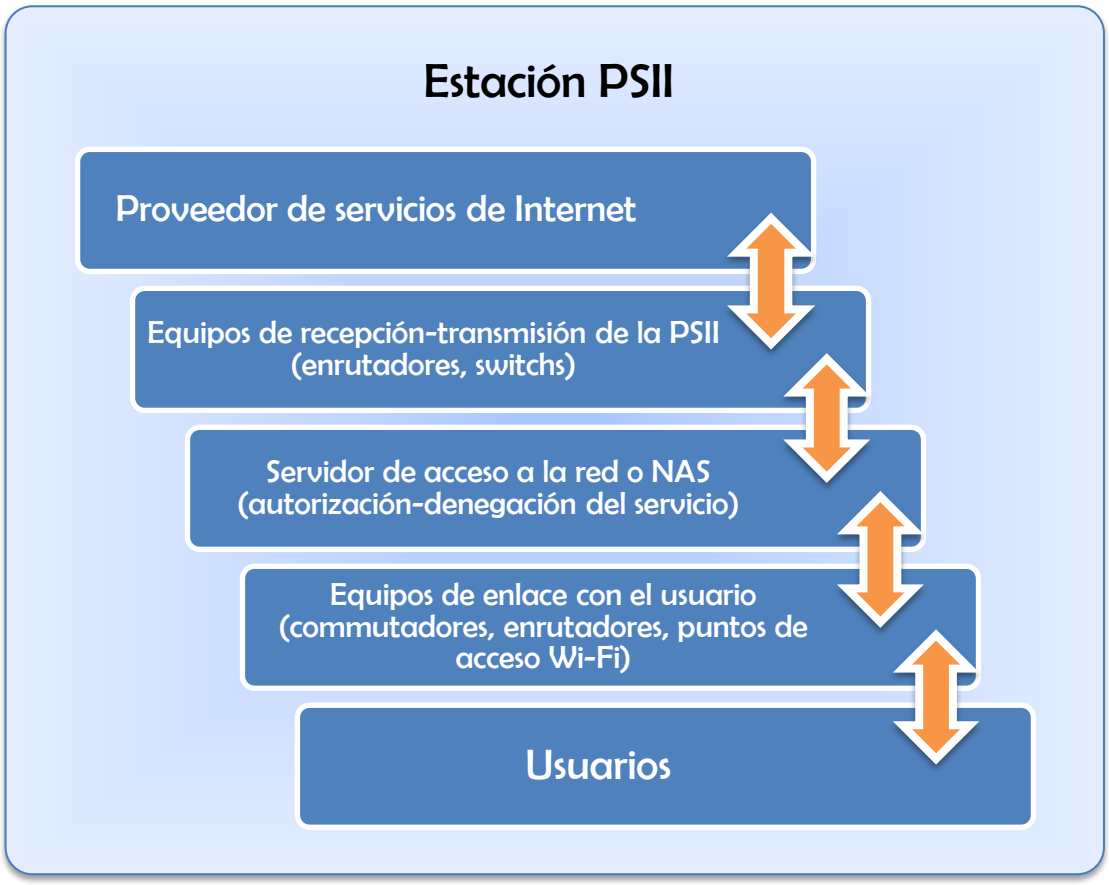


Fig. 2-2. Esquema general de la estación PSII

2.3 Elementos que conforman una estación PSII

Proveedor de servicios de Internet: En este bloque se encuentra la señal de Internet proveniente de algún PSI con la cual comercializará o difundirá la estación PSII.

Equipos de recepción-transmisión de la PSII (módems, switchs): Bloque en el cual se encuentran los equipos de conmutación quienes se encargan de establecer la conexión y re-direccionamiento del tráfico de datos entre la estación PSII y el proveedor de servicios de Internet.

Servidor de acceso a la red o NAS (autorización-denegación del servicio): Sección de la estación que se encarga del control de acceso y denegación del servicio a los usuarios, éste puede ser por medio de un protocolo de seguridad de la red tal como la autenticación (proceso de verificación y validación) WEP, WPA, RADIUS, etc.

Equipos de enlace con el usuario (commutadores, enrutadores, puntos de acceso Wi-Fi): Bloque formado por los dispositivos tales como los módems Wi-Fi, commutadores que controlan el tráfico de datos provenientes de los módems antes mencionados y lo re-direccionan al dispositivo difusor de Internet de la estación PSII.

Usuarios: Dispositivos finales a quienes se provee la señal de Internet proveniente de la estación PSII.

2.4 Seguridad de la estación PSII

Todo PSI cuenta con algún protocolo de autenticación que le permite aceptar las conexiones de sus subscriptores, esto se logra mediante una credencial de acceso que consiste en una clave y una contraseña dadas al subscriptor y sólo los que tengan autorizado el acceso podrán disponer del servicio para su uso.

2.4.1 Tipos de autenticación

Existen diversos tipos de autenticación para acceder a una red, algunos de ellos son WEP, WPA, WPA2, WPA-PSK que normalmente son los más usuales en conexiones inalámbricas Wi-Fi cotidianas y consisten en introducir la clave registrada en la estación de servicio Wi-Fi en el dispositivo cliente, la cual una vez introducida se determina si es correcta o no y se procede a establecer la conexión o denegarla según sea el resultado. Estos métodos tienen un inconveniente por dos razones principales cuando se trata de un sistema bajo el esquema de arrendamiento del servicio por tiempo, el primer inconveniente radica en que al ser una clave general propia de la estación Wi-Fi cualquier usuario que la posea podría conectarse a la misma y hacer uso del servicio, mientras que el segundo inconveniente radica en el tiempo de expiración de la concesión ya que al no haber un sistema que monitoree la vigencia de la misma el usuario podrá seguir conectándose a la red sin restricción alguna y no es viable desde el punto de vista comercial. Pero existen otros tipos de autenticaciones con políticas de acceso más estrictas en contraste con las antes mencionadas, tales como EAP-TLS, EAP PEAP, EAP-TTLS, etcétera, las cuales permiten o deniegan el acceso a la red en base al resultado obtenido de consultar la petición de conexión a un servidor de autenticación de acceso a la red o NAS. En el presente proyecto se usa EAP-TLS como método de autenticación el cual requiere de un nombre de usuario y una contraseña para poder ingresar a la red. De ésta manera se tiene control sobre la concesión de la

conexión al usuario, sin embargo, el uso de este método requiere del uso de certificados SSL y software que gestione el servicio.

2.5 El servidor NAS

Un servidor de acceso a la red o NAS "... es un punto de entrada que permite a los usuarios o clientes acceder a una red" y tiene como finalidad "... proteger el acceso a un recurso protegido, como puede ser una red telefónica, una impresora, o la salida directa a Internet", [**11**]. La Fig. 2-3 muestra el esquema general de funcionamiento de un servidor NAS.
En el presente trabajo el servidor NAS es el encargado de permitir o denegar el acceso a la red y está compuesto por un servidor RADIUS y un controlador de red llamado chillispot.

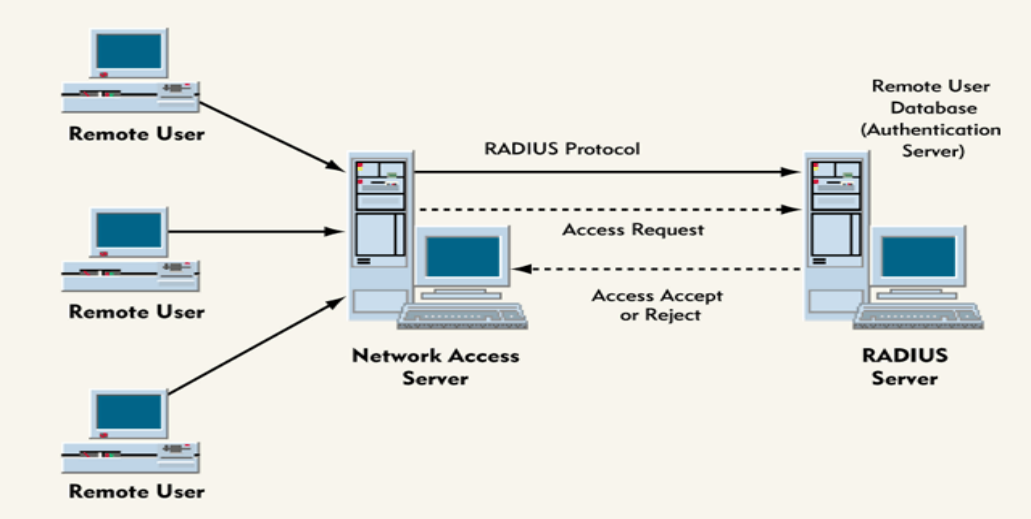


Fig. 2-3. Topología general para un servidor NAS [**12**]

Chillispot es un controlador de red que gestiona el ingreso a la red solicitando la credencial de acceso al cliente por medio de una interfaz web y consulta su validez en el servidor RADIUS. Por otra parte también administra las conexiones nuevas entrantes asignando los recursos necesarios que cualquier conexión de red necesita tales como la pasarela o puerta de enlace, IP y máscara de red. La topología de red típica de *chillispot* se muestra en la Fig. 2-4.

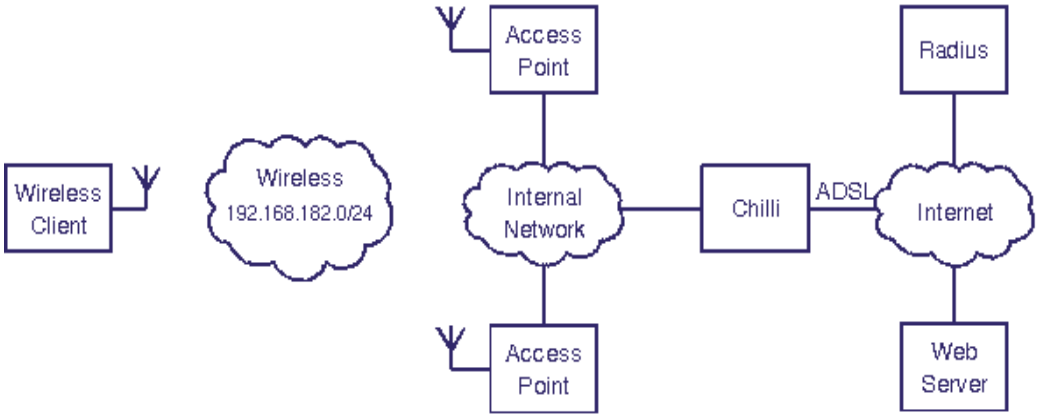


Fig. 2-4. Topología de red típica para chillispot [**13**]

2.6 El servidor RADIUS

RADIUS es un protocolo de "autenticación y autorización para aplicaciones de acceso a la red o movilidad IP" [**14**] cuyo propósito es la de comprobar o autenticar las credenciales de acceso a la red a través de un registro o por medio de una base de datos de claves de acceso válidas.
En este trabajo se usa freeRADIUS como software autenticador de credenciales el cual responde a las peticiones radius que Chillispot le solicite.

2.7 El certificado de seguridad SSL

2.7.1 El protocolo SSL

SSL es un protocolo de criptografía asimétrica usado en la transferencia de información entre un equipo servidor y un equipo cliente a través de Internet. Tiene como finalidad cifrar el intercambio de información en ambos equipos de modo que solamente entre ellos se entiendan, y por otro lado, verifica la identidad de un dominio al cual se está accediendo. Actualmente es muy usado en conexiones privadas tales como https.

2.7.2 El certificado de seguridad

Un certificado de seguridad SSL es un archivo digital que contiene información sobre la autenticidad de un sitio web en el que se incluyen datos como el correo electrónico de contacto, localización de la empresa propietaria, entre otros. Básicamente es lo que le da certeza a un cliente (navegador web) de que está accediendo a una página web verídica. Sin embargo, para que esto pueda ser corroborado y tener la evidencia de que el sitio es quién dice ser, se consulta a

una autoridad certificadora (CA) la cual certifica si el sitio web visitado es auténtico.

En los tiempos modernos la compra en línea así como la transferencia electrónica interbancaria han ocupado un lugar importante en la vida de las personas, razón por la que se han ideado protocolos de seguridad entre los cuales se encuentra SSL como medida de protección en la transmisión de datos sensibles en el mundo del Internet.

Existen dos razones fundamentales en el uso de un certificado SSL, la primera consiste en darle certeza jurídica a un sitio web y la segunda consiste en cifrar la información que se transfiere entre dos equipos, protegiendo de ésta manera las claves de usuario, contraseñas, claves interbancarias entre otros.

En este trabajo se recurre a la creación y emisión de un certificado de seguridad SSL auto firmado, con el propósito de sólo cifrar las claves de usuario entre los equipos clientes y la estación PSII a fin de que si alguien se dedicase a interceptarlas con herramientas de monitoreo de redes (sniffers) éstas no puedan ser obtenidas.

2.8 El servidor WEB

Al usar EAP-TLS como método de autenticación de la PSII es necesaria una interfaz visual interactiva con la cual interactuará el usuario para ingresar su clave de acceso a la red. En el presente trabajo esta interfaz es una página web o portal captivo alojada en el servidor de almacenamiento web o web host de la estación PSII por lo que se hace necesaria la instalación y configuración de alguno que responda a las peticiones al portal captivo, en este caso se usa el servidor web Apache.

2.9 La administración de las credenciales de usuario

Al tener varias cuentas de usuarios es necesario tener un software de administración con el cual gestionarlas para tener control sobre la conexión o prestación del servicio. En este trabajo se recurre al uso de MySQL como software de almacenamiento y consulta de las credenciales de acceso. Por otro lado, freeRADIUS tiene la opción de comunicarse con una base de datos para hacer las consultas por lo que se facilita esta tarea.

2.10 El entorno de trabajo

El sistema operativo usado para desarrollar el proyecto es Debian, por lo que todo el trabajo, software, desarrollo, programación y funcionamiento del PSII está hecho en y para ésta plataforma.

Capítulo 3. Diseño y Construcción de la Estación PSII

En el capítulo anterior se definió el concepto de la estación PSII. En ésta sección se presenta el diseño, construcción y funcionamiento de la misma.

3.1 Características de la estación

Al tratarse de un PSII automatizado y auto-atendido es necesario definir las características que debe tener la estación para su funcionamiento:

a) Interacción visual de fácil comprensión para el usuario al momento de contratar el servicio.
b) Forma y medio de cobro sencillo de usar.
c) Expedir de manera impresa el comprobante de acceso a la red.
d) Flexibilidad hacia el usuario en donde pueda decidir libremente el plan de renta que mejor se ajuste a sus necesidades.
e) Simplicidad en el acceso a la red.
f) Controlar el ancho de banda tanto de subida como de bajada para brindar un servicio de calidad a todos los usuarios.
g) Resistencia del equipo a las altas temperaturas que se alcanzan durante la temporada de calor.
h) Resistencia del equipo al flujo pluvial generado por la temporada lluviosa.

3.2 Estrategias de solución

Para lograr las características descritas en el apartado 3.1 se consideraron y diseñaron las estrategias que se mencionan a continuación:

- Para el inciso a) hacer uso de una pantalla LCD de 24 x 48 píxeles que tiene como propósito mostrar el costo de los periodos de renta del servicio a los usuarios y seleccionarlo presionando un botón.
- Para el inciso b) se usa una alcancía tragamonedas o coin selector el cual será el medio de cobranza de la estación PSII, ya que es sencillo de usar.
- En el caso de c) se hace uso de una impresora térmica para poder expedir la clave al usuario.
- Para d) y f) se han pre-configurado planes de renta que se le mostrarán al cliente.
- Para e) se usa una interfaz gráfica web que solicite la clave de usuario.

- Finalmente, para g) y h) se utilizan gabinetes metálicos de uso exterior los cuales contienen en su interior todas las piezas que hacen funcionar a la estación PSII plenamente.

3.3 Modelado general de la estación PSII

La estructura general de la estación PSII implementada se modela básicamente con el diagrama de bloques de la Fig. 3-1.

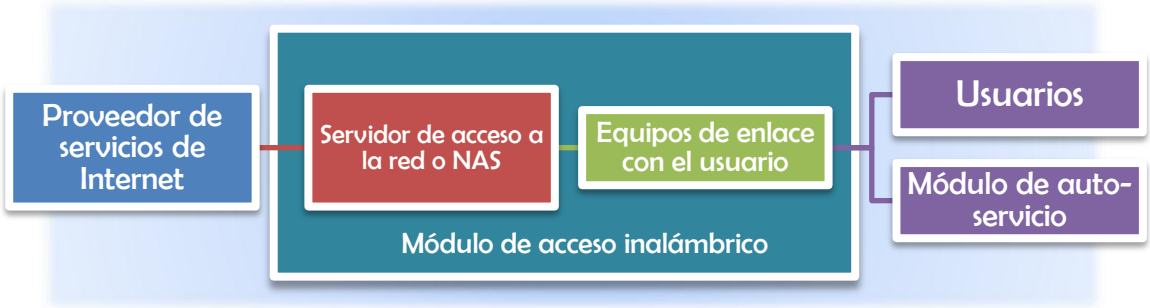


Fig. 3-1. Diagrama de bloques de la estación PSII

Como se aprecia en el diagrama de la Fig. 3-1 la estación cuenta con todos los componentes básicos con los que una estación PSII debe contar, con la excepción de que ahora cuenta con un elemento más que es el "Módulo de auto-servicio"; es precisamente éste módulo el que satisface las características de los incisos a), b) y c) del apartado 3.1.

3.4 Construcción del módulo de acceso inalámbrico

Esta sección aborda sobre la construcción de la módulo de acceso inalámbrico (ó hotspot). Los requisitos y materiales utilizados se enlistan en el apéndice 1.

3.4.1 La cubierta exterior o protección

La cubierta exterior es el gabinete metálico que protege a los componentes electrónicos de los factores climáticos tales como la lluvia o la temperatura excesiva. En el acondicionamiento de esta pieza se tomaron en cuenta los requisitos g) y h) de la sección 3.1. Para evitar el sobrecalentamiento de los componentes internos se diseñó un sistema de ventilación extractor de calor del interior del gabinete, mientras que por el lado de la resistencia a la lluvia se aplicó sellador hidráulico a las ranuras por donde pueda infiltrarse el agua.
Para tener control y ver el estado de funcionamiento del módulo de acceso inalámbrico se hizo el boceto de un panel de control el cual permite apagar,

reiniciar el CPU, así como ver el estado de la tarjeta Wi-Fi por medio de leds indicadores. En la Fig. 3-2 se muestra el gabinete metálico o cubierta exterior.

Fig. 3-2. Gabinete metálico o cubierta exterior [**15**]

3.4.1.1 Diseño del sistema de ventilación

El sistema de ventilación incluye al conjunto de ranuras [**16**] por donde ingresa el flujo de aire exterior al interior del sistema, al ventilador extractor y a la cubierta por donde se expulsa el aire caliente. El funcionamiento básico del sistema de ventilación consiste en hacer circular aire de temperatura ambiental en un circuito de entrada a salida recorriendo los lugares por donde se genera más calor que son el procesador del CPU, el procesador de la tarjeta Wi-Fi y la fuente de alimentación. En las figuras Fig. 3-3, Fig. 3-4 y Fig. 3-5 se puede observar la entrada, la salida y el circuito por donde circula el aire.

Fig. 3-3. Ranuras de entrada de aire exterior

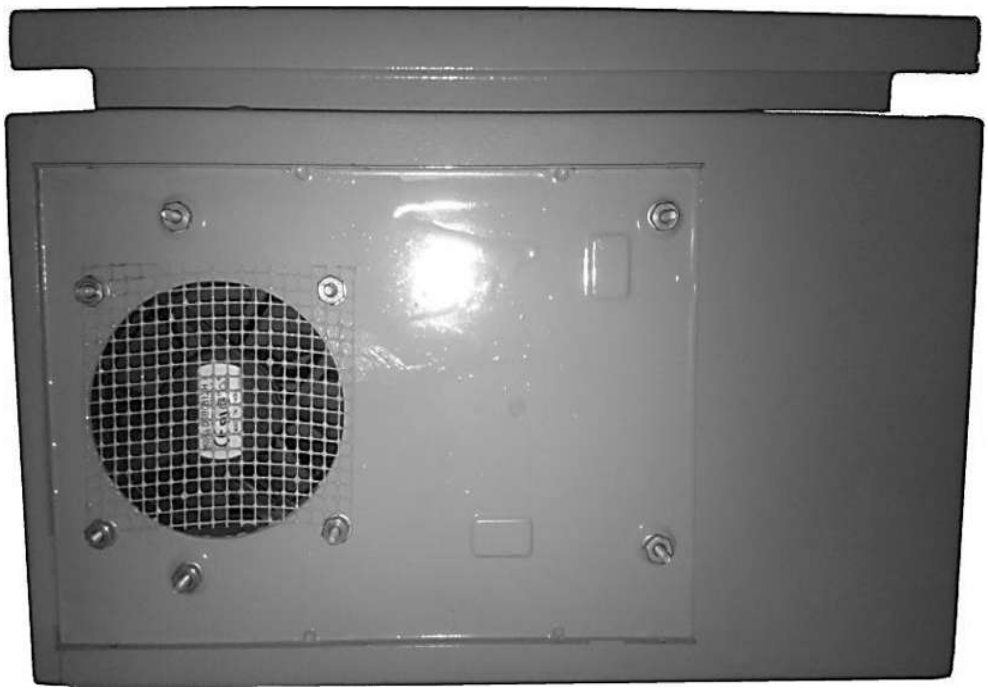


Fig. 3-4. Salida del aire caliente

Fig. 3-5. Circuito de circulación del aire al interior

En el anexo 1 se presentan los planos de corte, perforaciones y medidas del sistema de ventilación.

3.4.1.2 Diseño del panel de control exterior

El panel de control está intregado por el interruptor de alimentación de corriente alterna, el botón de reinicio del CPU, los leds indicadores (encendido, Ethernet y Wi-Fi) y los conectores de RF externos. Esta pieza tiene una función importante en el diseño del módulo de acceso inalámbrico ya que a través de ésta se puede apagar o reiniciar la tarjeta madre y observar si existe comunicación entre la misma y el punto de acceso. La comunicación entre las tarjetas puede observarse por medio del parpadeo de los leds de Ethernet y Wi-Fi; (de izquierda a derecha) el primero de ellos indica la conexión entre el CPU y el punto de acceso mientras que el segundo muestra la conexión entre el punto de acceso y los clientes. En la Fig. 3-6 se muestra el panel de control.

Fig. 3-6. Panel de control

En el anexo 2 se exponen los trazos perforaciones y medidas de la construcción de esta pieza.

Terminados todos los trabajos de acondicionamiento del gabinete, la perspectiva general obtenida se observar en la Fig. 3-7.

Fig. 3-7. Cubierta exterior o gabinete acondicionado

Finalmente, después de tener la cubierta acondicionada se procedió instalar las piezas y componentes electrónicos en su interior.

3.4.1.3 La tarjeta electrónica de encendido o controlador de arranque

Periódicamente es necesario apagar todo el módulo de acceso inalámbrico para evitar fallos de sincronización en las tarjetas de red o algún otro componente, debido a que existen diversos factores que pueden causar algún fallo tal como intermitencia de la señal en el canal de comunicación establecida con el cliente o bien porque todos los dispositivos tienen un tiempo medio entre fallos (TMEF). Como medida de prevención se ha diseñado una tarjeta electrónica que funciona como controlador o temporizador de encendido, cuya tarea es enviar un pulso de encendido a la tarjeta madre después de 10 segundos de apagarse ya que al encenderse inicia el arranque de la tarjeta Wi-Fi porque está ligada al encendido de la misma.

Para calcular el tiempo que se debe esperar el controlador de arranque antes de enviar el pulso se recurrió a la implementación de un circuito resistivo-capacitivo (*RC*) el cual tiene como principio de operación el aumento o decremento del voltaje en el capacitor (*C*) bajo el efecto de la resistencia (*R*) en un determinado intervalo de tiempo (*t*). Esta relación se expresa matemáticamente como:

$$t = RC \quad segundos \tag{3.1}$$

En donde *R* está en ohms, *C* en faradios y *t* en segundos. La ecuación 3.1 significa que idealmente existe un tiempo determinado por *R* y *C* en el que un capacitor de capacitancia *C* tardaría en cargarse al mismo nivel de voltaje (*Vc*) de corriente directa de la fuente (*VCD*) que lo excita, partiendo de que inicialmente el capacitor esté descargado, a este tiempo se le denomina "constante de tiempo *RC*"; mientras que la ecuación general para la carga de un capacitor partiendo del estado descarga es:

$$Vc = VCD(1 - e^{\frac{-t}{RC}}) \tag{3.2}$$

En la Fig. 3-8 b) se muestra la curva típica de carga del capacitor y en la Fig. 3-8 a) la representación electrónica de este tipo de circuito.

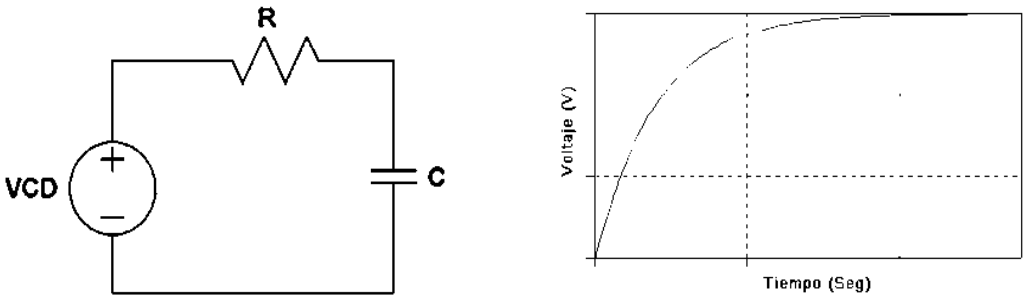


Fig. 3-8. a) Diagrama electrónico b) Curva típica de carga

Otro de los componentes en el diseño del controlador es el amplificador operacional, el cual puede funcionar como amplificador inversor, amplificador no inversor, comparador, etc. En el desarrollo de la tarjeta de encendido se usa como comparador, configuración que tiene la topología mostrada en la Fig. 3-9.

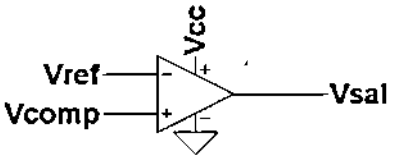


Fig. 3-9. Configuración como comparador del amplificador operacional

El modo de operación de la configuración anterior es:

$$\text{Si} \begin{cases} Vcomp > Vref \quad Vsal \approx Vcc \\ \\ Vcomp < Vref \quad Vsal \approx 0 \end{cases} \tag{3.3}$$

En donde:
Vref: Es el voltaje de referencia con el cual se desea comparar Vcomp.
Vcomp: Es el voltaje de entrada a comparar contra Vref.
Vsal: Es el voltaje a la salida del operacional.
Vcc: Voltaje de alimentación del operacional.

Haciendo un arreglo electrónico con el circuito *RC* y el comparador se tiene un circuito que conmuta dependiendo de la magnitud de voltaje *Vcom*. Éste es el principio de operación del controlador.

Para determinar los 10 segundos de espera, se sigue el siguiente procedimiento:

Como primer paso se utiliza un circuito *RC* en donde la constante de tiempo es mayor a 10 segundos. Por lo tanto, se necesita tener definido al menos dos valores, sea *t*=50 segundos y *C*=47uF, se eligió al capacitor como valor conocido ya que la resistencia es más fácil obtener a través de una combinación de arreglos en serie y paralelo ó bien por un resistor variable. Despejando *R* de la ecuación (3.1) y sustituyendo los valores definidos se tiene:

$$R = \frac{50\ segundos}{47uF} = 1063829.787234043\ \Omega \approx 1\ \text{M}\Omega \tag{3.4}$$

Al usar los valores de *t*, *R* y *C* anteriores y una fuente *VCD* de 12 volts, el circuito *RC* queda como se observa en la Fig. 3-10:

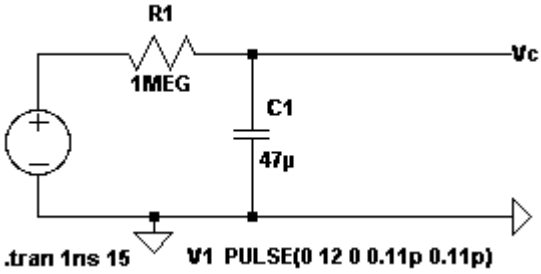


Fig. 3-10. Circuito RC

Al simular el circuito RC en LTSpiceIV en un lapso de 15 segundos el voltaje en el capacitor contra el tiempo se obtienen los resultados mostrados en la Fig. 3-11.

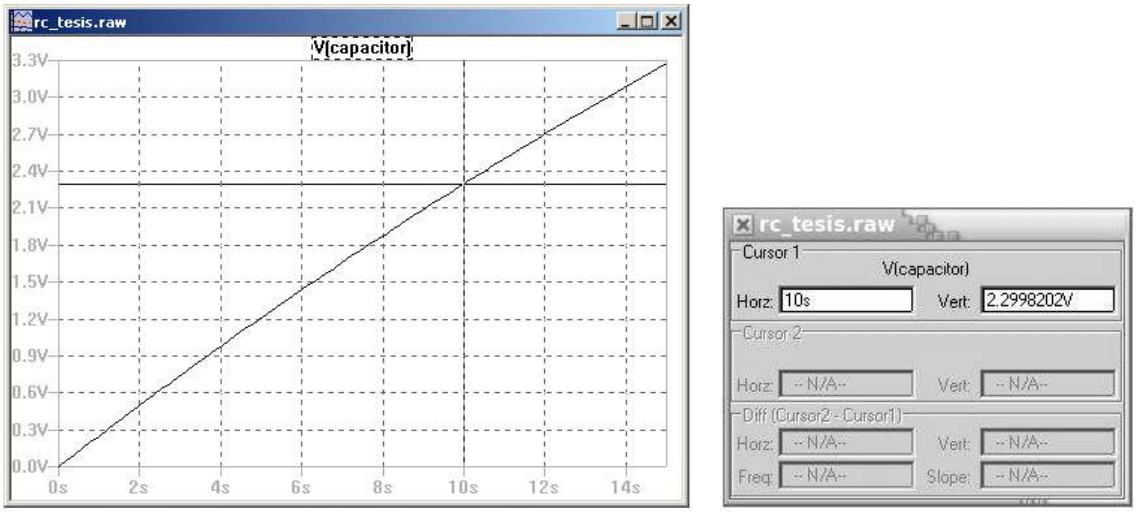


Fig. 3-11. Resultado de la simulación del circuito RC

La curva de la simulación anterior representa al voltaje en el capacitor respecto al tiempo y es el voltaje de comparación a la entrada de la terminal *Vcomp* del amplificador operacional. Al observarse en simulación, que para un tiempo de 10 segundos se tiene un voltaje en el capacitor de 2.2998 volts, dicha magnitud representa el voltaje de referencia que se le debe aplicar al comparador el cual una vez superado envía la señal de arranque o pulso de encendido. Para lograr que el voltaje en la entrada *Vref* sea de 2.2998 se implementó un divisor de tensión por medio de dos resistencias en serie tal como lo muestra la Fig. 3-12:

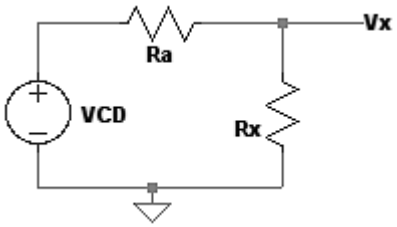


Fig. 3-12. Diagrama electrónico de un divisor de tensión

El siguiente paso es implementar el divisor de voltaje con dos resistencias, *Ra* de 10kΩ y *Rx* cuyo valor fue obtenido por medio de hacer el cálculo a través de la fórmula del divisor de tensión la cual se expresa como:

$$V_x = \left(\frac{Rx}{Rx + Ra}\right) VCD \tag{3.5}$$

En donde:
Vx es el voltaje en la resistencia *Rx.*
Rx es el valor de la resistencia por conocer.
Ra es la resistencia de valor conocido.
VCD es el voltaje aplicado a las resistencias.

Despejando a *Rx* se tiene

$$Rx = \frac{-Ra\,Vx}{Vx - VCD} \tag{3.6}$$

Como *Vx*=2.2998, *VCD*=12 volts y *Ra* = 10kΩ, sustituyendo valores se tiene:

$$Rx = \frac{-(10k\Omega)(2.2998\ volts)}{2.2998\ volts - 12\ volts} = 2370\ \Omega \tag{3.7}$$

Por lo que la resistencia *Rx* toma el valor de 2370 Ω, en consecuencia el circuito divisor de tensión ya con los valores definidos queda como se observa en la Fig. 3-13:

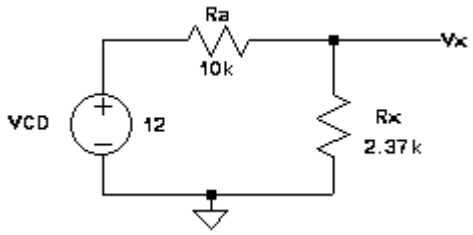


Fig. 3-13. Divisor de tensión con valores definidos

Al conectar el circuito RC y el divisor de voltaje con el comparador se tiene el circuito electrónico de la Fig. 3-14:

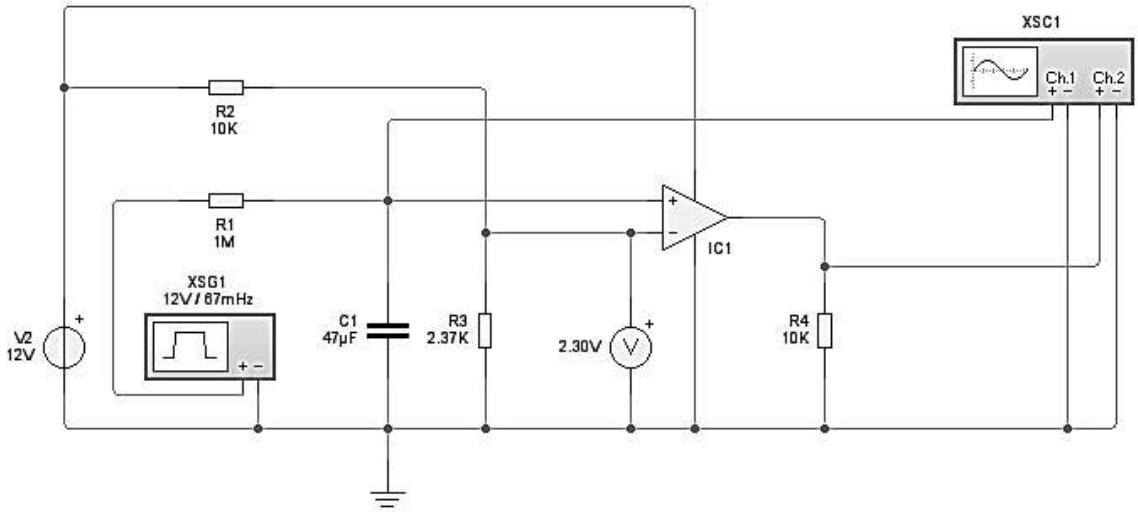


Fig. 3-14. Circuito retardador de encendido

Al simular en LiveWire – Profesional Edition el circuito de la Fig. 3-14 en un lapso de 15 segundos se tiene como resultado la gráfica de la Fig. 3-15.

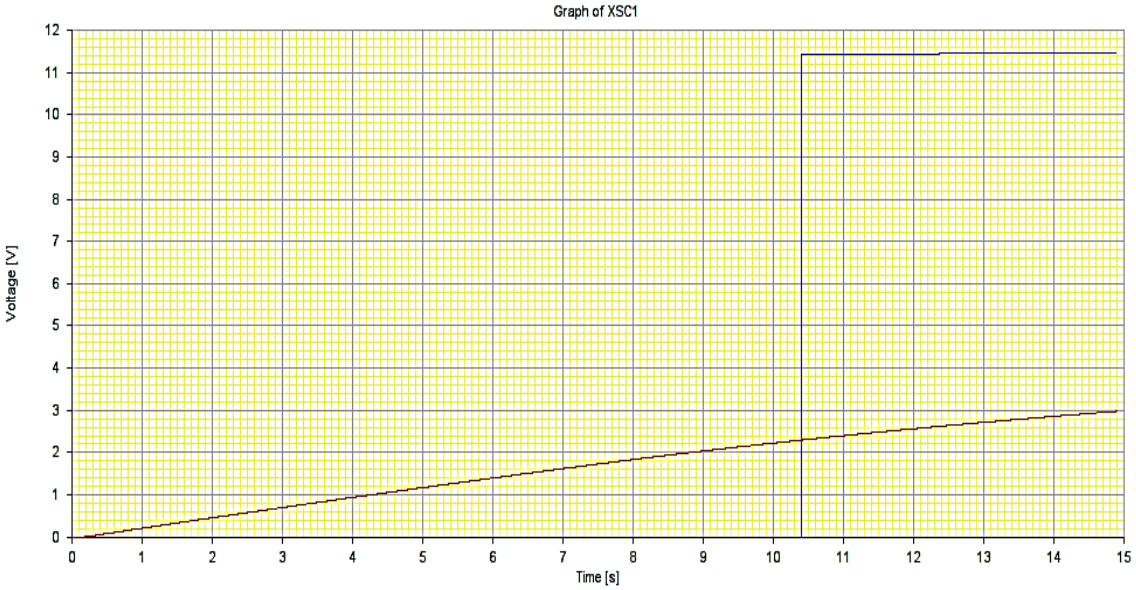


Fig. 3-15. Simulación de 15 segundos del circuito temporizador

La gráfica de la Fig. 3-15 muestra la variación del voltaje en el capacitor *Vc* y la terminal de salida de amplificador *Vsal*; tal como se observa, después de 10 segundos que equivalen a 2.3 volts el comparador se activa enviando la señal de arranque. Este es el principio básico de funcionamiento del controlador. La topología del controlador plenamente desarrollado se muestra en la Fig. 3-16.

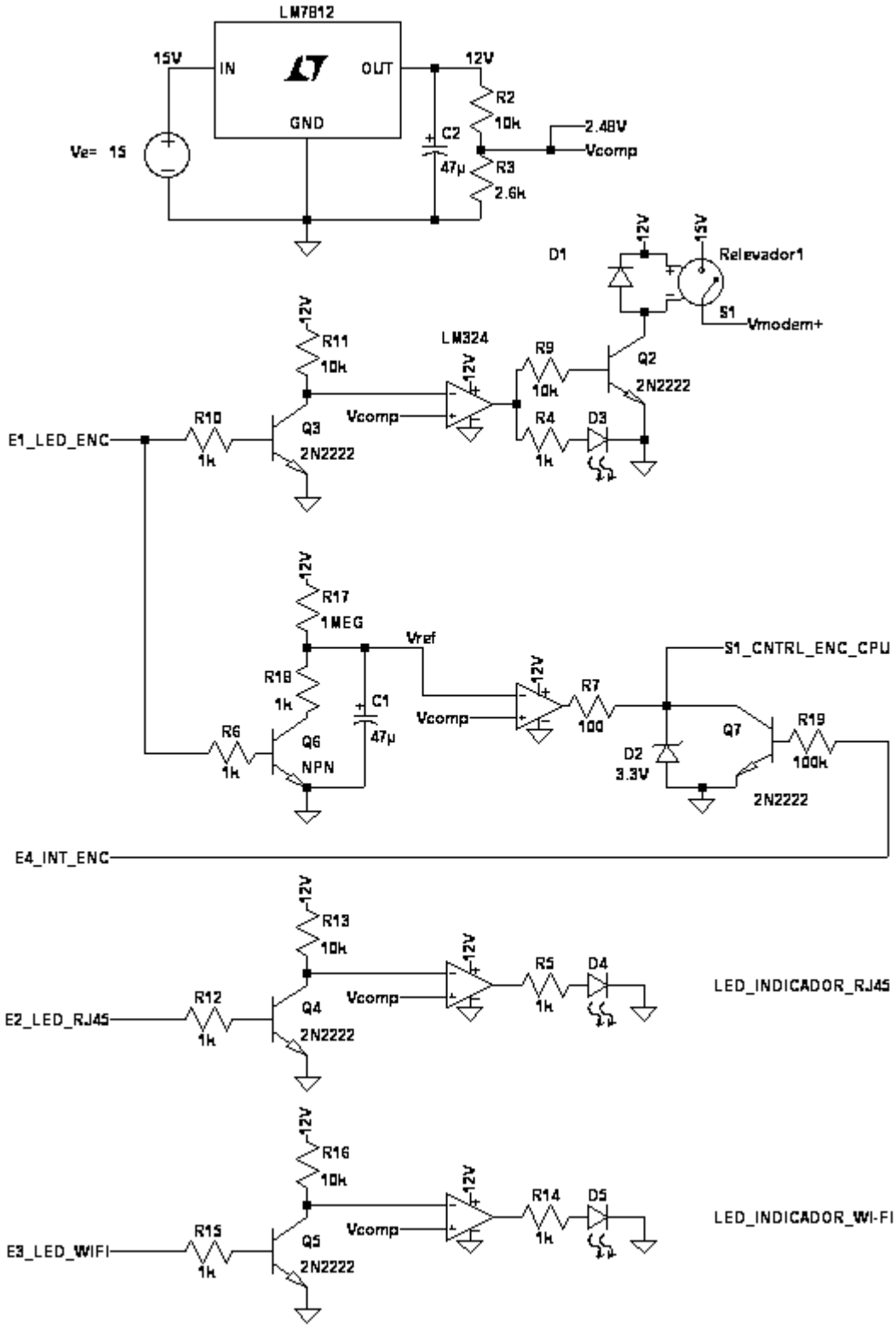


Fig. 3-16. Diagrama electrónico completo del controlador de encendido

El controlador para poder funcionar necesita el estímulo eléctrico del led de encendido de la tarjeta madre, esta señal mientras esté activa el transistor Q6 está activo y el capacitor se encuentra descargado, al apagarse el led de encendido el transistor se desactiva dando como resultado la carga del

capacitor y al sobrepasar los 2.3 volts se envía la señal de arranque, al arrancar la tarjeta madre el ciclo vuelve a iniciarse nuevamente.

La Fig. 3-17 muestra la implementación física del controlador, mientras que el anexo 3 se presenta el diseño del prototipo en PCB .

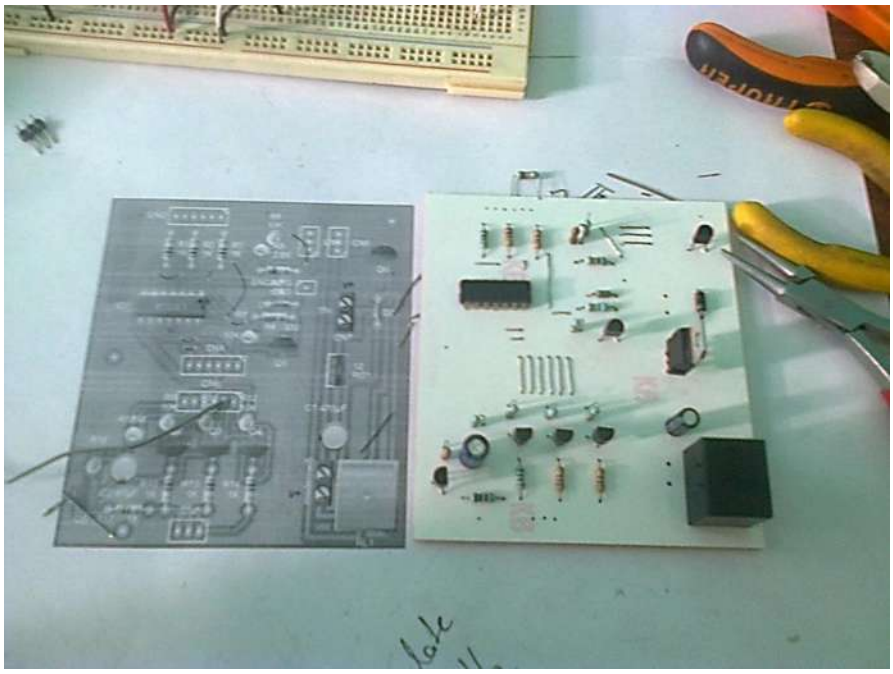


Fig. 3-17. Implementación en placa impresa del circuito controlador

3.4.1.4 Montaje y conexión de las tarjetas

Después del acondicionado externo del gabinete se procede a montar las tarjetas electrónicas al interior del mismo en el siguiente orden: 1) montaje de la tarjeta madre, la tarjeta de red y fuente de alimentación; 2) montaje de la tarjeta Wi-Fi y de circuito controlador. En las figuras Fig. 3-18, Fig. 3-19 y Fig. 3-20 se muestra este proceso.

Fig. 3-18. CPU, tarjetas de red y fuente de alimentación

En la Fig. 3-19 se fija la tarjeta madre, la tarjeta de red y la fuente de alimentación al chasis. Es importante mencionar que la tarjeta madre ya trae incorporada una tarjeta de Ethernet de 100 Mbps y es por donde se difunde la señal de Internet a los clientes.

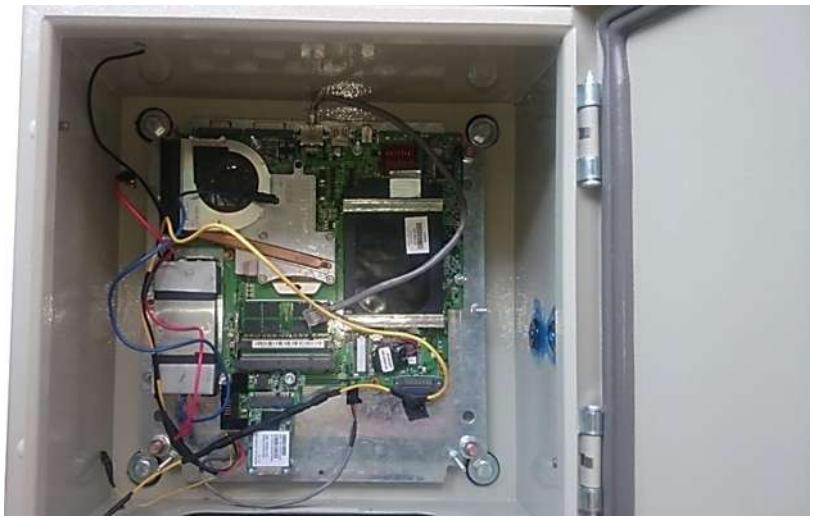


Fig. 3-19. Chasis atornillado en el gabinete

La Fig. 3-20 muestra el montaje de la tarjeta Wi-Fi que funciona como punto de acceso y el circuito controlador de arranque. Este paso se ilustra con las imagen siguiente:

Fig. 3-20. Tarjeta Wi-Fi y controlador de arranque montados.

Con los componentes instalados y la conexión entre las interfaces se concluye el proceso del acondicionamiento físico e instalación de las tarjetas electrónicas del módulo de acceso inalámbrico.

3.4.2 Instalación y configuración del software

3.4.2.1 El sistema operativo

El CPU del módulo de acceso inalámbrico ejecuta Ubuntu 10.04 Server como sistema operativo mientras que como software administrativo en la gestión de las claves de acceso: openssh-server, lamp server (linux apache, mysql y php), chillispot, freeradius y freeradius-mysql. Una vez listo el sistema operativo e instaladas las aplicaciones la siguiente tarea es configurar cada una de ellas, este proceso se describe en los puntos posteriores. En la Fig. 3-21 se puede apreciar la instalación del sistema operativo.

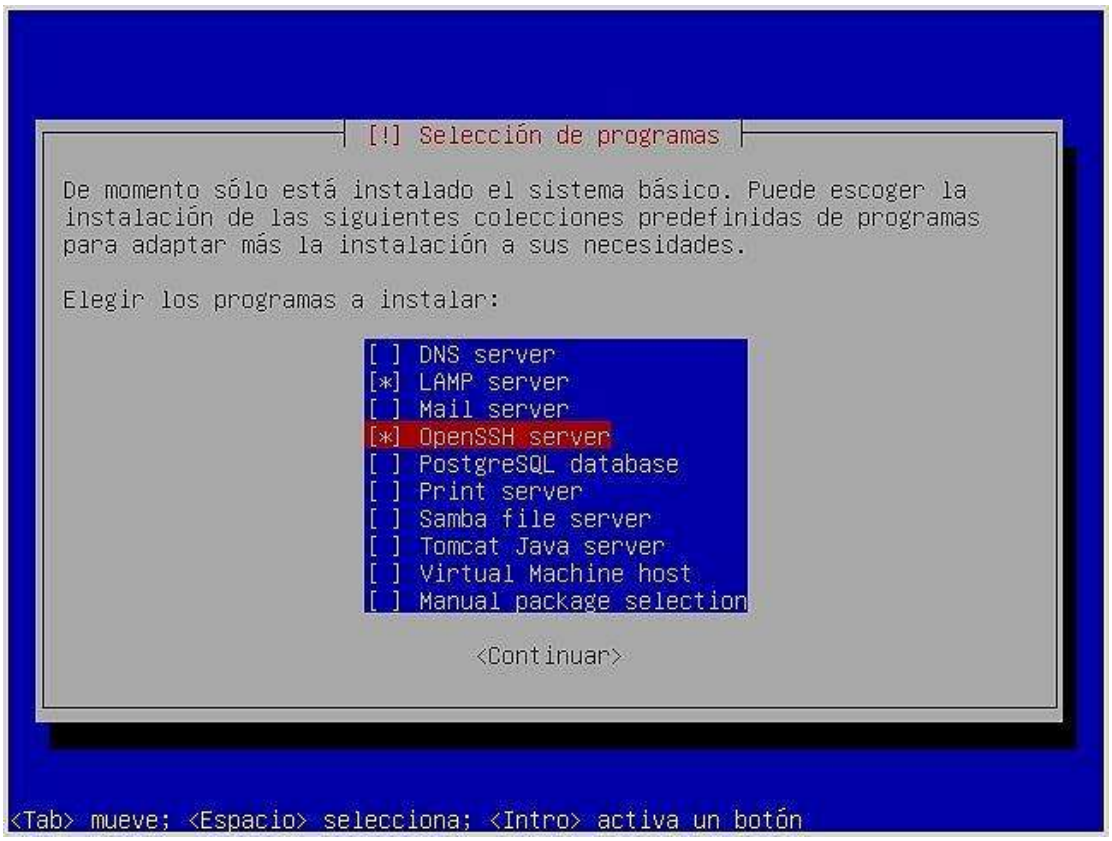


Fig. 3-21. Instalación de Ubuntu10.04 server, lamp y openssh server en el CPU.

3.4.2.2 La base de datos de MySQL

Chillispot trae un archivo llamado "schema.sql" alojado en el directorio /etc/freeradius/sql/mysql/ el cual contiene el modelo esquema de la base de datos de mysql cuya estructura incluye las tablas donde se almacena la información de las credenciales de acceso, tal como la vigencia, estado actual de la conexión, cantidad de datos consumidos, etcétera, que freeradius necesita para trabajar. La Fig. 3-22 muestra las tablas contenidas de la base de datos.

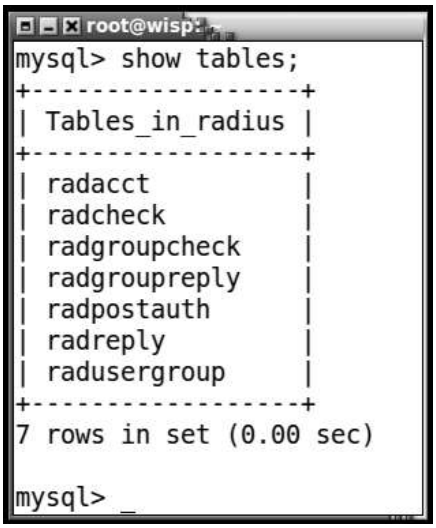


Fig. 3-22. Tablas de la base de datos

La importación del modelo esquema se realiza mediante el siguiente procedimiento:

1.- Crear primeramente la base de datos donde se almacenarán las tablas, introducciendo en la terminal las siguientes órdenes:

```
mysql -uroot –ptoor;
create database radius;
exit;
```

2.- Importación del modelo esquema en la base radius:

```
mysql -uroot -ptoor radius < /etc/freeradius/sql/mysql/schema.sql
```

Con las instrucciones anteriores se tiene ahora la base de datos creada y la estructura disponible para agregar planes de renta y credenciales de acceso.

3.4.2.3 Los planes de renta y su estructura en la base de datos

Tener uno o varios planes de renta es necesario antes de crear credenciales de acceso o claves de usuarios, porque ellas necesitan referencias del plan al cual van a pertenecer. Para poder registrar una clave de acceso son necesarios los siguientes parámetros: tiempo de concesión, ancho de banda de descarga, ancho de banda de subida, sesiones simultáneas por clave de usuario y tiempo de inactividad. En ésta sección se presentan los planes de renta con los cuales trabaja de manera pre-configurada la estación PSII.

En la Tabla 3.1 se proponen planes tarifarios para la concesión del servicio así como el costo de cada uno de ellos. El concepto de costo es simplemente como referencia, ya que en el módulo de auto-servicio se maneja dicho concepto porque al efectuar el cobro de una clave recibe monedas de una misma denominación, pero en el caso del módulo de acceso inalámbrico el costo carece de sentido debido a que su tarea solamente es permitir el acceso a la red, sin embargo, al tratarse de la creación de los planes tarifarios es necesario definir el precio de cada uno de ellos.

Tabla 3.1. Planes de renta y su costo

Plan de renta	Costo
2 horas	$ 5.00
1 día	$ 10.00
7 días	$ 20.00
15 días	$ 35.00
30 días	$ 50.00

Como se puede apreciar en la Tabla 3.1 cada plan tiene un precio, mismo que está relacionado con el número de monedas que recibirá el módulo de auto-servicio ya que todos son múltiplos de cinco por lo que el tipo de moneda usada es una de cinco pesos. Por ejemplo, para el plan de dos horas el cliente solo pagará con una moneda mientras que el caso del plan de 30 días tendría que pagar con diez.

Con planes tarifarios de renta definidos se procede a configurar en mysql las tablas que manejan los planes tarifarios. Las tablas que se configuran son: radgroupcheck y radgroupreply. La primera de ellas maneja el tiempo de expiración y la cantidad de sesiones simultáneas de la clave, mientras que la segunda administra el ancho de banda tanto de descarga como de subida y el tiempo de inactividad de la misma. El modelo del esquema de la tabla radgroupcheck se muestra en la Tabla 3.2.

Tabla 3.2. Modelo esquemático de la tabla radgroupcheck

Radgroupcheck			
nombre_del_plan	Session-Timeout	:=	tiempo_de_vigencia_en_segundos
nombre_del_plan	Simultaneous-Use	:=	sesiones_simultaneas

Los registros y su valores son:

- Session-Timeout: Parámetro asociado el tiempo de vigencia de una clave, está dado en segundos.
- Simultaneous-Use: Parámetro que indica cuántas sesiones simultáneas puede iniciar una clave.

El modelo del esquema de la tabla radgroupreply se muestra en la Tabla 3.3.

Tabla 3.3. Modelo esquemático de la tabla radgroupreply

Radgroupreply			
nombre_del_plan	WISPr-Bandwidth-Max-Down	:=	bps_de_descarga
nombre_del_plan	WISPr-Bandwidth-Max-Up	:=	bps_de_subida
nombre_del_plan	Session-Timeout	:=	tiempo_de_vigencia_en_segundos
nombre_del_plan	Idle-Timeout	:=	tiempo_inactivo_en_segundos
nombre_del_plan	Acct-Interim-Interval	:=	intervalo_de_registro_en_segundos

Los registros y sus valores son:

- WISPr-Bandwidth-Max-Down: Controla el ancho de banda de descarga, en bits por segundo.
- WISPr-Bandwidth-Max-Up: Controla el ancho de banda de subida, en bits por segundo.
- Session-Timeout: Campo que define el tiempo de vigencia de una clave, está dada en segundos.
- Idle-Timeout: Tiempo de inactividad del usuario conectado a la estación WISP, está dada en segundos.
- Acct-Interim-Interval: Intervalo de tiempo en segundos en que es actualizada la facturación del consumo en la base de datos.

Para configurar la Tabla 3.3 es necesario definir el ancho de banda tanto para descarga y subida de datos, por lo que en este trabajo se ha definido un ancho de banda de 512 kbps de descarga y 256 kbps de subida, aunque se pueden modificar posteriormente. Con la información de la Tabla 3.1 y las velocidades de transferencia anteriores, al sustituirse en las Tablas 3.2 y 3.3, dan como resultado las siguientes tablas:

De la tabla radgroupcheck o Tabla 3.2 se obtiene la Tabla 3.4 la cual se presenta a continuación:

Tabla 3.4. Configuración de los valores de la tabla radgroupcheck

Parámetro	Planes de renta				
Plan de renta	2 horas	1 día	7 días	15 días	30 días
nombre_del_plan	2horas	1dia	7dias	15dias	30dias
Session-Timeout	7200	86400	604800	1296000	2592000
Simultaneous-Use	1	1	1	1	1

De la tabla radgroupreply o Tabla 3.3 se obtiene la Tabla 3.5 la cual se muestra abajo.

Tabla 3.5. Configuración de los valores de la tabla radgroupreply

Parámetro	Planes de renta				
Plan	2 horas	1 día	7 días	15 días	30 días
nombre_del_plan	2horas	1dia	7dias	15dias	30dias
WISPr-Bandwidth-Max-Down	512000	512000	512000	512000	512000
WISPr-Bandwidth-Max-Up	256000	256000	256000	256000	256000
Session-Timeout	7200	86400	604800	1296000	2592000
Idle-Timeout	300	300	300	300	300
Acct-Interim-Interval	120	120	120	120	120

Con los valores de las Tablas 3.4 y 3.5 se procede a registrarlos en la base de datos radius. En el anexo 4 se presenta el código implementado en mysql para realizar éste procedimiento. La sintaxis para correrlo es la siguiente:

```
mysql -uroot –ptoor radius < $HOME/registrar_planes.sql;
```

Después de la instrucción anterior al revisar la base de datos se puede observar que las tablas radgroupcheck y radgroupreply de la base de datos radius, tienen los valores introducidos de acuerdo a la Tabla 3.5, la Fig. 3-23 muestra este paso.

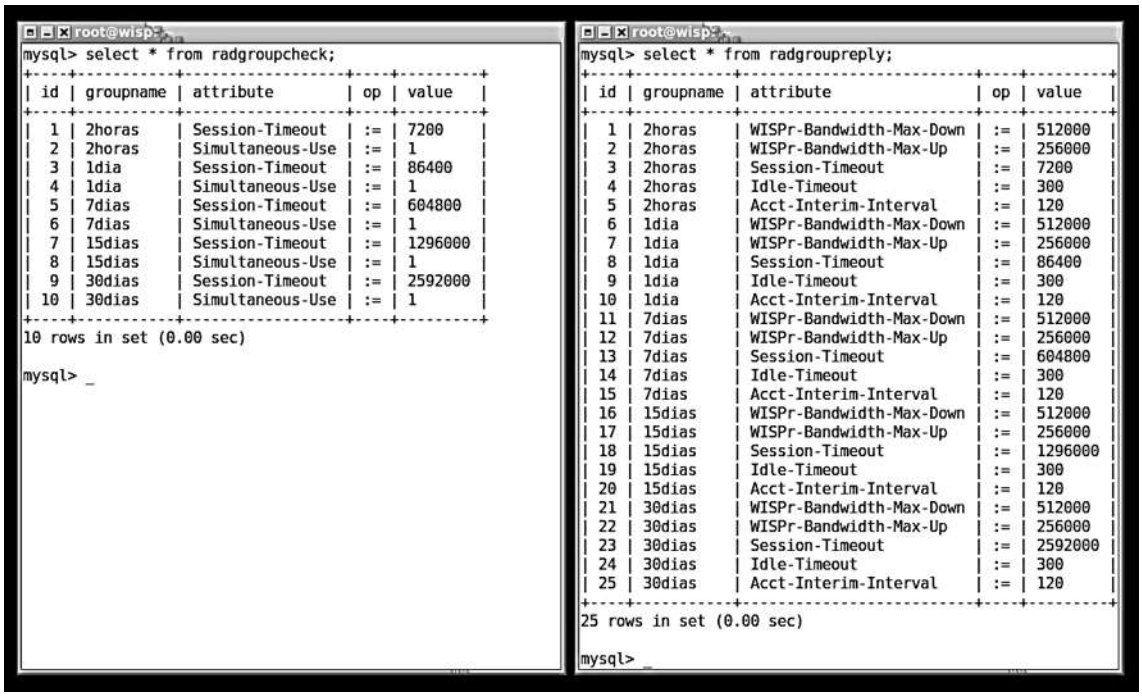


Fig. 3-23. Tablas de los planes de renta configuradas

3.4.2.4 Las claves de usuario

La clave de usuario es un código alfanumérico que se le proporciona al cliente para poder conectarse a la estación PSII. En este apartado se presenta la relación que guardan los planes de renta y las claves de usuario, las tablas que se deben de rellenar y sus configuraciones.

Las tablas en donde se registra la información necesaria para que una clave de usuario sea válida son: radcheck, radreply y radusergroup. Deben rellenarse según los modelos esquemáticos que se presentan en las Tablas 3.6, 3.7 y 3.8 ya que es la manera como freeradius las procesa.

La tabla radcheck

Tabla 3.6. Modelo esquemático de radcheck

Radcheck			
clave_de_usuario	Cleartext-Password	:=	Contrasenia
clave_de_usuario	Expiration	:=	Mes dd aaaa hh:mm:ss

Los registros que maneja son:

- Cleartext-Password: Maneja la contraseña de acceso, su valor es una cadena de caracteres.

- Expiration: Registro que tiene la fecha de expiración de la clave, el formato es "Mes dd aaaa hh:mm:ss", por ejemplo “December 31 1999 23:59:59”.

Uno de los inconvenientes detectatos al hacer este paso de manera automatizada es el formato de la fecha del sistema, ya que ésta se visualiza según el lenguaje del sistema el cual puede ser en español o inglés o a veces el formato no es el mismo. Para resolver esta situación con la compatibilidad de otros formatos de fecha se ha implementado un script (ver código en el anexo 6) el cual analiza la fecha e idioma del sistema para convertirlo al formato manejado por freeradius y poder registrarlo en la base de datos.

La tabla radreply

Tabla 3.7. Modelo esquemático de radreply

Radreply			
clave_de_usuario	WISPr-Session-Terminate-Time	:=	aaaa-mm-ddThh:mm:ss

El registro a configurar es:
WISPr-Session-Terminate-Time: Registro que almacena la fecha de expiración pero en formato "aaaa-mm-ddThh:mm:ss", como por ejemplo “1999-12-31T23:59:59”.

La tabla radusergroup

Tabla 3.8. Modelo esquemático de radusergroup

Radusergroup			
clave_de_usuario	nombre_del_plan_al_que_pertenece	:=	1

El registro a configurar es:
nombre_del_plan_al_que_pertenece: Es el nombre de alguno de los planes de renta existentes en la base de datos, por ejemplo “30dias”.

3.4.2.5 La generación de claves de usuario

Esta sección describe las características, el rango de caracteres válidos y la generación por volumen de las claves de usuario.

Una clave de usuario está formada por diez caracteres alfanuméricos comprendidos en los rangos de la [a-z], [A-Z] y [0-9] combinados aleatoriamente con el fin de hacerla más robusta frente a intentos de hackeo por medio de ataque de fuerza bruta o el por el uso de un diccionario de palabras, minimizando así la probabilidad de obtener una clave vigente por alguno de los métodos anteriores.
Las combinaciones posibles para una clave alfanumérica de diez caracteres está dada por la expresión:

$$combinaciones = 62^{10} \tag{3.8}$$

Donde 62 es la cantidad de caracteres válidos de una clave y 10 es el número de caracteres que forman la clave. De manera que haciendo uso de computadoras personales se tardaría demasiado tiempo para encontrar alguna combinación válida como clave de usuario.

En el presente trabajo, para generar las claves de usuario se hace uso del programa "apg" el cual es un software generador de claves aleatorias, una herramienta importante en el modelo automatizado de la estación PSII que se está diseñando. La sintaxis de código implementado es la siguiente:

```
apg -n 1 -m 10 -x 10 -E \'\*\_\!\"\-
\`\"\ç\{\}\,\.\i\!\"\·\$\%\&\/\(\)\=\?\¿\,\#\|\@\#\~\+\º\ª\\\^\<\>\: | sed 's/;/P/g' | sed
's/\[/I/g'| sed 's/\]/D/g'
```

El resultado obtenido se muestra en la Fig. 3-24:

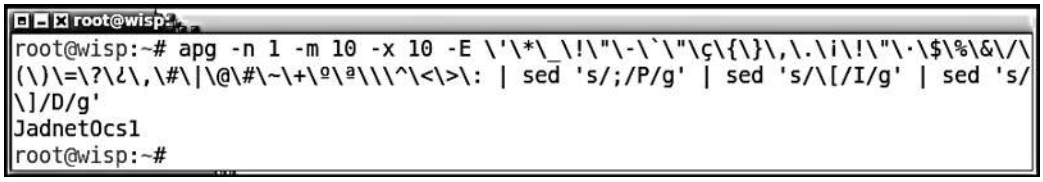


Fig. 3-24. Generación de un código alfanumérico con apg

Para la generación de claves por volumen se cambia el parámetro "-n 1" a "-n X" donde X es el número de claves que se deseen generar. En el presente trabajo se ha escrito un script (puede verse el código en el anexo 5) que genera las claves y las exporta a un archivo en texto plano (llamado claves.dat) creando un paquete de claves con el fin de tenerlas disponibles debido a que la

generación de las mismas tarda un lapso de tiempo (a veces prolongado) en regenerarse.

3.4.2.6 Activación de una clave de usuario en mysql

Para poder activar una clave en la base de datos es necesario asociarla con un plan de renta, si se toma la clave "Jadnet0cs1" como ejemplo, el plan tarifario de 30 días, "metanet" como contraseña de ingreso y suponiendo que la fecha de expedición de la clave es: 21 junio 2018 11:10:49 se tienen todos los requisitos necesarios para registrarla en la base de datos. Al sustituir estos valores en las Tablas 3.6, 3.7 y 3.8 (radcheck, radreply y radusergroup) se obtienen las Tablas 3.9, 3.10 y 3.11 respectivamente.

La tabla radcheck o Tabla 3.9:

Tabla 3.9. Valores de radcheck

Radcheck			
Jadnet0cs1	Cleartext-Password	:=	metanet
Jadnet0cs1	Expiration	:=	July 21 2018 11:10:49

Mientras que la tabla radreply o Tabla 3.10:

Tabla 3.10. Valores de radreply

Radreply			
Jadnet0cs1	WISPr-Session-Terminate-Time	:=	2018-7-21T11:10:49

Y finalmente la tabla radusergroup o Tabla 3.11:

Tabla 3.11. Valores de radusergroup

Radusergroup			
Jadnet0cs1	30dias	:=	1

El código implementado para activar la clave en la base de datos es el siguiente:

```
insert into radcheck(username,attribute,op,value)\
values('Jadnet0cs1','Cleartext-Password',':=','metanet');
insert into radcheck(username,attribute,op,value)\
values('Jadnet0cs1','Expiration',':=','July 21 2018 11:10:49');
insert into radreply(username,attribute,op,value)\
values('Jadnet0cs1','WISPr-Session-Terminate-Time',':=','2018-7-21T11:10:49');
```

```
insert into radusergroup(username,groupname,priority)\
values('Jadnet0cs1','30dias',1);
```

Al revisar las Tablas 3.9, 3.10 y 3.11 en mysql se tiene el resultado de la Fig. 3-25:

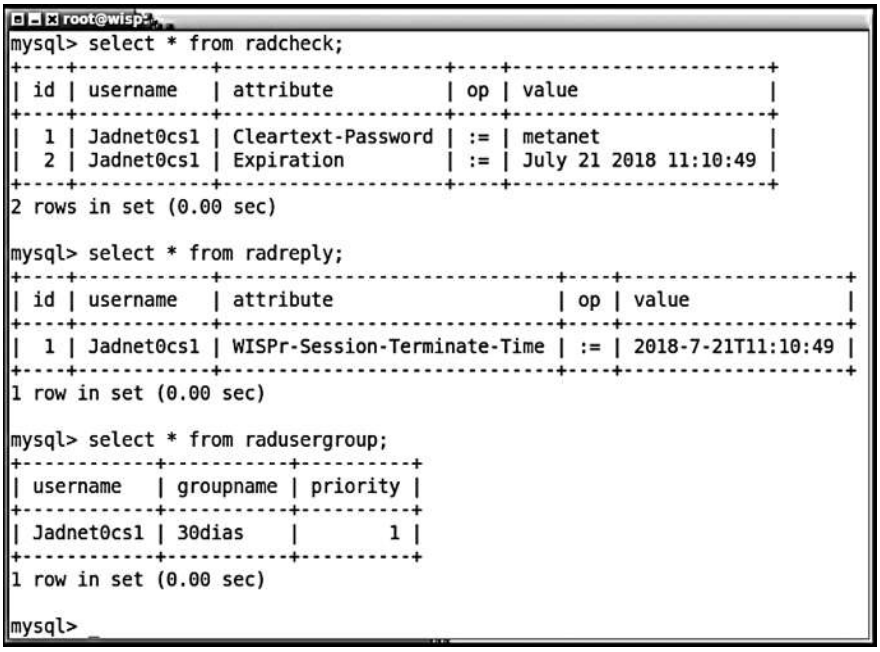


Fig. 3-25. Clave "Jdnet0cs1" registrada en la base de datos

Como se puede observar en la Fig. 3-25 la clave "Jdnet0cs1" ha quedado registrada en la base de datos de la estación PSII.

3.4.2.7 La activación automatizada de las claves de usuario

La activación de cualquier clave de manera manual sigue el mismo procedimiento descrito en la sección 3.4.2.6, sin embargo, se requiere de un software que realice el procedimiento de manera autónoma cuando un cliente desee adquirir una clave. Para lograr este objetivo se ha escrito un script llamado "alta_usuario.sh" (el código puede verse en el anexo 6) el cual contiene todas las instrucciones necesarias que el CPU del módulo de acceso inalámbrico necesita ejecutar para poder generar y activar una clave.

El script recibe como argumento un número que le indica a qué plan de renta deberá asociar la clave por activar. La Tabla 3.12 muestra la codificación implementada para relacionar los planes de renta con los números que puede recibir el script.

Tabla 3.12. Números asociados a los planes tarifarios

Nombre del plan	Código de activación
2horas	2
1dia	3
7dias	4
15dias	5
30dias	6

La estrategia anterior permite activar una clave de acceso inmediatamente por lo que solo se necesita el plan de renta a usar y la clave se extrae del paquete de claves.

El diagrama de bloques de la Fig. 3-26 muestra el proceso seguido por el script para registrar una clave en mysql.

Fig. 3-26. Proceso automatizado para activar una clave

Al ejecutar el script en la terminal de comandos se debe escribir también el número del plan de renta a utilizar, tal como se muestra enseguida:

sh alta_usuario.sh 6;

Y el resultado obtenido se aprecia en la Fig. 3-27:

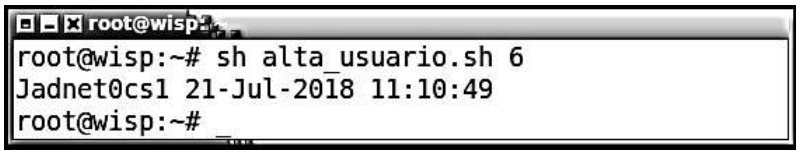


Fig. 3-27. Resultado del script activador de claves

La ejecución de script anterior da como resultado la clave activada y su fecha de expiración, ya que de acuerdo a la sección 3.4.2.6, la fecha de expedición de la clave del ejemplo es "21 junio 2018 11:10:49". Este resultado puede imprimirse en boletos, guardarse en archivo digital o enviarse de manera remota a algún dispositivo que quiera procesarla. En el presente trabajo es la información a enviar al módulo de auto-servicio que se encarga de solicitar la activación de las clave y entregársela al cliente.

3.4.2.8 Activación remota de claves

La activación remota radica en solicitar, a través de un protocolo diseñado para ésta tarea, la ejecución del script de activación desde el módulo de auto-servicio. Dicho protocolo consiste en ejecutar por medio del comando "wget" páginas web escritas en php que ejecutan el script, capturan el resultado y se lo entregan al módulo de auto-servicio a través del protolo http.

Debido a que se tienen varios planes de renta fue necesario hacer una página web para cada uno de ellos, de acuerdo a esto será el código que ejecutará cada una. El contenido de las páginas php es el siguiente:

```
<?php $x = shell_exec("sh alta_usuario.sh 6;"); echo "<pre>$x</pre>"; ?>
```

En el código anterior ejecuta el script "alta_usuario.sh" con el código de activación 6 que corresponde a la activación de una clave de usuario con una concesión de 30 días. El resultado obtenido es capturado y enviado como respuesta al módulo de auto-servicio.

Básicamente, todas las páginas php de activación de clave implementadas en este trabajo tienen la misma estructura lo único que cambia es el número conforme a la codificación de la Tabla 3.12.
A continuación se listan las páginas web php por sus nombres y el plan de renta asociado a cada una de ellas.

- 2horas: gacFecDeisjidjutsAkP.php

- 1dia: GhovMyWoabgeidryodEt.php
- 7dias: LophnetnoadKawbarEj9.php
- 15dias: muBlarefyeawdOdOjfij.php
- 30dias: UshaccyHepciwrefErs1.php

Al ejecutar la página "UshaccyHepciwrefErs1.php" a través del navegador web de un computador se tiene el resultado mostrado en la Fig. 3-28.

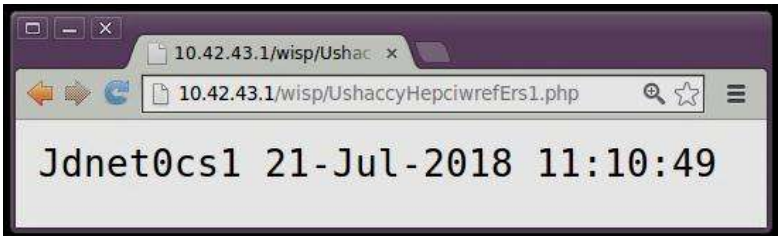


Fig. 3-28. Activación de una clave desde el navegador chromium en ubuntu 10.04

Como se puede observar en la ilustración 36, el resultado es la clave liberada, la hora y fecha hasta que será vigente la misma.
Para concluir ésta sección cabe mencionar que las páginas web php deben de estar alojadas en el mismo directorio de ejecución del script de activación.

3.4.2.9 Generación del certificado de seguridad SSL

En esta sección se explica el procedimiento seguido para crear el certificado de seguridad, por medio de OpenSSL, que le permite al servidor web de la estación PSII usar un canal de comunicación cifrada con el cliente al solicitar la credencial de acceso mientras que por otro lado es un requisito fundamental para el funcionamiento de freeradius.

El primer lugar se creó un certificado raíz o de autoridad con el objetivo de poder firmar certificados de clientes ya que es necesario que una autoridad certificadora los firme con su certificado de autoridad. Para conseguir esto se obtuvo la plantilla de certificado que OpenSSL trae como ejemplo disponible en /usr/share/doc/openvpn/examples/easy-rsa/2.0/openssl.cnf.

Posteriormente, se rellenaron los campos requeridos tales como el correo electrónico, dominio, vigencia, etcetéra. En el anexo 7 se muestra la plantilla configurada en dichos campos. Una vez hecho esto se procedió a generar el certificado raíz con la siguiente instrucción en la terminal:

openssl req -new -x509 -extensions v3_ca -keyout ./privado/cakey.pem -out cacert.pem -config ./openssl.cnf

En la Fig. 3-29 se muestra este paso.

```
juliocesar@bt:~/espacio_borrador/certificados_ssl/CA$ openssl req -new -x509 -extensions v3_ca -keyout ./privado/cakey.pem -out ca
cert.pem -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.
-----
Mexico [MX]:
Michoacan de Ocampo [Michoacan de Ocampo]:
Morelia [Morelia]:
Internet emonarca [Internet emonarca]:
Internet emonarca [Internet emonarca]:
e-Monarca [e-Monarca]:
metarmorfosisnet@yahoo.com [metarmorfosisnet@yahoo.com]:
juliocesar@bt:~/espacio_borrador/certificados_ssl/CA$ _
```

Fig. 3-29. Creación del certificado raíz

Lo ejecución anterior da como resultado los archivos “cacert.pem” y “cakey.pem”, el primero es el certificado raíz mientras que el segundo es la llave privada de la autoridad certificadora. La llave privada está directamente relacionada con una frase confidencial introducida por la autoridad certificadora y es solicitada para poder firmar certificados de cliente.

En segundo lugar se procedió a crear un certificado de cliente con solicitud de firmado (csr) para que una autoridad certificadora lo avale o lo firme. En este paso se ejecutó la siguiente instrucción:

openssl req -new -nodes -out solicitud_firmado_de_certificado.csr -config ./openssl.cnf

dando como resultado el archivo “solicitud_firmado_de_certificado.csr”. La implementación de este paso se muestra en la Fig. 3-30.

```
juliocesar@bt:~/espacio_borrador/certificados_ssl/CA$ openssl req -new -nodes -out solicitud_firmado_de_certificado.csr -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.
-----
Mexico [MX]:
Michoacan de Ocampo [Michoacan de Ocampo]:
Morelia [Morelia]:
Internet emonarca [Internet emonarca]:
Internet emonarca [Internet emonarca]:
e-Monarca [e-Monarca]:
metarmorfosisnet@yahoo.com [metarmorfosisnet@yahoo.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
www.emonarca.net [www.emonarca.net]:
juliocesar@bt:~/espacio_borrador/certificados_ssl/CA$ _
```

Fig. 3-30. Creación del certificado con solicitud de firmado

El tercer y último paso de esta sección es la firma del certificado con solicitud de firmado por la autoridad certificadora (en este caso es simulada por el autor del trabajo). En este punto es requerida la clave o frase de la autoridad certificadora que fue solicitada en el primer paso.

La ejecución de este paso se muestra en la Fig. 3-31.

```
juliocesar@bt:~/espacio_borrador/certificados_ssl/CA$ openssl ca -out certificado_wisp.pem -config ./openssl.cnf -infiles solicitu
d_firmado_de_certificado.csr
Using configuration from ./openssl.cnf
Enter pass phrase for ./privado/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: May 24 03:08:33 2018 GMT
            Not After : May 21 03:08:33 2028 GMT
        Subject:
            countryName               = MX
            stateOrProvinceName       = Michoacan de Ocampo
            organizationName          = Internet emonarca
            organizationalUnitName    = Internet emonarca
            commonName                = e-Monarca
            emailAddress              = metarmorfosisnet@yahoo.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                DC:E3:9B:C5:39:7A:1D:85:38:E2:57:25:32:A1:09:72:69:D9:F7:4D
            X509v3 Authority Key Identifier:
                keyid:FA:A6:E5:2F:C7:E4:98:6B:4F:18:AC:C0:BD:C3:AE:8B:37:83:B0:71

Certificate is to be certified until May 21 03:08:33 2028 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
juliocesar@bt:~/espacio_borrador/certificados_ssl/CA$
```

Fig. 3-31. Firmado de la solicitud de certificado csr

de lo cual se obtiene un archivo “certificado_wisp.pem” que es el certificado de seguridad firmado por la autoridad certificadora.

Con esto se da por terminada la generación del certificado de seguridad SSL, por lo que ahora ya pueden instalarse en freeradius y en el servidor web apache.

3.4.2.10 Configuraciones de freeradius

Los cambios realizados a ésta sección corresponden a la instalación de los certificados SSL generados en la sección 3.4.2.9 copiando los certificados al directorio "/etc/freeradius/certs" y registrarlos en el archivo radius.conf; rellenar los campos requeridos y/o descomentar líneas de los archivos de configuración. Los archivos de configuración modificados en ésta sección se encuentran ubicados en el directorio "/etc/freeradius/". En el apéndice 3 se muestra el contenido de cada archivo modificado.

3.4.2.11 Configuraciones del servidor web apache

Las configuraciones realizadas en este punto consisten en registrar los certificados SSL generados en la sección 3.4.2.9 y a la implementación de medidas de seguridad en el folder de ejecución del script de activador de claves. El registro de los certificados se hizo modificando el apartado "SSL Engine Switch" del archivo /etc/apache2/default-ssl. El apéndice 4 muestra el código final de configuración.

Por otra parte es necesario proteger el folder de ejecución del script activador para evitar accesos no autorizados al mismo y extraer información sensible como lo es el archivo fuente del script activador de claves. Para lograr lo anterior se aplican reglas de seguridad que impiden el acceso al folder de ejecución configurándolas, para este propósito, en el archivo "/etc/apache2/default". En el apéndice 5 se muestra el código final de este archivo. La configuración anterior impide ver a través de un navegador web el contenido del directorio "/directorio_ejecucion/" y solo puede accederse al contenido si previamente se conoce lo que hay dentro de él.

3.4.2.12 Configuraciones del portal captivo

Chillispot trae como parte de su instalación el modelo de una página web o portal captivo con la que el usuario deberá interactuar previamente para conectarse al servicio de internet [**17**], en ella se introduce la clave de usuario y contraseña. Sin embargo, en este proyecto se ha diseñado una página personalizada (ver código en el anexo 8) antes de la autenticación la cual es meramente por presentación hacia el cliente.

La implementación anterior se aprecia en la Fig. 3-32.

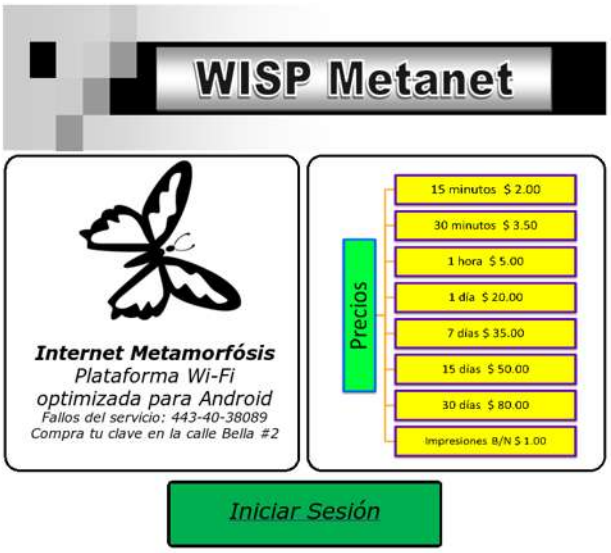


Fig. 3-32. Pagina de presentación y logo de la estación PSII

En particular, hay dos cambios que se realizan al código del portal captivo ubicado en /usr/lib/cgi-bin/hotspotlogin.cgi, los cuales son: primeramente establecer una contraseña compartida (uamsecret) entre el controlador de red y el portal captivo mientras que el segundo cambio consiste en modificar el código html de autenticación para hacer que sólo se introduzca la clave de usuario con la contraseña predeterminada evitando de ésta manera que el cliente rellene dos veces el formulario. El código final del portal captivo implementado puede observarse en el apéndice 6.

En la Fig. 3-33 a) y b) se presenta el aspecto visual del portal captivo del código original y del modificado.

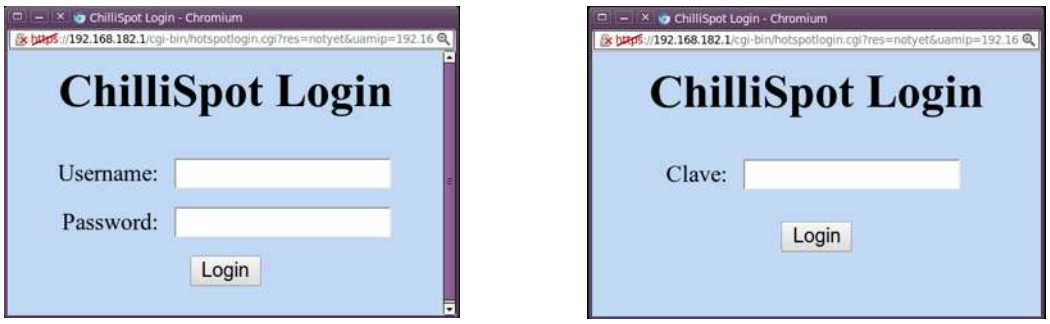


Fig. 3-33. a) Visualización original b) Visualización personalizada

3.4.2.13 Configuraciones del controlador de red o Chillispot

El controlador de red tiene sus valores de configuración en el archivo "/etc/chilli.conf". En dicho archivo se registran la ip raíz con la que trabajará la estación PSII, máscara de red, ip de los servidores dns y freeradius entre otros. Las variables y sus valores que se deben de configurar para su funcionamiento son los siguientes:

- IP raíz y máscara de la red con las que el sistema trabajará (net).
- IP de los servidores de nombres de dominio DNS (dns1 y dns2).
- IP del servidor radius (radiusserver1 y radiusserver2).
- Contraseña compartida de autenticación entre Chillispot y freeradius.
- Interfaz que difundirá la señal de Internet (dhcpif).
- URL de la página web o portal captivo(uamserver).
- URL de la página de inicio (uamhomepage).
- Contraseña compartida entre chillispot y el portal captivo (uamsecret).
- Direcciones y/o dominios a los cuales se puede permitir el libre acceso sin autenticación previa (uamallowed).

En el apéndice 7 se presentan las configuraciones finales del archivo "/etc/chilli.conf" con las que trabaja el controlador de red o chillispot.

Para un inicio automático de chillispot al arranque del sistema se deben de aplicar las siguientes instrucciones en la terminal de comandos:

```
sed -i "s/ENABLED=0/ENABLED=1/g" /etc/default/chillispot;
cp /usr/share/doc/chillispot/firewall.iptables /etc/init.d;
cd /etc/init.d/ && update-rc.d firewall.iptables defaults;
```

Finalmente, los cambios tienen efecto al reiniciar el CPU.

3.5 Construcción del módulo de auto-servicio

La construcción del módulo de auto-servicio es presentado en prototipo, es decir, sólo se presentan los componentes que la forman ya que el desarrollo de un modelo es complejo dado que se requieren recursos económicos para su construcción con los cuales el autor no cuenta. Por consiguiente, se deja como un trabajo a futuro. En el apéndice 2 se muestra la lista de materiales utilizados para su construcción.

3.5.1 Construcción del hardware del prototipo

3.5.1.1 Raspberry Pi Zero W

La tarjeta Raspberry Pi Zero W (de aquí en adelante raspberry) se alimenta con 5 VCD, es la responsable de ejecutar el programa de venta que atiende la solicitud de compra del usuario, a través de los componentes conectados a sus pines o terminales de propósito general (GPIO). En la Fig. 3-34 se muestra la tarjeta utilizada en este trabajo.

Fig. 3-34. Raspberry Pi Zero W [**18**]

Los nombres y enumeración de los pines GPIO se muestran en la Fig. 3-35.

Pin#	NAME	NAME	Pin#
01	**3.3v** DC Power	DC Power **5v**	02
03	**GPIO02** (SDA1 , I2C)	DC Power **5v**	04
05	**GPIO03** (SCL1 , I2C)	**Ground**	06
07	**GPIO04** (GPIO_GCLK)	(TXD0) **GPIO14**	08
09	**Ground**	(RXD0) **GPIO15**	10
11	**GPIO17** (GPIO_GEN0)	(GPIO_GEN1) **GPIO18**	12
13	**GPIO27** (GPIO_GEN2)	**Ground**	14
15	**GPIO22** (GPIO_GEN3)	(GPIO_GEN4) **GPIO23**	16
17	**3.3v** DC Power	(GPIO_GEN5) **GPIO24**	18
19	**GPIO10** (SPI_MOSI)	**Ground**	20
21	**GPIO09** (SPI_MISO)	(GPIO_GEN6) **GPIO25**	22
23	**GPIO11** (SPI_CLK)	(SPI_CE0_N) **GPIO08**	24
25	**Ground**	(SPI_CE1_N) **GPIO07**	26
27	**ID_SD** (I2C ID EEPROM)	(I2C ID EEPROM) **ID_SC**	28
29	**GPIO05**	**Ground**	30
31	**GPIO06**	**GPIO12**	32
33	**GPIO13**	**Ground**	34
35	**GPIO19**	**GPIO16**	36
37	**GPIO26**	**GPIO20**	38
39	**Ground**	**GPIO21**	40

Fig. 3-35. Enumeración de los pines de la raspberry [**19**]

Como interfaz entre la raspberry y el usuario se usa una pantalla LCD de 84x48 píxeles, un botón de presión o push boton, un selector de monedas o coin selector, leds indicadores de estado y una impresora térmica.

3.5.1.2 Pantalla LCD de 84x48 píxeles.

Es la interfaz visual de salida en donde se muestra al cliente los planes de renta y su costo, así como los detalles de la clave cuando se ha comprado. Para esto se una la pantalla LCD-10168 como la mostrada en la Fig. 3-36.

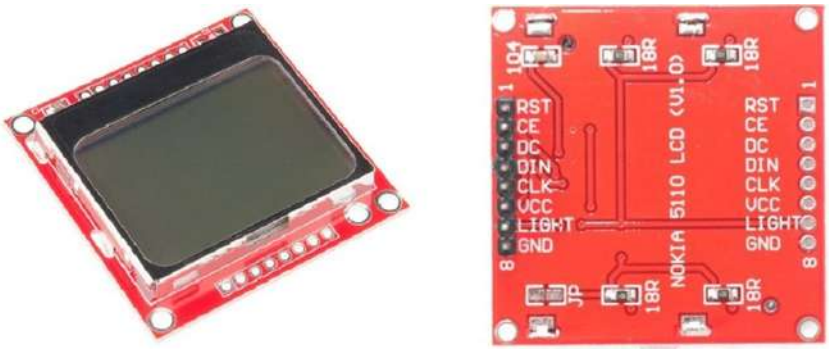


Fig. 3-36. Vista frontal y posterior de la pantalla LCD-10168 [**20**] [**21**]

El dispositivo anterior se controla, configura y funciona por medio del protocolo SPI, el cual tiene el diagrama de tiempo de acorde a la Fig. 3-37.

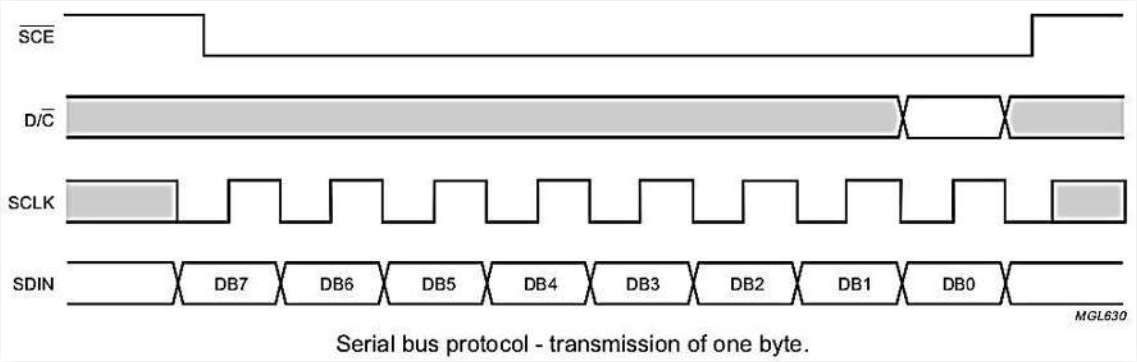


Fig. 3-37. Diagrama de tiempos del protocolo SPI [**22**]

El código de configuración y uso de la pantalla puede verse en el anexo 9.

3.5.1.3 Botón de presión o push boton

Este componente es la interfaz con la cual la raspberry detecta cuando el usuario desea comprar un plan y cuando ha terminado su compra, ya que al

presionarlo envía una señal eléctrica a la misma. El botón usado en este trabajo es como el que se muestra en la Fig. 3-38.

Fig. 3-38. Botón de presión o push boton [**23**]

3.5.1.4 Leds indicadores

Los leds indicadores tienen como objetivo indicar visualmente al cliente si está disponible el servicio del módulo de auto-servicio, para esto se han configurado de la siguiente manera: cuando inicia el programa se enciende el led rojo parpadeando con el verde apagado indicando que apenas se está estableciendo la conexión (fuera de servicio) con el módulo de acceso inalámbrico; cuando la conexión se ha establecido (en servicio) se enciende de manera continua el led verde y se apaga el rojo; finalmente, en caso de que se pierda la conexión (fuera de servicio) se enciende y parpadea el led rojo mientras que el verde se apaga hasta que la conexión se restablezca.
Los diodos leds están encapsulados en uno solo llamado led RGB y es como el que se muestra en la Fig. 3-39.

Fig. 3-39. Led RGB [**24**]

El led RGB está ubicado al interior del botón, por lo que al encenderse le dan un aspecto brillante tal como se muestra en la Fig. 3-40:

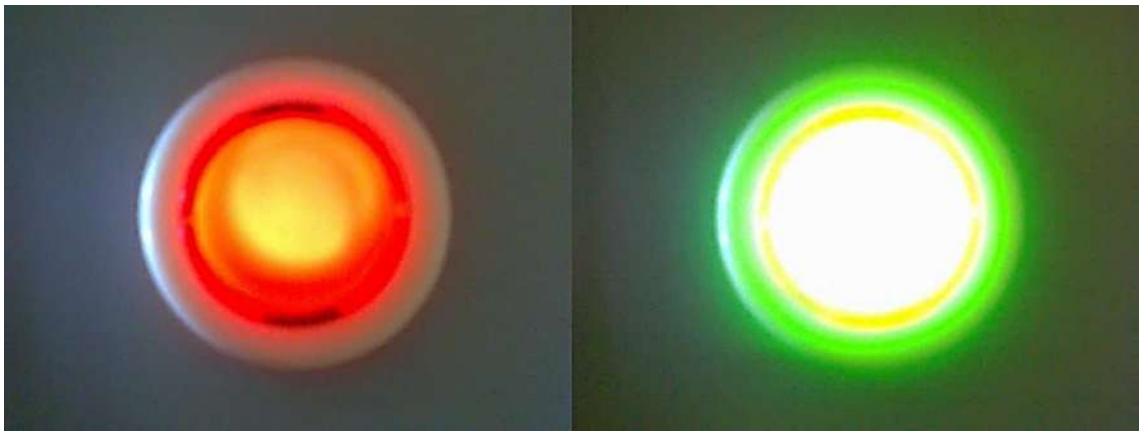


Fig. 3-40. Leds de estado del servicio dentro del botón de presión

3.5.1.5 Impresora térmica

La impresora es el dipositivo con el cual se le entrega la credencial de acceso (boleto) al cliente. El modelo usado en este trabajo es QUIAN ANJET80 QIT801701 y es como la que se muestra en la Fig. 3-41.

Fig. 3-41. Impresora QUIAN ANJET80 QIT801701 [**25**]

El controlador de dicho dispositivo no está disponible para raspberry por lo fué necesario usar un adaptador USB a RJ45 para poder imprimir el boleto, la impresión se manda desde el módulo de acceso inalámbrico a través del protocolo IPP [**26**] a la raspberry y de ella a la impresora. Para que la orden de impresión le llegue a la impresora es necesario configurar el cortafuegos [**27**] [**28**] [**29**] [**30**] de la raspberry para que toda petición hacia la impresora se redireccione a la misma, esto se hace con las reglas firewall del apéndice 8.

Las reglas del apéndice 8 permiten que todo el tráfico que salga del módulo de acceso inalámbrico que tenga como destino la impresora sea redireccionado a ella a través de la raspberry, de esta manera es como se consigue imprimir [**31**] [**32**] [**33**] el boleto en la impresora. El código del apéndice 9 muestra el contenido de un script cuya tarea dedicada es activar la interfaz USB, asignarle los recursos de red y crear la conexión con la impresora al encenderse la raspberry.

El script del apéndice 9 está corriendo continuamente en la raspberry, monitoreando el estado de la interfaz eth0 (USB) ya que se observó que después de un determinado tiempo tiende a cambiar su IP; para controlar está situación es necesario que dicho script esté reconfigurando periódicamente la interfaz a fin de que no se pierda la conexión con la impresora.

3.5.1.6 Selector de monedas o coin selector

El selector de monedas es el dispositivo que se encarga de recibir las monedas y cobrar el costo de la clave de acceso. El dispositivo usado en éste trabajo es como el que se muestra en la Fig. 3-42.

Fig. 3-42. Selector de monedas

Este dispositivo se alimenta con 12 VCD, cuenta con una ranura en donde se deposita la moneda de referencia a recibir, en este caso es una moneda de 5

pesos. Al introducir una moneda el dispositivo envía un impulso eléctrico de 100 milisegundos por su terminal de salida y dicha señal es recibida en la terminal de la raspberry configurada para ésta función. La moneda que recibe se muestra en la Fig. 3-43.

Fig. 3-43. Moneda de 5 pesos mexicanos [34]

3.5.1.7 Diagrama electrónico de interconexiones del prototipo

En la Fig. 3-44 se muestra el diagrama electrónico de las interconexiones entre los pines de la raspberry y las interfaces con las que se comunica con el cliente.

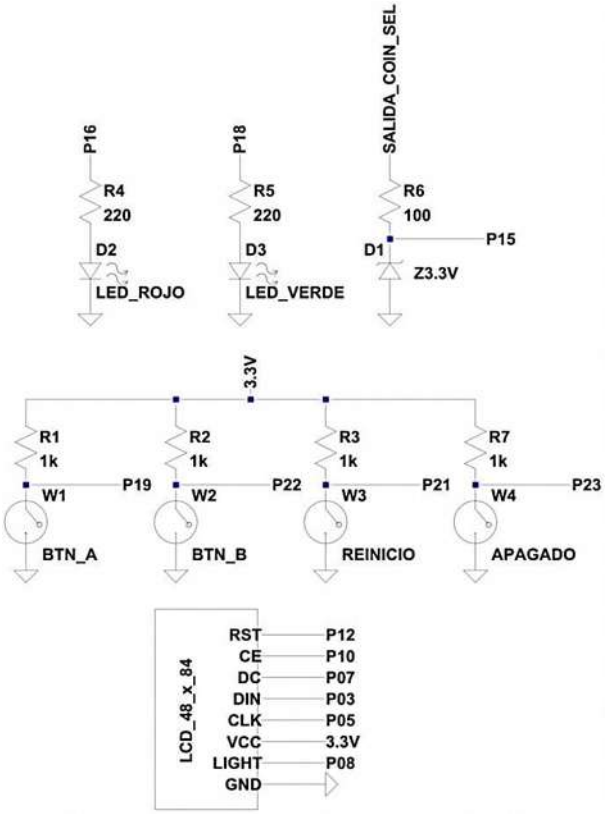


Fig. 3-44. Conexiones de los pines de la raspberry

Finalmente, el prototipo con sus componentes conectados y funcionando puede observarse en las figuras Fig. 3-45, 3-46 y 3-47.

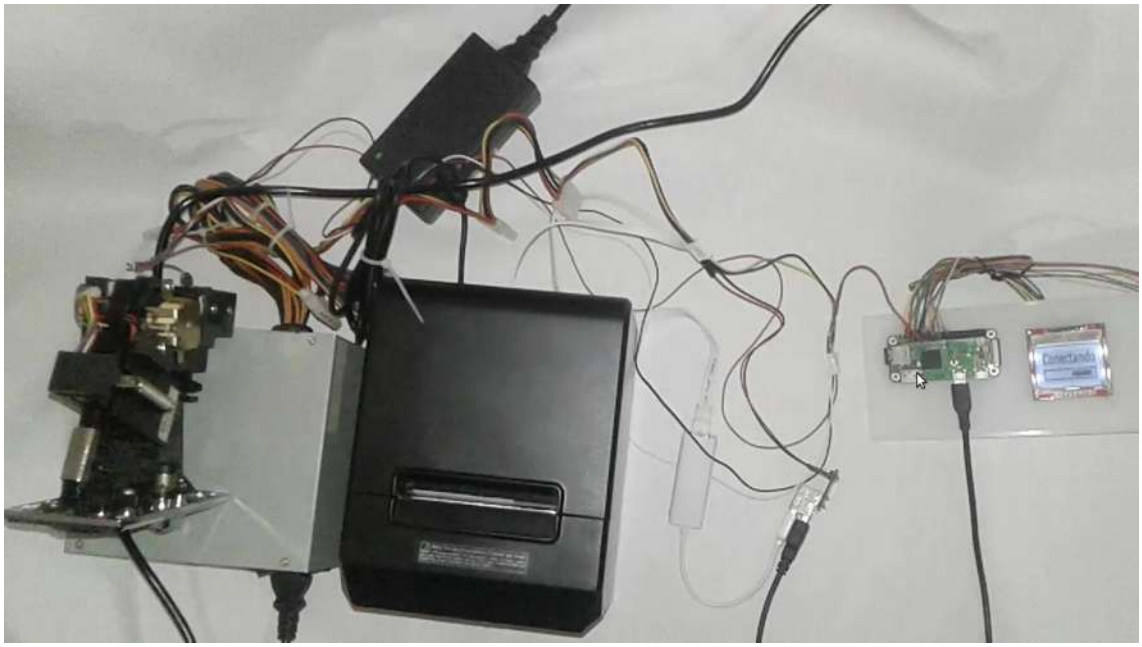


Fig. 3-45. Aspecto general de las piezas interconectadas y funcionando 1.

Fig. 3-46. Aspecto general de las piezas interconectadas y funcionando 2.

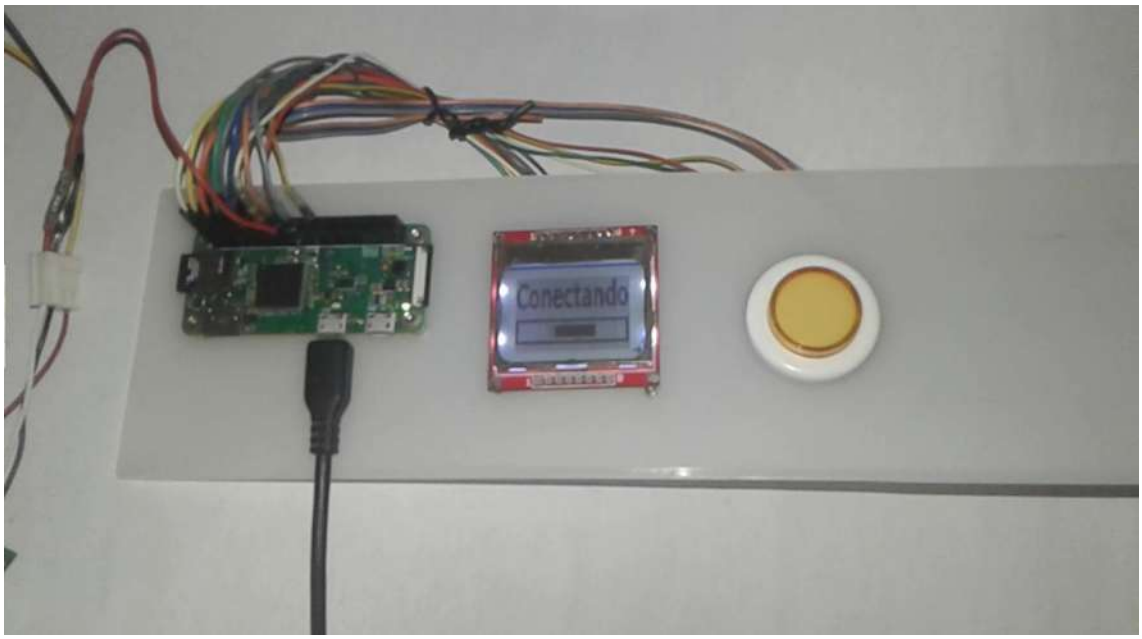


Fig. 3-47. Aspecto general de las piezas interconectadas y funcionando 3.

3.5.2 Del software

3.5.2.1 Configuración del programa de venta

El software empleado para el funcionamiento del módulo de auto-servicio está desarrollado en lenguaje de programación C; tiene el propósito de configurar el hardware como son los pines GPIO de entrada y salida de la raspberry la cual funciona como intermediario entre el cliente y la estación PSII. Dicho software está escrito para trabajar con las páginas php de activaciones de claves o crendenciales de acceso por lo que cualquier cambio en las páginas php de la estación PSII debe también hacerce en el código fuente. El código del software del módulo de auto-servicio se muestra en el anexo 9. La Fig. 3-48 muestra la lógica de funcionamiento general del programa mientras que la Fig. 3-49 describe la lógica seguida al realizar la venta de una clave.

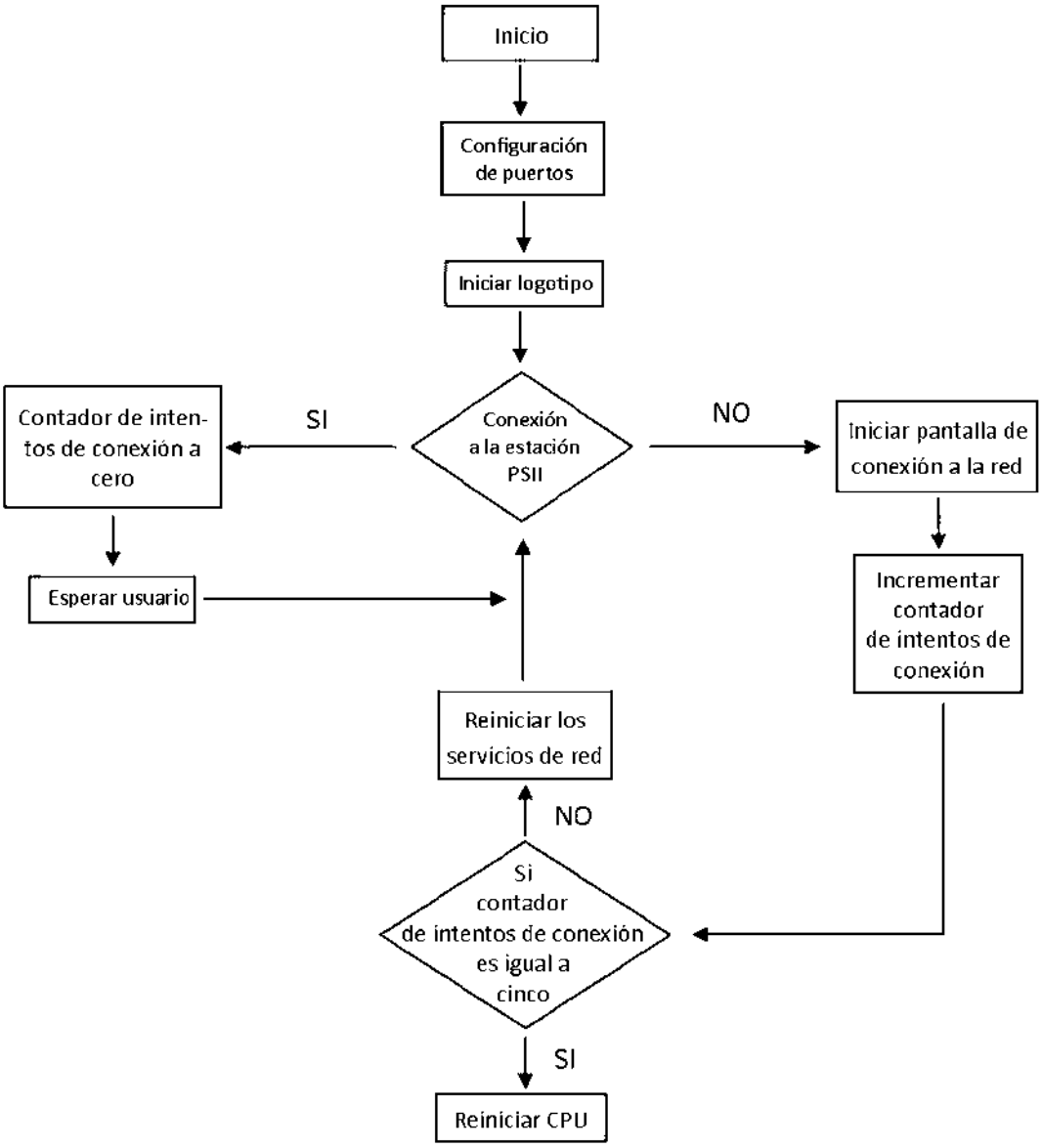


Fig. 3-48. Lógica de funcionamiento del programa general.

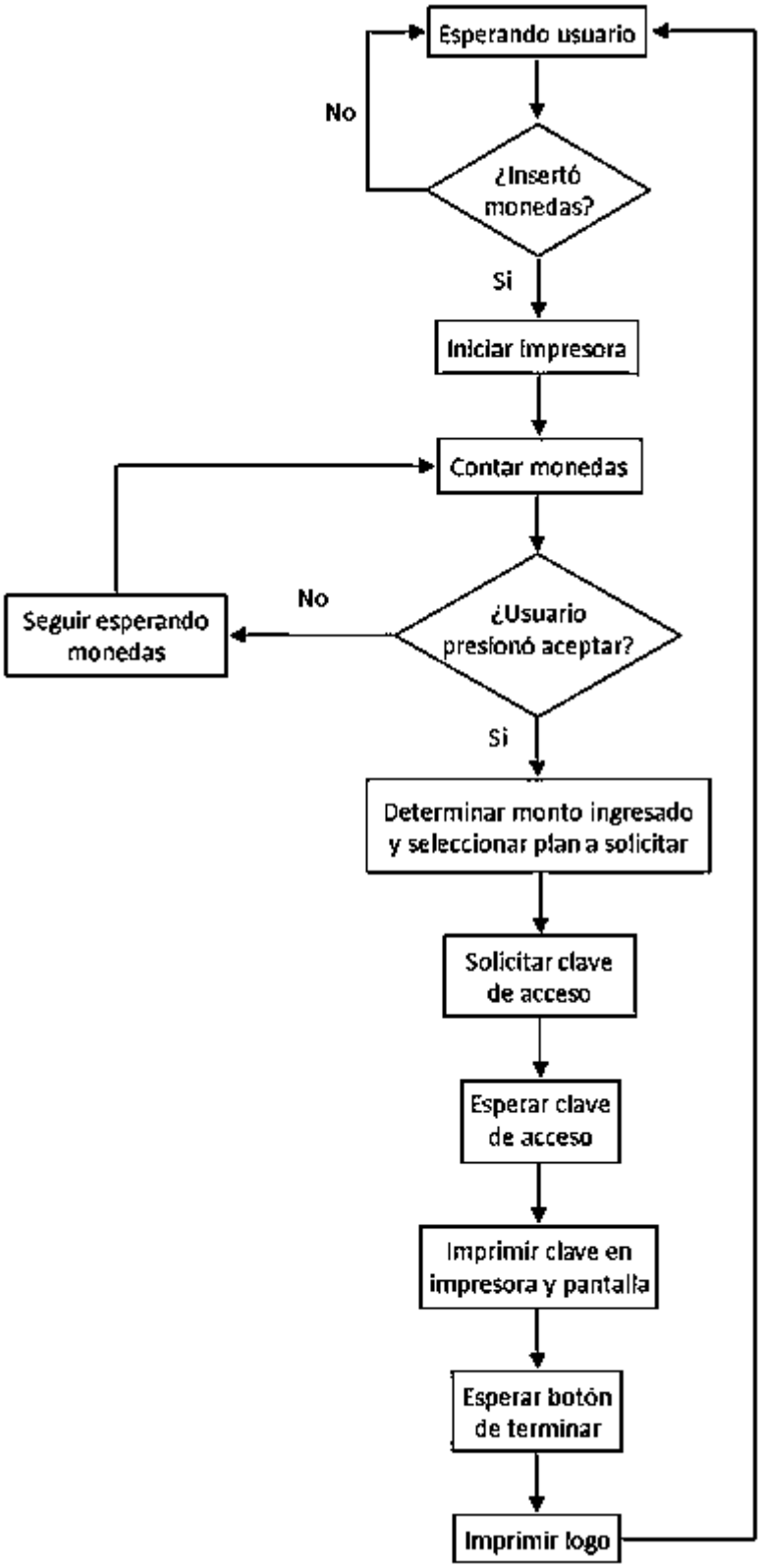


Fig. 3-49. Lógica del programa para activar y cobrar una clave.

El programa se ejecuta automáticamente al arrancar [**35**] [**36**] la raspberry a través de un script llamado arranque.sh, el cual está guardado en el directorio "/root/" y cuya tarea es iniciar el programa así como otros scripts que se necesitan. EL código implementado puede observarse en el apéndice 10. Para activar el script se debe de introducir en la terminal las siguientes órdenes:

```
ln --symbolic /root/arranque.sh /etc/init.d;
cd /etc/init.c/;
update-rc.d arranque.sh defaults;
```

A partir de ahora el programa se inicia cada vez que se enciende la raspberry.

Capítulo 4. Pruebas y Resultados

El presente capítulo recopila los resultados obtenidos al probar la operación del hardware y software de los elementos que componen la estación PSII.

4.1 Del módulo de acceso inalámbrico

4.1.1 El hardware

Las pruebas de hardware realizadas consistieron en la medición de la temperatura interna a una temperatura ambiente alcanzando los 45º centígrados (exposición directa a la luz solar) bajo el efecto del sistema de ventilación y la observación sobre la presencia de agua al interior de la máquina. De las cuales se obtuvieron los siguientes resultados:

4.1.1.1 Del sistema de ventilación

Al no funcionar el sistema de ventilación el calor generado por los componentes se acumula llegando al punto de correr el riesgo de dañarse alguna pieza electrónica del sistema ya que la temperatura es muy alta, capaz de generar la sensación de quemazón en la piel, sin embargo al activarse el ventilador la temperatura baja y se mantiene estable, logrando de esta manera la operación de manera continua bajo el calor que puede llegar en las horas más calurosas del día. La Fig. 4-1 muestra un termómetro de uso médico midiendo la temperatura al interior del módulo de acceso inalámbrico y en la cual se observa que la temperatura está por encima de los 45º que es máximo rango de temperatura que puede leer el termómetro. Cabe aclarar que se usó un termómetro de uso médico debido a que no se pudo contar en el momento las pruebas con un termómetro de laboratorio ya que éste alcanza a medir temperaturas superiores a 45º.

Fig. 4-1. Temperatura mayor a 42º sin ventilación.

4.1.1.2 De la exposición al flujo pluvial o lluvia

Al exponer el gabinete metálico a la lluvia se observa que el agua no entra por ningún lado ya que previamente se selló con sellador hidráulico todos los orificios por donde pudiese entrar el agua.

4.1.2 El software

4.1.2.1 Freeradius

La configuración de freeradius es compleja, sin embargo, para conocer si el servicio puede iniciar después de configurar los archivos del directorio "/etc/freeradius/" es necesario ejecutarlo en modo de prueba. Este paso es fundamental ya que indica la disposición del mismo para procesar solicitudes de autenticación o bien sobre la existencia de algún error en las configuraciones realizadas, por lo que sirve en modo depurador. La depuración se lleva a cabo con la instrucción en la terminal:

```
freeradius -X;
```

Al ejecutar el comando anterior se obtiene el resultado mostrado en la Fig. 4-2.

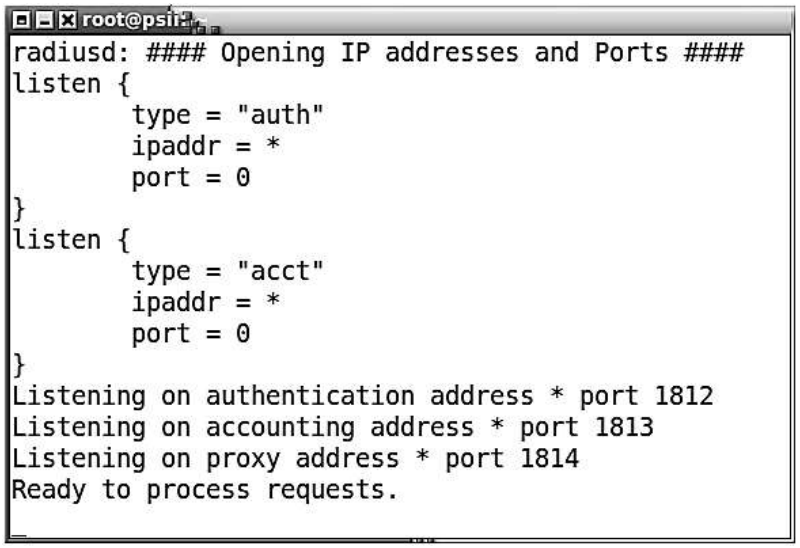


Fig. 4-2. Freeradius preparado para procesar solicitudes de autenticación.

La respuesta obtenida es la disposición del servicio de atender solicitudes de autenticación.

4.1.2.2 Radtest

La prueba radtest consiste en hacer consultas de autenticación a freeradius para determinar si puede comunicarse con la base de datos y leer las credenciales de acceso.

El desarrollo de esta verificación requiere tener una credencial de acceso registrada en la base de datos. Para simplificar esta actividad se tomó como ejemplo la clave "Jadnet0cs1" de la sección 3.4.2.5 y se actualizó su fecha de expiración en la base de datos hasta el 31 de diciembre del 2020. La ejecución de la consulta se realizó con la instrucción en la terminal:

```
radtest Jadnet0cs1 metanet 127.0.0.1 1812 testing123;
```

obteniéndose como resultado el mensaje de paquete aceptado. La Fig. 4-3 se presenta una captura de pantalla de la respuesta obtenida.

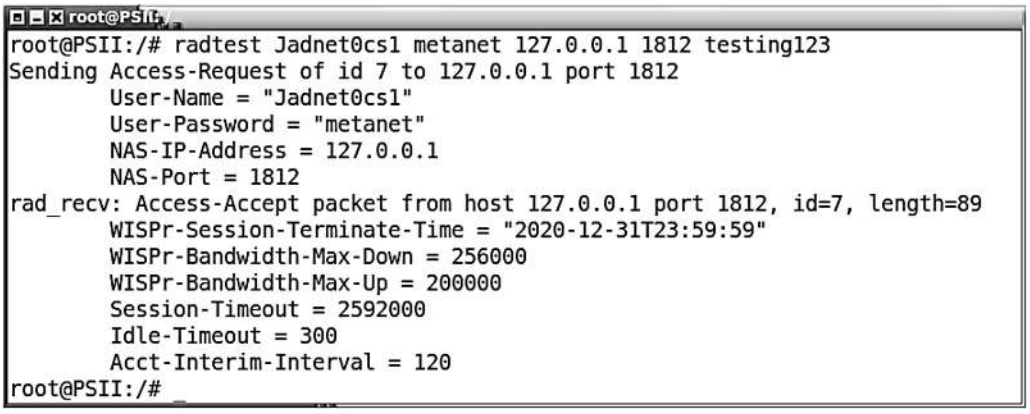


Fig. 4-3. Acceso aceptado para una clave vigente.

Sí la fecha de vencimiento de la clave fuera el 31 de diciembre del 2000 se tendría como resultado la respuesta de la Fig. 4-4:

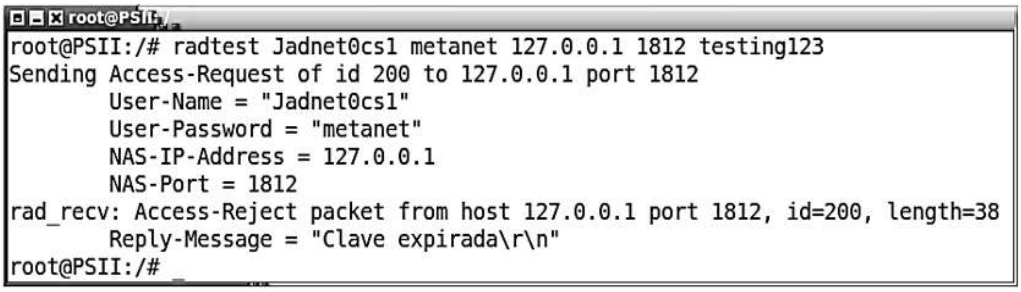


Fig. 4-4. Acceso rechazado para una clave expirada.

Como se puede observar en la Fig. 4-4 la respuesta es un acceso rechazado con el mensaje de: "Clave expirada". Los resultados anteriores significan que freeradius logró conectarse a la base de datos, leer y procesar la credencial de acceso. Finalmente, para una clave de usuario no registrada o bien para una mala configuración en los registros de las credenciales de acceso (lo que implica la imposibilidad de procesar sus valores) la respuesta es un acceso rechazado tal como lo muestra la Fig. 4-5:

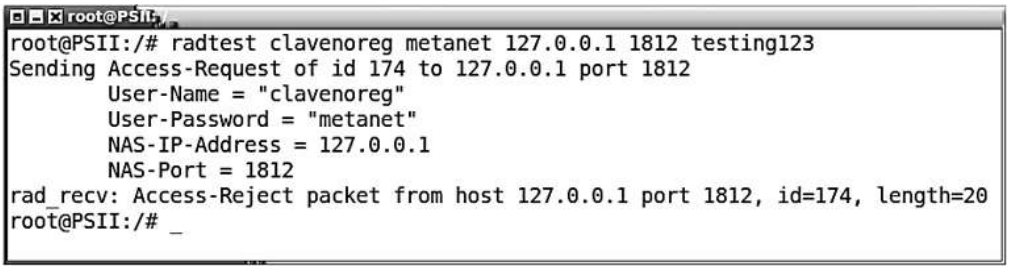


Fig. 4-5. Acceso rechazado para una clave no registrada o una clave mal configurada.

4.1.2.3 El controlador de red o chillispot

Esta comprobación se basa en ejecutar chillispot en modo de prueba o depuración para diagnosticar su disposición de atender y asignar los recursos de red a las solicitudes de conexión de los clientes, mientras que por otro lado sirve para verificar que las configuraciones realizadas en el archivo "/etc/chilli.conf" sean las correctas. La sintáxis para ejecutarlo en modo prueba es:

```
chilli --fg;
```

El resultado obtenido es la aceptación de las conexiones de los clientes desplegando en pantalla la dirección MAC) y la IP asignada al cliente al conectarse. La Fig. 4-6 muestra la captura de pantalla de la terminal de ésta prueba.

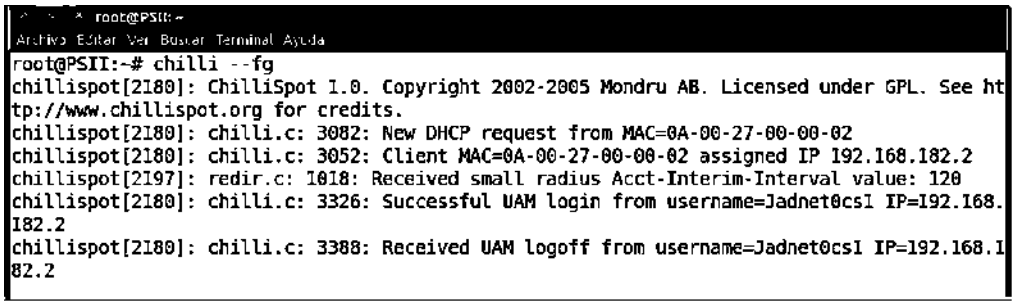


Fig. 4-6. Chillispot atendiendo la solicitud de conexión y autenticación.

4.1.2.4 El servidor web apache

Después de reiniciar el servidor web apache con las configuraciones realizadas en la sección 3.4.2.11, se puede apreciar que los cambios han tomado efecto. Al observar el certificado SSL que usa el servidor web al usar el protocolo https se tiene la respuesta de la Fig. 4-7.

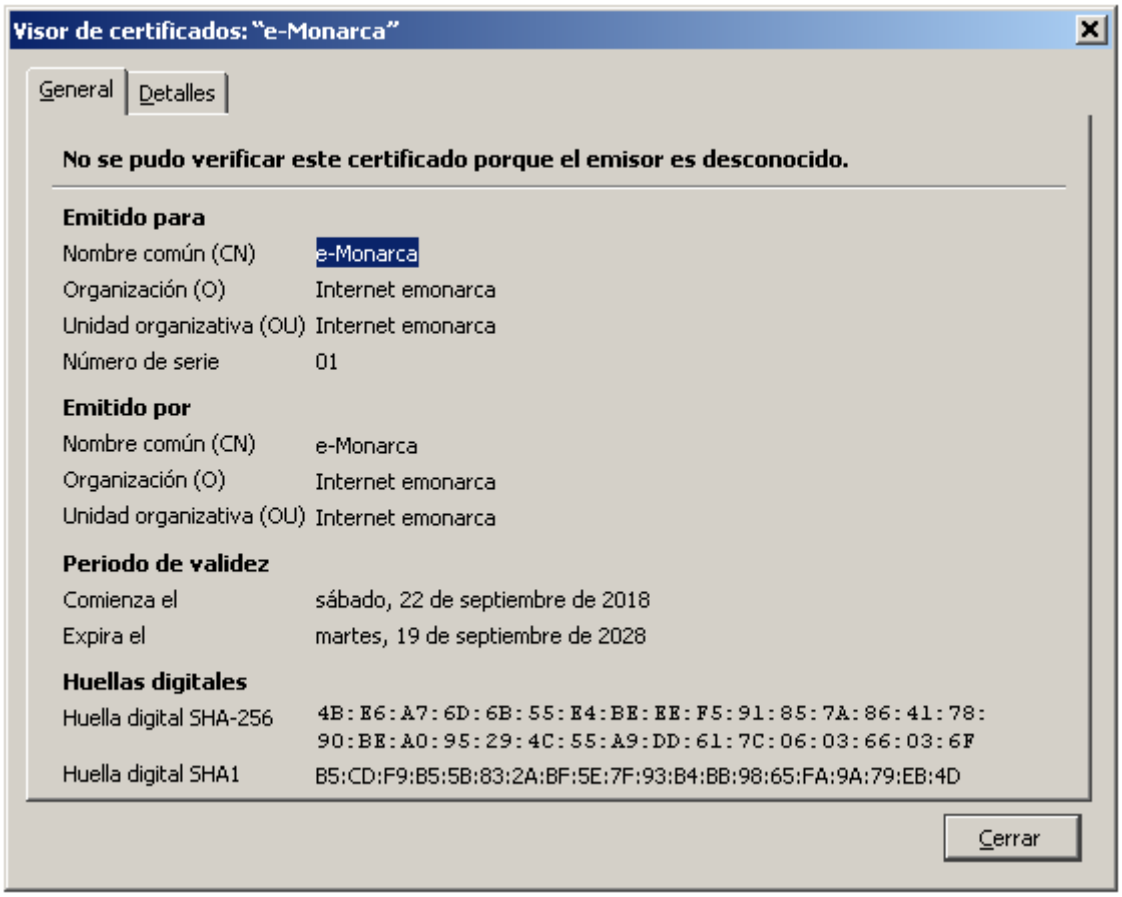


Fig. 4-7. Certificado SSL funcionando en el servidor web

La Fig. 4-7 muestra una captura de pantalla tomada usando el navegador mozilla firefox y en ella pueden apreciarse los detalles del certificado SSL de la estación PSII.

Respecto a la seguridad al folder de ejecución se tienen dos resultados:

El primero consiste en ver la respuesta del servidor cuando se ingresa por medio de un navegador web (en esta prueba se usó google chrome) al folder de ejecución sin reglas de seguridad; el resultado es que se puede acceder y visualizarse todo el contenido tal como la Fig. 4-8 lo demuestra.

El segundo consiste en repetir el procedimiento anterior pero ahora con las reglas de seguridad aplicadas al folder de ejecución, el resultado obtenido es el acceso prohibido al folder. Al visualizar el navegador se obtiene como respuesta la imagen de la Fig. 4-9.

Al tratar de ingresar al folder de ejecución través de wget el resultado también es una denegación o acceso prohibido tal como y en la Fig. 4-10 se demuestra.

Fig. 4-8. Acceso al directorio de ejecución sin reglas de seguridad

Fig. 4-9. Acceso al directorio de ejecución con reglas de seguridad

```
root@PSII:~# wget -r http://127.0.0.1/directorio_de_ejecucion/
--2018-09-25 01:44:17--  http://127.0.0.1/directorio_de_ejecucion/
Conectando a 127.0.0.1:80... conectado.
Petición HTTP enviada, esperando respuesta... 403 Forbidden
2018-09-25 01:44:17 ERROR 403: Forbidden.

root@PSII:~#
```

Fig. 4-10. Acceso con wget al directorio de ejecución con reglas de seguridad

Como se observa en las figuras Fig. 4-8, 4-9 y 4-10 todo el contenido del directorio de ejecución está oculto y solo puede accederse al contenido si se conoce previamente.

4.1.2.5 El portal captivo

El portal captivo funciona si el certificado SSL ha sido instalado correctamente y el módulo SSL se ha activado en las configuraciones del servidor web.
Para determinar si el portal funciona adecuadamente es necesario conectarse a la estación Wi-Fi del módulo de acceso inalámbrico, posteriormente, desde el explorador web del equipo cliente se intenta acceder a algún sitio web, obteniéndose como respuesta el redireccionamiento a la página web de bienvenida. El resultado anterior significa que el portal captivo está funcionando de manera correcta. En la Fig. 4-11 se puede apreciar el re-direccionamiento al portal de inicio o página de bienvenida de un dispositivo cliente android con el que se realizó éste diagnóstico.

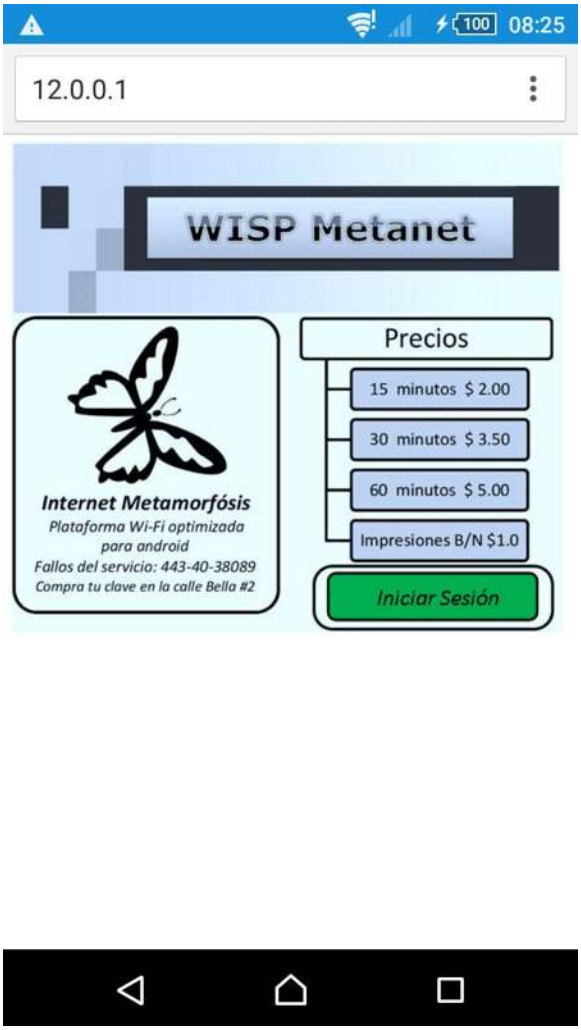


Fig. 4-11. Re-direccionamiento al portal captivo antes de la autenticación.

4.1.2.6 Pruebas de ancho de banda

Los planes de renta tienen configurado un ancho de banda para descarga y otro para subida. En esta sección se muestra el resultado de comprobar las tasas de transferencia configuradas en la sección 3.4.2.3, los cuales tienen como velocidad de descarga 256 kbps y de subida 512 kbps. Para el desarrollo de esta verificación se usa la clave "Jadnet0cs1" que se configuró en la sección 3.4.2.7 y se realizó en un CPU cliente con Ubuntu 10.04 ya que dicha plataforma trae incluído "Monitor del sistema", software de monitoreo de procesos del sistema y puede mostrar en tiempo real la velocidad de transferencia en las interfaces de red. Los resultados obtenidos son los siguientes: En la Fig. 4-12 se observa el resultado de la velocidad de descarga, mientras que en la Fig. 4-13 se muestra el resultado de la velocidad de subida.

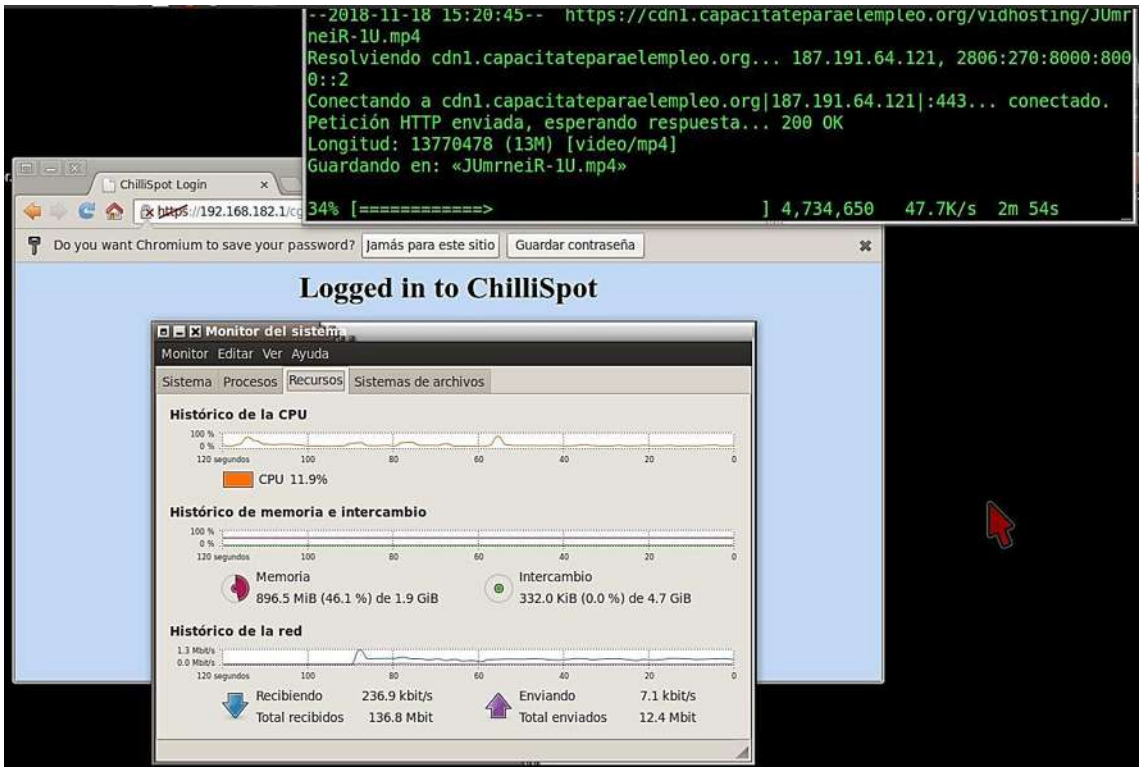


Fig. 4-12. Velocidad de transferencia de descarga

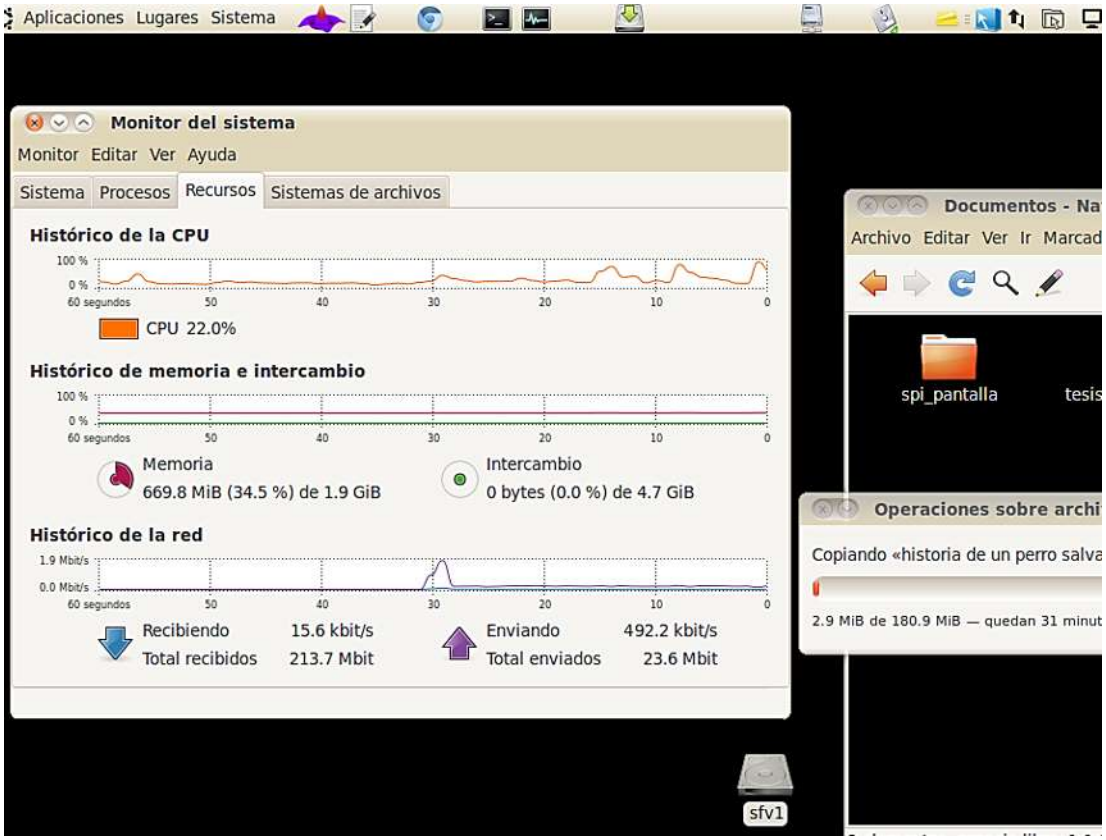


Fig. 4-13. Velocidad de subida a una unidad de almacenamiento en red llamada sfv1.

Como se puede observar en las figuras Fig. 4-12 y 4-13, el ancho de banda de descarga y de subida se mantienen en los límites establecidos por la velocidad establecida en las tablas de configuración del plan. Sin embargo, se observó que a veces se reduce o se aumenta ligeramente pero en promedio se mantiente.

4.2 Resultados del prototipo del módulo de auto-servicio

Esta sección presenta los resultados de las pruebas realizadas al prototipo del módulo de auto-servicio.

4.2.1 Hardware

4.2.1.1 Prueba de la compra de un plan de renta

Esta prueba consistió en comprar el plan de renta de 2 horas, por lo cual fue necesario introducir una moneda de 5 pesos en la alcancía tragamonedas, posteriormente en la pantalla LCD se desplegó la pantalla de bienvenida y enseguida el precio del plan, en las imágenes de abajo se muestra la secuencia de cada una de ellas.

Antes de introducir la moneda se tiene la imagen de la Fig. 4-14, posteriormente, al introducir la moneda de 5 pesos se visualiza la pantalla mostrada por la Fig. 4-15.

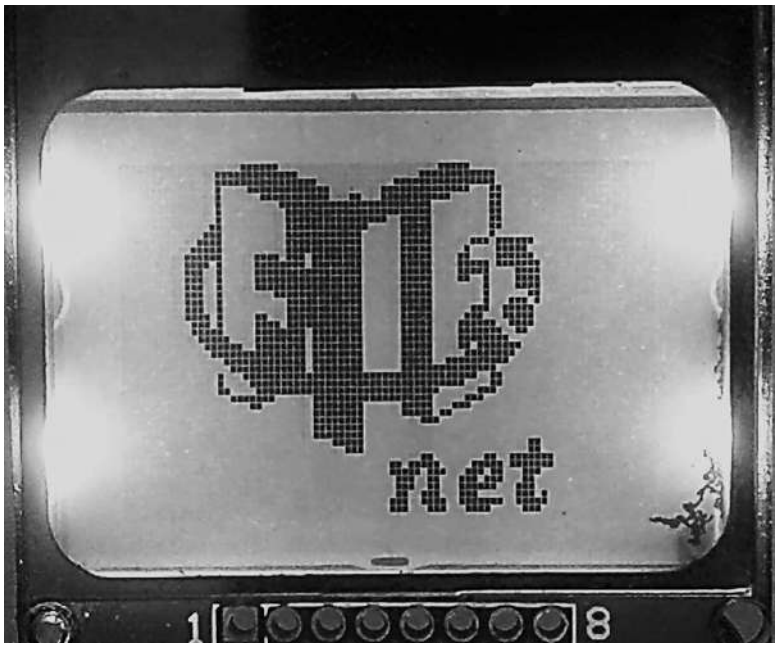


Fig. 4-14. Pantalla de espera

Fig. 4-15. Pantalla de bienvenida

Después de la pantalla de bienvenida cambia a la pantalla de la Fig. 4-16.

Fig. 4-16. Pantalla de selección de compra

Al presionar el botón de compra se tiene la pantalla de espera de activación de clave tal como se observa en la Fig. 4-17.

Fig. 4-17. Pantalla de activación de la clave.

Posteriormente, en la Fig. 4-18 se puede observar la clave de acceso activada y su vigencia.

Fig. 4-18. Pantalla de clave activada

Como parte final del proceso de venta, la clave se imprime en un ticket tal como se puede observar en la Fig. 4-19.

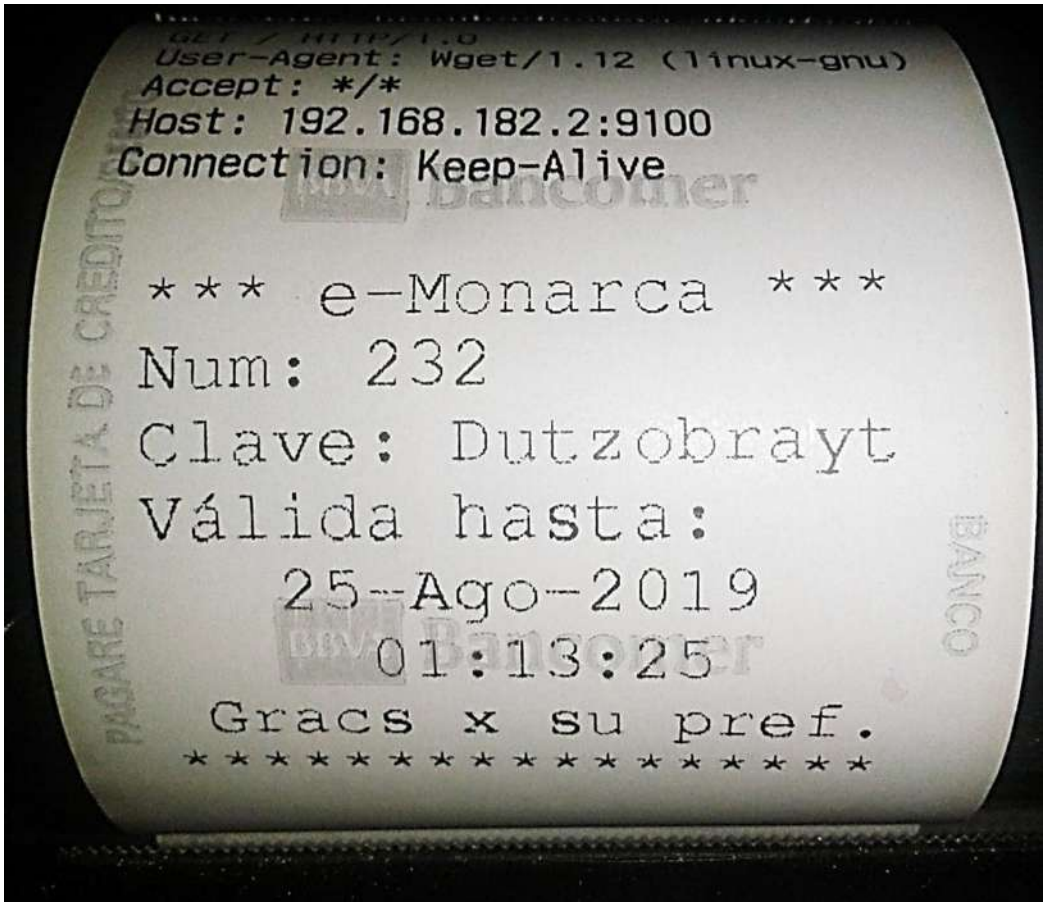


Fig. 4-19. Impresión de la clave activada en papel térmico

Finalmente, para terminar la compra fue necesario presionar el botón de comprar para iniciarse de nuevo el proceso de venta.

Al revisar la base de datos se observa que la clave liberada ya está registrada en la base de datos lista para ser usada. La Fig. 4-20 muestra la clave registrada en la base de datos después de haber sido activada.

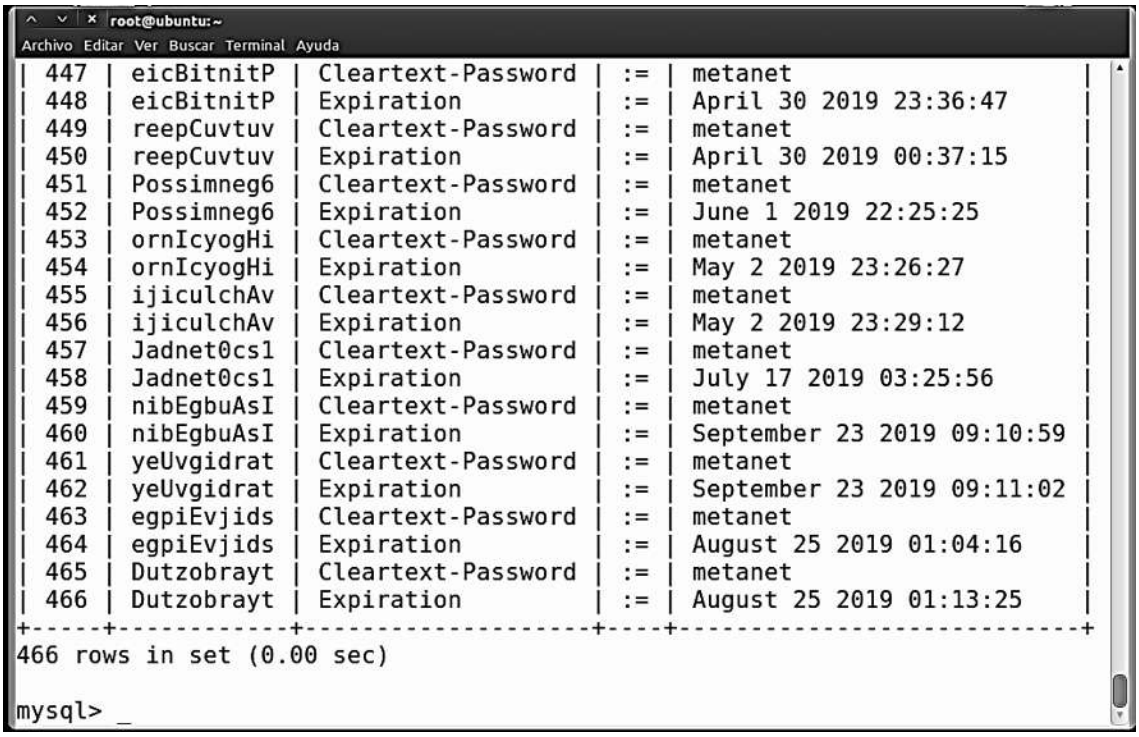


Fig. 4-20. Clave de usuario registrada en la base de datos.

De ésta manera se concluye las pruebas y resultados del prototipo del módulo de auto-servicio.

Capítulo 5. Conclusiones

Este capítulo aborda sobre las ventajas y desventajas detectadas durante las pruebas de la estación PSII, así como las propuestas de mejoramiento que se pueden implementar.

5.1 Ventajas

Hoy en día el mercado de servicio de Internet en México está a la alza. Con la llegada del nuevo gobierno federal (2018-2024) dicho mercado irá en aumento, porque de acuerdo al plan nacional de desarrollo 2018-2024 entre sus propuestas se encuentra dotar la mayoría del territorio nacional con el servicio de Internet [**37**]. Relacionando lo anterior con el prototipo desarrollado e implementado en el presente trabajo, se puede afirmar que es una solución para negocios y pequeñas empresas que deseen comercializar con el servicio de Internet, ya que todas las tareas que implica la administración de un PSII están automatizadas en un solo sistema lo que le da mucha facilidad la instalación y uso del mismo. Sumando a esto los negocios como restaurantes, comedores, cafeterías, etc., podrán mejorar sus prestaciones de servicio, brindando la conectividad a Internet como algo extra, con un estricto control en el tiempo de acceso, evitando de esta manera el abuso o el mal uso del mismo con la ayuda de la administración del prototipo PSII porque con el simple hecho de presionar un botón se puede liberar una credencial de acceso lista para su uso. De manera resumida se tienen las siguientes ventajas:

- Facilidad en la instalación y uso del mismo.

- Libre de interfaces cableadas.

- Compatibilidad con diferentes proveedores de Internet (banda ancha, ethernet, enlances punto a punto, etc).

- Resistencia a factores climáticos ambientales.

- Sistema auto-administrado las 24 horas del día.

- Posibilidad de modificar y/o agregar planes de renta según las necesidades del comerciante o vendedor.

- Control del ancho de banda.

- Capacidad de navegación web en páginas autorizadas sin previa autenticación.

- Redireccionamiento a sitios web personalizados.

5.2 Mejoras

Entre las mejoas que se pueden mencionar en el funcionamiento de la PSII, se encuentran la dificultad en el arreglo de fallos a nivel de software que pudieran presentarse durante el transcurso de su funcionamiento, aunque es raro, no se descarta la posibilidad, ya que las configuraciones son complejas por lo que requieren del conocimiento y dominio sobre el conjunto de programas que hacen posible el funcionamiento de la PSII. Las mejoras más relevantes que se desean arreglar son:

- Dificultad al configurar el software debido a la escasa documentación en Internet.

- A veces se tiene que experimentar a prueba y error hasta conseguir resultados satisfactorios.

- Cambios que se realizen en las páginas php de activaciones de claves también deben de hacerse en el código fuente del módulo de auto-servicio lo que implica de nuevo la compilación.

- Cambios realizados en el script de activación deben de dar como resultado el mismo formato de salida que se manda al módulo de auto-servicio.

- No se admite otro tipo de moneda diferente a la de la Fig. 3-43 debido a que el software del módulo de auto-servicio está hecho para funcionar con este tipo de monedas.

- El sistema operativo (ubuntu server 10.04) usado para hacer funcionar el CPU del módulo de acceso inalámbrico está obsoleto ya que chillispot no está disponibles en sistemas operativos recientes o superiores a la version 16.04.

- El alcance y calidad de la señal inalámbrica puede verse afectada por obstáculos entre el módulo de acceso inalámbrico y el cliente.

- El número de clientes máximos es 20 para evitar la pérdida de la calidad de la señal, ya que entre más clientes exista más ancho de banda se requiere y por otro lado se satura la red debido a que en redes inalámbricas Wi-Fi con arquitectura BBS el control de la comunicación lo tiene el punto de acceso.

- Los cambios de hardware no son recomendados ya que todas las piezas están para funcionar de manera conjunta.

5.3 Trabajos a futuro

Al ser un prototipo de una máquina automatizada presenta ciertas limitantes tales como su excesivo volumen, el alto consumo de energía ya que es una tarjeta madre de computadora la que se utiliza para la el servidor NAS y la dificultad para configurarla, que son características no deseables. En el futuro se tiene la visión de utilizar equipos de bajo consumo tales como una raspberry o módems con sistemas embebidos tales como open-WRT para disminuir el consumo energético y automatizar todo el proceso de configuración en un script o programa.
Entre las características que se desean mejorar y/o agregar se encuentran:

- Automatizar completamente el proceso de instalación.

- Realizar las ventas a través de mensajes de texto de telefonía celular.

- Aceptar diversos tipos de monedas.

- Devolver el cambio (si se requiere) al cliente cuando haya adquirido un plan.

- Controlar remotamente la estación PSII através del sistema de telefonía celular.

- Usar componentes de bajo consumo energético.

- Usar paneles fotovoltaicos cuando se pueda.

- Implementar un sistema de rastreo GPS a la PSII como medida de prevención anti-robos.

5.4 Costos aproximados de los materiales y equipo

La inversión aproximada para crear el prototipo se estima en $5000.00 para cada uno de los módulos, ya que en ocasiones se tuvo que comprar herramientas, materiales y accesorios para el acondicionamiento de los gabinetes; otras veces fábricar piezas necesarias en la construcción de los módulos. Sin embargo, se tiene la visión a corto plazo de que el prototipo aquí desarrollado se ponga en operación en los alrededores de la ciudad y lugares en donde la infraestrutura cableada no llega pero exista la manera de hacer llegar la señal de Internet por medios inalámbricos tales como el enlace punto a punto, la telefonía celular o el enlace satelital; puesto que el prototipo admite conectividad al distribuidor PSI (haciendo algunos ligeros ajutes) con diferentes tipos de hardware tales como la interfaz USB, cable de ethernet o Wi-Fi.

Apéndices y anexos

Apéndice 1. Materiales empleados para la construcción del módulo de acceso inalámbrico

- Señal de internet proveniente de algún proveedor (de preferencia cableada para soportar el ancho de banda necesario en contraste con la inalámbrica).
- Un CPU el cual funcionará como servidor NAS, controlador de red, servidor de alojamiento WEB o webhosting y portal captivo.
- Dos tarjetas de red, al menos una de Ethernet con velocidad de transferencia de 100 Mbps o superior.
- Un conmutador o swicth el cual enrutará todo el tráfico proveniente de los módems Wi-Fi que estarán funcionando como puntos de acceso o access point al puerto de Ethernet del CPU.
- Uno o más módems Wi-Fi configurados como punto de acceso o access point.
- Imagen ISO o CD-ROM de Ubuntu 10.04 server.
- 1 gabinete de metal para uso exterior de 30 x 30 x 20 cms.
- Maya mosquitera de metal como filtro del sistema de ventilación.
- 2 capacitores de 47µF a 35V
- 1 diodo zener de 3.3V
- 1 diodo 1N4148
- 1 diodo LED 5 mm verde
- 1 diodo LED 5 mm amarillo
- 1 diodo LED 5 mm rojo
- 1 circuito integrado LM324
- 2 reguladores lineales LM7812
- 6 transistores BJT 2N2222
- 1 resistencia de 100 Ω de ¼ de watt
- 8 resistencias de 1 kΩ de ¼ de watt
- 1 resistencia de 2.6 kΩ de ¼ de watt
- 5 resistencias de 10 kΩ de ¼ de watt
- 1 resistencia de 100 kΩ de ¼ de watt
- 1 resistencia de 1 MΩ de ¼ de watt
- Tornillos de 1/8" x 1/2"
- Tornillos de 1/8" x 1"
- Tornillos de 1/8" x 2"
- Tuercas para tornillos de 1/8"
- Rondanas para tornillos de 1/8"
- 1 relevador de 12 VCD con capacidad de 10 amperios a 125 VCA
- 1 placa fenólica de 10.5 x 8.5 cm
- 2 placas fenólicas de 10 x 5 cm
- 1 ventilador para CPU de 3" de 12 VCD

- 2 metros de cable de dos polos del #12
- 1 interruptor con lámpara de neón de 10 amperios a 125 VCA
- 1 interruptor de presión (push botón)
- 4 coples hembra tipo F
- 4 conectores macho tipo F para cable coaxial RG6
- 2 metros de cable coaxial RG58A
- Cables para conexiones
- 1 metro de cable de red UTP.
- Conectores macho RJ45.

Apéndice 2. Materiales utilizados para la construcción del módulo de auto-servicio

- Una tarjeta Raspberry Pi Zero W
- Archivo imagen del sistema operativo Raspbian Stretch with desktop and recommended software (disponible en: http://director.downloads.raspberrypi.org/raspbian_full/images/raspbian_full-2019-04-09/2019-04-08-raspbian-stretch-full.zip)
- Un selector de monedas o coin selector
- Una impresora térmica
- Un adaptador USB a RJ45
- Tornillos de 1/8" x 1/2"
- Tornillos de 1/8" x 1"
- Tornillos de 1/8" x 2"
- Tuercas para tornillos de 1/8"
- Rondanas para tornillos de 1/8"
- 1 interruptor de presión (push botón)
- Una pantalla LCD-10168
- Un diodo LED RGB de 3.5 mm de diámetro
- Cuatro resistencias de 1 kΩ de ¼ de watt
- Dos resistencias de 220 Ω de ¼ de watt
- Una resistencia de 100 Ω de ¼ de watt
- Un diodo zener de 3.3V
- Cables para conexiones

Apéndice 3. Archivos modificados de freeradius

a) El archivo radiusd.conf

```
prefix = /usr
exec_prefix = /usr
sysconfdir = /etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct
name = freeradius
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/${name}
db_dir = ${raddbdir}
libdir = /usr/lib/freeradius
pidfile = ${run_dir}/${name}.pid
user = freerad
group = freerad
max_request_time = 30
cleanup_delay = 5
max_requests = 1024
listen {
        type = auth
        ipaddr = *
        port = 0
}
listen {
        ipaddr = *
#       ipv6addr = ::
        port = 0
        type = acct
#       interface = eth0
#       clients = per_socket_clients
}
hostname_lookups = no
allow_core_dumps = no
regular_expressions     = yes
extended_expressions    = yes
log {
        destination = files
        file = ${logdir}/radius.log
        syslog_facility = daemon
        stripped_names = no
        auth = no
        auth_badpass = no
        auth_goodpass = no
}
checkrad = ${sbindir}/checkrad
security {
        max_attributes = 200
```

```
        reject_delay = 1
        status_server = yes
}
proxy_requests  = yes
$INCLUDE proxy.conf
$INCLUDE clients.conf
thread pool {
        start_servers = 5
        max_servers = 32
        min_spare_servers = 3
        max_spare_servers = 10
        max_requests_per_server = 0
}
modules {
        $INCLUDE ${confdir}/modules/
        $INCLUDE eap.conf
        $INCLUDE sql.conf
}
instantiate {
        exec
        expr
        expiration
        logintime
}
$INCLUDE policy.conf
$INCLUDE sites-enabled/
```

b) El archivo clients.conf

```
client localhost {
        ipaddr = 127.0.0.1
        secret              = testing123 #Contraseña compartida entre
                        # freeradius y el controlador de red o chillispot
        require_message_authenticator = no
        nastype     = other         # localhost isn't usually a NAS...
}
client 12.0.0.0/24 {
        secret    = contraseniaradius
}
```

c) El archivo eap.conf

```
eap {
        default_eap_type = md5
        timer_expire     = 60
        ignore_unknown_eap_types = no
        cisco_accounting_username_bug = no
        max_sessions = 4096
        md5 {
        }
        leap {
        }
```

```
        gtc {
                auth_type = PAP
        }
        tls {
                certdir = ${confdir}/certs
                cadir = ${confdir}/certs
                private_key_password = 3otAcOwbOjyetyeycTonshyoobzamfem
#frase con la que se encripta
                private_key_file = ${certdir}/serverkey.key
#La clave del certificado de cliente
                certificate_file = ${certdir}/certificado_psii.cert
#Este es el certificado de cliente
                CA_file = ${cadir}/cacert.cert
#Este es el certificado de autoridad
                dh_file = ${certdir}/dh #Sin cambios
                random_file = ${certdir}/random #Sin cambios
                cipher_list = "DEFAULT"
                make_cert_command = "${certdir}/bootstrap"
                cache {
                    enable = no
                    lifetime = 24 # hours
                    max_entries = 255
                }
        }
        ttls {
                default_eap_type = md5
                copy_request_to_tunnel = no
                use_tunneled_reply = no
                virtual_server = "inner-tunnel"
        }
        peap {
                default_eap_type = mschapv2
                copy_request_to_tunnel = no
                use_tunneled_reply = no
                virtual_server = "inner-tunnel"
        }
        mschapv2 {
        }
}
```

d) El archivo sql.conf

```
sql {
        database = "mysql"
        driver = "rlm_sql_${database}"
        server = "localhost"
        #port = 3306
        login = "root"
        password = "toor"
        radius_db = "radius"
        acct_table1 = "radacct"
        acct_table2 = "radacct"
```

```
        postauth_table = "radpostauth"
        authcheck_table = "radcheck"
        authreply_table = "radreply"
        groupcheck_table = "radgroupcheck"
        groupreply_table = "radgroupreply"
        usergroup_table = "radusergroup"
        deletestalesessions = yes
        sqltrace = no
        sqltracefile = ${logdir}/sqltrace.sql
        num_sql_socks = 5
        connect_failure_retry_delay = 60
        lifetime = 0
        max_queries = 0
        nas_table = "nas"
        $INCLUDE sql/${database}/dialup.conf
}
```

e) El archivo users (solo para pruebas)

```
#steve  Cleartext-Password := "testing"
#       Service-Type = Framed-User,
#       Framed-Protocol = PPP,
#       Framed-IP-Address = 172.16.3.33,
#       Framed-IP-Netmask = 255.255.255.0,
#       Framed-Routing = Broadcast-Listen,
#       Framed-Filter-Id = "std.ppp",
#       Framed-MTU = 1500,
#       Framed-Compression = Van-Jacobsen-TCP-IP
usuario Cleartext-Password := "abcd12340ff"
```

Apéndice 4. Configuración del archivo "/etc/apache2/default-ssl"

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog /var/log/apache2/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog /var/log/apache2/ssl_access.log combined

        Alias /doc/ "/usr/share/doc/"
        <Directory "/usr/share/doc/">
                Options Indexes MultiViews FollowSymLinks
                AllowOverride None
                Order deny,allow
                Deny from all
                Allow from 127.0.0.0/255.0.0.0 ::1/128
        </Directory>

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
```

```
#   SSLCertificateFile directive is needed.
SSLCertificateFile    /etc/freeradius/certs/certificado_wisp.cert
SSLCertificateKeyFile /etc/freeradius/certs/serverkey.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
SSLCACertificatePath /etc/freeradius/certs/
SSLCACertificateFile /etc/freeradius/certs/cacert.cert

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   Access Control:
#   With SSLRequire you can do per-directory access control based
#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#         and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#         and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#         and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
```

```
#          and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20      ) \
#         or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.  The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
#     exportation for CGI and SSI requests only.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#     under a "Satisfy any" situation, i.e. when it applies access is denied
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is send or allowed to received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
```

```
        #    SSL close notify alert is send and mod_ssl waits for the close notify
        #    alert of the client. This is 100% SSL/TLS standard compliant, but in
        #    practice often causes hanging connections with brain-dead browsers. Use
        #    this only for browsers where you know that their SSL implementation
        #    works correctly.
        #   Notice: Most problems of broken clients are also related to the HTTP
        #   keep-alive facility, so you usually additionally want to disable
        #   keep-alive for those clients, too. Use variable "nokeepalive" for this.
        #   Similarly, one has to force some clients to use HTTP/1.0 to workaround
        #   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
        #   "force-response-1.0" for this.
        BrowserMatch "MSIE [2-6]" \
                nokeepalive ssl-unclean-shutdown \
                downgrade-1.0 force-response-1.0
        # MSIE 7 and newer should be able to use keepalive
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
</IfModule>
```

Apéndice 5. Configuración del archivo "/etc/apache2/default"

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog /var/log/apache2/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>

    Alias /directorio_ejecucion/ "directorio_ejecucion/"
    <Directory "directorio_ejecucion/">
        Options -Indexes -MultiViews -FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 192.168.56.0/255.255.255.0 192.168.182.0/255.255.255.0
    </Directory>
</VirtualHost>
```

Apéndice 6. Código del portal captivo

```
#!/usr/bin/perl

# chilli - ChilliSpot.org. A Wireless LAN Access Point Controller
# Copyright (C) 2003, 2004 Mondru AB.
#
# The contents of this file may be used under the terms of the GNU
# General Public License Version 2, provided that the above copyright
# notice and this permission notice is included in all copies or
# substantial portions of the software.

# Redirects from ChilliSpot daemon:
#
# Redirection when not yet or already authenticated
#   notyet:  ChilliSpot daemon redirects to login page.
#   already: ChilliSpot daemon redirects to success status page.
#
# Response to login:
#   already: Attempt to login when already logged in.
#   failed:  Login failed
#   success: Login succeded
#
# logoff:  Response to a logout


# Shared secret used to encrypt challenge with. Prevents dictionary attacks.
# You should change this to your own shared secret.
$uamsecret = "ht2eb8ej6s4et3rg1ulp";

# Uncomment the following line if you want to use ordinary user-password
# for radius authentication. Must be used together with $uamsecret.
#$userpassword=1;

# Our own path
$loginpath = "/cgi-bin/hotspotlogin.cgi";

use Digest::MD5  qw(md5 md5_hex md5_base64);

# Make sure that the form parameters are clean
$OK_CHARS='-a-zA-Z0-9_.@&=%!';
$_ = $input = <STDIN>;
s/[^$OK_CHARS]/_/go;
$input = $_;

# Make sure that the get query parameters are clean
$OK_CHARS='-a-zA-Z0-9_.@&=%!';
$_ = $query=$ENV{QUERY_STRING};
s/[^$OK_CHARS]/_/go;
$query = $_;
```

```
# If she did not use https tell her that it was wrong.
if (!($ENV{HTTPS} =~ /^on$/)) {
   print "Content-type: text/html\n\n
<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\">
<html>
<head>
  <title>ChilliSpot Login Failed</title>
  <meta http-equiv=\"Cache-control\" content=\"no-cache\">
  <meta http-equiv=\"Pragma\" content=\"no-cache\">
</head>
<body bgColor = '#c0d8f4'>
  <h1 style=\"text-align: center;\">ChilliSpot Login Failed</h1>
  <center>
    Login must use encrypted connection.
  </center>
</body>
<!--
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<WISPAccessGatewayParam
  xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"

xsi:noNamespaceSchemaLocation=\"http://www.acmewisp.com/WISPAccessGatewayParam.x
sd\">
<AuthenticationReply>
<MessageType>120</MessageType>
<ResponseCode>102</ResponseCode>
<ReplyMessage>Login must use encrypted connection</ReplyMessage>
</AuthenticationReply>
</WISPAccessGatewayParam>
-->
</html>
";
   exit(0);
}


#Read form parameters which we care about
@array = split('&',$input);
foreach $var ( @array )
{
   @array2 = split('=',$var);
   if ($array2[0] =~ /^UserName$/) { $username = $array2[1]; }
   if ($array2[0] =~ /^Password$/) { $password = $array2[1]; }
   if ($array2[0] =~ /^challenge$/) { $challenge = $array2[1]; }
   if ($array2[0] =~ /^button$/) { $button = $array2[1]; }
   if ($array2[0] =~ /^logout$/) { $logout = $array2[1]; }
   if ($array2[0] =~ /^prelogin$/) { $prelogin = $array2[1]; }
   if ($array2[0] =~ /^res$/) { $res = $array2[1]; }
   if ($array2[0] =~ /^uamip$/) { $uamip = $array2[1]; }
   if ($array2[0] =~ /^uamport$/) { $uamport = $array2[1]; }
   if ($array2[0] =~ /^userurl$/)   { $userurl = $array2[1]; }
```

```
   if ($array2[0] =~ /^timeleft$/)  { $timeleft = $array2[1]; }
   if ($array2[0] =~ /^redirurl$/)  { $redirurl = $array2[1]; }
}

#Read query parameters which we care about
@array = split('&',$query);
foreach $var ( @array )
{
   @array2 = split('=',$var);
   if ($array2[0] =~ /^res$/)      { $res = $array2[1]; }
   if ($array2[0] =~ /^challenge$/) { $challenge = $array2[1]; }
   if ($array2[0] =~ /^uamip$/)    { $uamip = $array2[1]; }
   if ($array2[0] =~ /^uamport$/)  { $uamport = $array2[1]; }
   if ($array2[0] =~ /^reply$/)    { $reply = $array2[1]; }
   if ($array2[0] =~ /^userurl$/)  { $userurl = $array2[1]; }
   if ($array2[0] =~ /^timeleft$/)  { $timeleft = $array2[1]; }
   if ($array2[0] =~ /^redirurl$/)  { $redirurl = $array2[1]; }
}


$reply =~ s/\+/ /g;
$reply =~s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/seg;

$userurldecode = $userurl;
$userurldecode =~ s/\+/ /g;
$userurldecode =~s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/seg;

$redirurldecode = $redirurl;
$redirurldecode =~ s/\+/ /g;
$redirurldecode =~s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/seg;

$password =~ s/\+/ /g;
$password =~s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/seg;

# If attempt to login
if ($button =~ /^Login$/) {
   $hexchal  = pack "H32", $challenge;
   if (defined $uamsecret) {
         $newchal  = md5($hexchal, $uamsecret);
   }
   else {
         $newchal  = $hexchal;
   }
   $response = md5_hex("\0", $password, $newchal);
   $pappassword = unpack "H32", ($password ^ $newchal);
#sleep 5;
print "Content-type: text/html\n\n";
print "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\">
<html>
<head>
  <title>ChilliSpot Login</title>
  <meta http-equiv=\"Cache-control\" content=\"no-cache\">
```

```
  <meta http-equiv=\"Pragma\" content=\"no-cache\">";
   if ((defined $uamsecret) && defined($userpassword)) {
          print "  <meta http-equiv=\"refresh\"
content=\"0;url=http://$uamip:$uamport/logon?username=$username&password=$pappass
word\">";
   } else {
          print "  <meta http-equiv=\"refresh\"
content=\"0;url=http://$uamip:$uamport/logon?username=$username&response=$response
&userurl=$userurl\">";
   }
print "</head>
<body bgColor = '#c0d8f4'>";
  print "<h1 style=\"text-align: center;\">Logging in to ChilliSpot</h1>";
  print "
  <center>
   Please wait.....
  </center>
</body>
<!--
<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<WISPAccessGatewayParam
  xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"

xsi:noNamespaceSchemaLocation=\"http://www.acmewisp.com/WISPAccessGatewayParam.x
sd\">
<AuthenticationReply>
<MessageType>120</MessageType>
<ResponseCode>201</ResponseCode>
";
   if ((defined $uamsecret) && defined($userpassword)) {
          print
"<LoginResultsURL>http://$uamip:$uamport/logon?username=$username&password=$pappa
ssword</LoginResultsURL>";
   } else {
          print
"<LoginResultsURL>http://$uamip:$uamport/logon?username=$username&response=$respo
nse&userurl=$userurl</LoginResultsURL>";
   }
print "</AuthenticationReply>
</WISPAccessGatewayParam>
-->
</html>
";
   exit(0);
}


# Default: It was not a form request
$result = 0;

# If login successful
if ($res =~ /^success$/) {
```

```
   $result = 1;
}

# If login failed
if ($res =~ /^failed$/) {
   $result = 2;
}

# If logout successful
if ($res =~ /^logoff$/) {
   $result = 3;
}

# If tried to login while already logged in
if ($res =~ /^already$/) {
   $result = 4;
}

# If not logged in yet
if ($res =~ /^notyet$/) {
   $result = 5;
}

# If login from smart client
if ($res =~ /^smartclient$/) {
   $result = 6;
}

# If requested a logging in pop up window
if ($res =~ /^popup1$/) {
   $result = 11;
}

# If requested a success pop up window
if ($res =~ /^popup2$/) {
   $result = 12;
}

# If requested a logout pop up window
if ($res =~ /^popup3$/) {
   $result = 13;
}


# Otherwise it was not a form request
# Send out an error message
if ($result == 0) {
   print "Content-type: text/html\n\n
<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\">
<html>
<head>
  <title>ChilliSpot Login Failed</title>
```

```
  <meta http-equiv=\"Cache-control\" content=\"no-cache\">
  <meta http-equiv=\"Pragma\" content=\"no-cache\">
</head>
<body bgColor = '#c0d8f4'>
  <h1 style=\"text-align: center;\">ChilliSpot Login Failed</h1>
  <center>
    Login must be performed through ChilliSpot daemon.
  </center>
</body>
</html>
";
    exit(0);
}

#Generate the output
print "Content-type: text/html\n\n
<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\">
<html>
<head>
  <title>ChilliSpot Login</title>
  <meta http-equiv=\"Cache-control\" content=\"no-cache\">
  <meta http-equiv=\"Pragma\" content=\"no-cache\">
  <SCRIPT LANGUAGE=\"JavaScript\">
    var blur = 0;
    var starttime = new Date();
    var startclock = starttime.getTime();
    var mytimeleft = 0;

    function doTime() {
      window.setTimeout( \"doTime()\", 1000 );
      t = new Date();
      time = Math.round((t.getTime() - starttime.getTime())/1000);
      if (mytimeleft) {
        time = mytimeleft - time;
        if (time <= 0) {
          window.location = \"$loginpath?res=popup3&uamip=$uamip&uamport=$uamport\";
        }
      }
      if (time < 0) time = 0;
      hours = (time - (time % 3600)) / 3600;
      time = time - (hours * 3600);
      mins = (time - (time % 60)) / 60;
      secs = time - (mins * 60);
      if (hours < 10) hours = \"0\" + hours;
      if (mins < 10) mins = \"0\" + mins;
      if (secs < 10) secs = \"0\" + secs;
      title = \"Online time: \" + hours + \":\" + mins + \":\" + secs;
      if (mytimeleft) {
        title = \"Remaining time: \" + hours + \":\" + mins + \":\" + secs;
      }
      if(document.all || document.getElementById){
        document.title = title;
```

```
    }
    else {
      self.status = title;
    }
  }

  function popUp(URL) {
    if (self.name != \"chillispot_popup\") {
      chillispot_popup = window.open(URL, 'chillispot_popup',
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=0,width=500,height=375'
);
    }
  }

  function doOnLoad(result, URL, userurl, redirurl, timeleft) {
    if (timeleft) {
      mytimeleft = timeleft;
    }
    if ((result == 1) && (self.name == \"chillispot_popup\")) {
      doTime();
    }
    if ((result == 1) && (self.name != \"chillispot_popup\")) {
      chillispot_popup = window.open(URL, 'chillispot_popup',
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=0,width=500,height=375'
);
    }
    if ((result == 2) || result == 5) {
      document.form1.UserName.focus()
    }
    if ((result == 2) && (self.name != \"chillispot_popup\")) {
      chillispot_popup = window.open('', 'chillispot_popup',
'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=0,width=400,height=200'
);
      chillispot_popup.close();
    }
    if ((result == 12) && (self.name == \"chillispot_popup\")) {
      doTime();
      if (redirurl) {
        opener.location = redirurl;
      }
      else if (opener.home) {
        opener.home();
      }
      else {
        opener.location = \"about:home\";
      }
      self.focus();
      blur = 0;
    }
    if ((result == 13) && (self.name == \"chillispot_popup\")) {
      self.focus();
      blur = 1;
```

```
    }
   }

   function doOnBlur(result) {
    if ((result == 12) && (self.name == \"chillispot_popup\")) {
     if (blur == 0) {
      blur = 1;
      self.focus();
     }
    }
   }
 </script>
</head>
<body onLoad=\"javascript:doOnLoad($result,
'$loginpath?res=popup2&uamip=$uamip&uamport=$uamport&userurl=$userurl&redirurl=$re
dirurl&timeleft=$timeleft','$userurldecode', '$redirurldecode', '$timeleft')\" onBlur =
\"javascript:doOnBlur($result)\" bgColor = '#c0d8f4'>";


#     if (!window.opener) {
#       document.bgColor = '#c0d8f4';
#     }

#print "THE INPUT: $input";
#foreach $key (sort (keys %ENV)) {
#          print $key, ' = ', $ENV{$key}, "<br>\n";
#}

if ($result == 2) {
   print "
  <h1 style=\"text-align: center;\">ChilliSpot Login Failed</h1>";
   if ($reply) {
          print "<center> $reply </BR></BR></center>";
   }
}

if ($result == 5) {
   print "
  <h1 style=\"text-align: center;\">ChilliSpot Login</h1>";
}

if ($result == 2 || $result == 5) {
  print "
  <form name=\"form1\" method=\"post\" action=\"$loginpath\">
  <INPUT TYPE=\"hidden\" NAME=\"challenge\" VALUE=\"$challenge\">
  <INPUT TYPE=\"hidden\" NAME=\"uamip\" VALUE=\"$uamip\">
  <INPUT TYPE=\"hidden\" NAME=\"uamport\" VALUE=\"$uamport\">
  <INPUT TYPE=\"hidden\" NAME=\"userurl\" VALUE=\"$userurl\">
  <center>
  <table border=\"0\" cellpadding=\"5\" cellspacing=\"0\" style=\"width: 217px;\">
   <tbody>
    <tr>
```

```
      <td align=\"right\">Clave:</td>
      <td><input STYLE=\"font-family: Arial\" type=\"text\" name=\"UserName\" size=\"20\"
maxlength=\"128\"></td>
    </tr>
    <tr>
      <td><input STYLE=\"font-family: Arial\" type=\"hidden\" name=\"Password\"
value=\"metanet\" size=\"20\" maxlength=\"128\"></td>
    </tr>
    <tr>
      <td align=\"center\" colspan=\"2\" height=\"23\"><input type=\"submit\"
name=\"button\" value=\"Login\"
onClick=\"javascript:popUp('$loginpath?res=popup1&uamip=$uamip&uamport=$uamport')\"
></td>
    </tr>
   </tbody>
  </table>
  </center>
  </form>
</body>
</html>";
}

if ($result == 1) {
  print "
  <h1 style=\"text-align: center;\">Logged in to ChilliSpot</h1>";

  if ($reply) {
     print "<center> $reply </BR></BR></center>";
  }

  print "
  <center>
   <a href=\"http://$uamip:$uamport/logoff\">Logout</a>
  </center>
</body>
</html>";
}

if (($result == 4) || ($result == 12)) {
  print "
  <h1 style=\"text-align: center;\">Logged in to ChilliSpot</h1>
  <center>
   <a href=\"http://$uamip:$uamport/logoff\">Logout</a>
  </center>
</body>
</html>";
}


if ($result == 11) {
  print "<h1 style=\"text-align: center;\">Logging in to ChilliSpot</h1>";
  print "
```

```
  <center>
    Please wait.....
  </center>
</body>
</html>";
}


if (($result == 3) || ($result == 13)) {
    print "
  <h1 style=\"text-align: center;\">Logged out from ChilliSpot</h1>
  <center>
    <a href=\"http://$uamip:$uamport/prelogin\">Login</a>
  </center>
</body>
</html>";
}

exit(0);
```

Apéndice 7. Archivo de configuración del chillispot

```
##################################################################################
#
#
# Sample ChilliSpot configuration file
#
##################################################################################
#

# TAG: fg
# Include this flag if process is to run in the foreground
#fg

# TAG: debug
# Include this flag to include debug information.
#debug

# TAG: interval
# Re-read configuration file at this interval. Will also cause new domain
# name lookups to be performed. Value is given in seconds.
interval 3600

# TAG: pidfile
# File to store information about the process id of the program.
# The program must have write access to this file/directory.
#pidfile /var/run/chilli.pid

# TAG: statedir
# Directory to use for nonvolatile storage.
# The program must have write access to this directory.
# This tag is currently ignored
#statedir ./


# TUN parameters

# TAG: net
# IP network address of external packet data network
# Used to allocate dynamic IP addresses and set up routing.
# Normally you do not need to uncomment this tag.
net 192.168.182.0/24

# TAG: dynip
# Dynamic IP address pool
# Used to allocate dynamic IP addresses to clients.
# If not set it defaults to the net tag.
# Do not uncomment this tag unless you are an experienced user!
#dynip 192.168.182.0/24
```

```
# TAG: statip
# Static IP address pool
# Used to allocate static IP addresses to clients.
# Do not uncomment this tag unless you are an experienced user!
#statip 192.168.182.0/24


# TAG: dns1
# Primary DNS server.
# Will be suggested to the client.
# If omitted the system default will be used.
# Normally you do not need to uncomment this tag.
dns1 172.16.0.5

# TAG: dns2
# Secondary DNS server.
# Will be suggested to the client.
# If omitted the system default will be used.
# Normally you do not need to uncomment this tag.
dns2 172.16.0.6

# TAG: domain
# Domain name
# Will be suggested to the client.
# Normally you do not need to uncomment this tag.
#domain key.chillispot.org

# TAG: ipup
# Script executed after network interface has been brought up.
# Executed with the following parameters: <devicename> <ip address>
# <mask>
# Normally you do not need to uncomment this tag.
#ipup /etc/chilli.ipup

# TAG: ipdown
# Script executed after network interface has been taken down.
# Executed with the following parameters: <devicename> <ip address>
# <mask>
# Normally you do not need to uncomment this tag.
#ipdown /etc/chilli.ipdown


# Radius parameters

# TAG: radiuslisten
# IP address to listen to
# Normally you do not need to uncomment this tag.
#radiuslisten 127.0.0.1

# TAG: radiusserver1
# IP address of radius server 1
```

```
# For most installations you need to modify this tag.
radiusserver1 127.0.0.1

# TAG: radiusserver2
# IP address of radius server 2
# If you have only one radius server you should set radiusserver2 to the
# same value as radiusserver1.
# For most installations you need to modify this tag.
radiusserver2 127.0.0.1

# TAG: radiusauthport
# Radius authentication port
# The UDP port number to use for radius authentication requests.
# The same port number is used for both radiusserver1 and radiusserver2.
# Normally you do not need to uncomment this tag.
#radiusauthport 1812

# TAG: radiusacctport
# Radius accounting port
# The UDP port number to use for radius accounting requests.
# The same port number is used for both radiusserver1 and radiusserver2.
# Normally you do not need to uncomment this tag.
#radiusacctport 1813

# TAG: radiussecret
# Radius shared secret for both servers
# For all installations you should modify this tag.
radiussecret testing123

# TAG: radiusnasid
# Radius NAS-Identifier
# Normally you do not need to uncomment this tag.
#radiusnasid nas01

# TAG: radiuslocationid
# WISPr Location ID. Should be in the format: isocc=<ISO_Country_Code>,
# cc=<E.164_Country_Code>,ac=<E.164_Area_Code>,network=<ssid/ZONE>
# Normally you do not need to uncomment this tag.
#radiuslocationid isocc=us,cc=1,ac=408,network=ACMEWISP_NewarkAirport

# TAG: radiuslocationname
# WISPr Location Name. Should be in the format:
# <HOTSPOT_OPERATOR_NAME>,<LOCATION>
# Normally you do not need to uncomment this tag.
#radiuslocationname ACMEWISP,Gate_14_Terminal_C_of_Newark_Airport


# Radius proxy parameters

# TAG: proxylisten
# IP address to listen to
# Normally you do not need to uncomment this tag.
```

```
#proxylisten 10.0.0.1

# TAG: proxyport
# UDP port to listen to.
# If not specified a port will be selected by the system
# Normally you do not need to uncomment this tag.
#proxyport 1645

# TAG: proxyclient
# Client(s) from which we accept radius requests
# Normally you do not need to uncomment this tag.
#proxyclient 10.0.0.1/24

# TAG: proxysecret
# Radius proxy shared secret for all clients
# If not specified defaults to radiussecret
# Normally you do not need to uncomment this tag.
#proxysecret testing123


# DHCP Parameters

# TAG: dhcpif
# Ethernet interface to listen to.
# This is the network interface which is connected to the access points.
# In a typical configuration this tag should be set to eth1.
dhcpif eth1

# TAG: dhcpmac
# Use specified MAC address.
# An address in the range  00:00:5E:00:02:00 - 00:00:5E:FF:FF:FF falls
# within the IANA range of addresses and is not allocated for other
# purposes.
# Normally you do not need to uncomment this tag.
#dhcpmac 00:00:5E:00:02:00

# TAG: lease
# Time before DHCP lease expires
# Normally you do not need to uncomment this tag.
#lease 600


# Universal access method (UAM) parameters

# TAG: uamserver
# URL of web server handling authentication.
uamserver https://radius.chillispot.org/hotspotlogin

# TAG: uamhomepage
# URL of welcome homepage.
# Unauthenticated users will be redirected to this URL. If not specified
# users will be redirected to the uamserver instead.
```

```
# Normally you do not need to uncomment this tag.
uamhomepage http://192.168.182.1/welcome.html

# TAG: uamsecret
# Shared between chilli and authentication web server
uamsecret ht2eb8ej6s4et3rg1ulp

# TAG: uamlisten
# IP address to listen to for authentication requests
# Do not uncomment this tag unless you are an experienced user!
uamlisten 192.168.182.1

# TAG: uamport
# TCP port to listen to for authentication requests
# Do not uncomment this tag unless you are an experienced user!
uamport 3990

# TAG: uamallowed
# Comma separated list of domain names, IP addresses or network segments
# the client can access without first authenticating.
# It is possible to specify this tag multiple times.
# Normally you do not need to uncomment this tag.
uamallowed 192.168.182.0/24,12.0.0.0/24

# TAG: uamanydns
# If this flag is given unauthenticated users are allowed to use
# any DNS server.
# Normally you do not need to uncomment this tag.
#uamanydns


# MAC authentication

# TAG: macauth
# If this flag is given users will be authenticated only on their MAC
# address.
# Normally you do not need to uncomment this tag.
#macauth

# TAG: macallowed
# List of MAC addresses.
# The MAC addresses specified in this list will be authenticated only on
# their MAC address.
# This tag is ignored if the macauth tag is given.
# It is possible to specify this tag multiple times.
# Normally you do not need to uncomment this tag.
#macallowed 00-0A-5E-AC-BE-51,00-30-1B-3C-32-E9

# TAG: macpasswd
# Password to use for MAC authentication.
# Normally you do not need to uncomment this tag.
#macpasswd password
```

TAG: macsuffix
Suffix to add to MAC address in order to form the username.
Normally you do not need to uncomment this tag.
#macsuffix suffix

Apéndice 8. Reglas del cortafuegos de la raspberry.

```
#!/bin/sh
echo "1" > /proc/sys/net/ipv4/ip_forward;
iptables -F;
iptables -X;
iptables -Z;
iptables -t nat -F;
iptables -P INPUT ACCEPT;
iptables -P OUTPUT ACCEPT;
iptables -P FORWARD ACCEPT;
iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 9100 -j DNAT --to-destination
192.168.0.23:9100;
iptables -t nat -A POSTROUTING -j MASQUERADE;
```

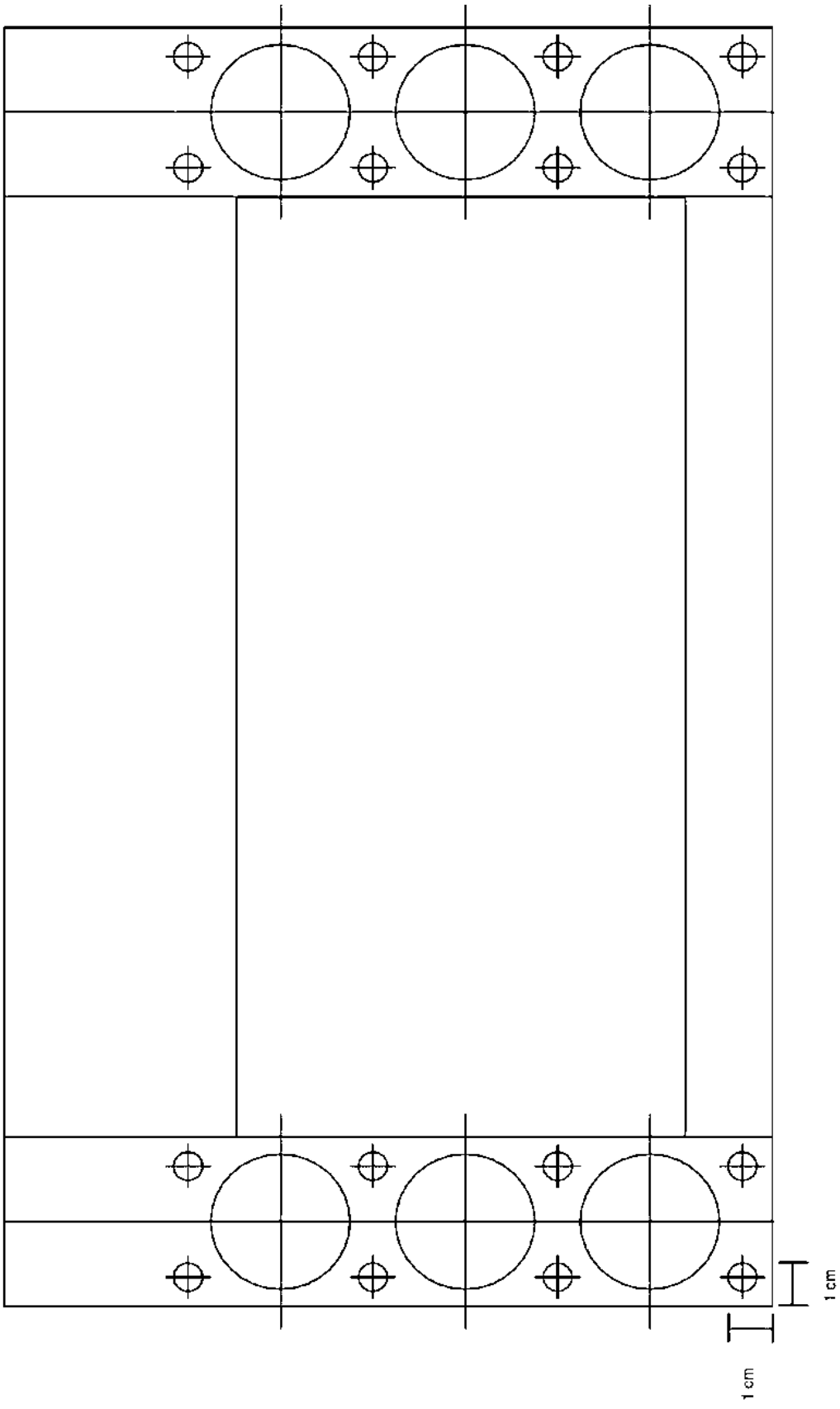
Apéndice 9. Asignación de los recursos de red a la interfaz USB de la impresora.

```
#!/bin/sh
GW_RASPBERRY="192.168.0.1";
while [ "1" ]
do
#se extrae la ip actual de eth0.
        if [ "$(ifconfig | grep eth0 | awk {'print $1'})" = "eth0" ]; #Revisa si eth0 está levantada
 then
                iface=$(ifconfig eth0 | grep "inet addr" | awk {'print $2'} | sed "s/:/ /g" | awk
{'print $2'});
                if [ "$iface" != "$GW_RASPBERRY" ]; #Revisa que la ip de eth0 sea la misma de
GW_RASPBERRY.
                #Si es distinta la reconfigura.
                then
                        ifconfig eth0 $ GW_RASPBERRY netmask 255.255.255.0;
                fi;
        else #Si no está levantada la levanta y la configura
                ifconfig eth0 up;
                ifconfig eth0 $ GW_RASPBERRY netmask 255.255.255.0;
        fi;
        sleep 5; # Tiempo de intervalo entre revisión y revisión a eth0
done;
```

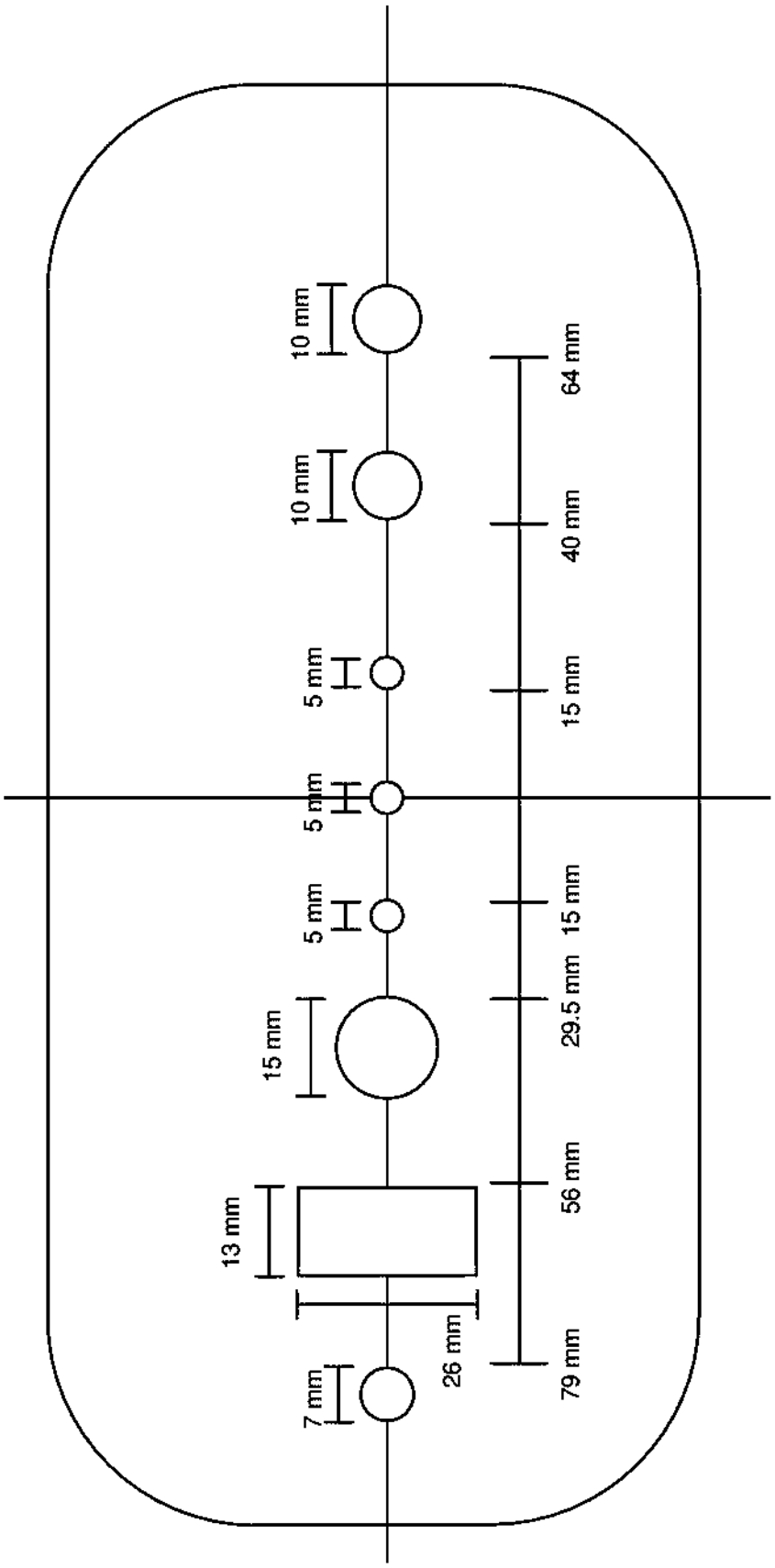
Apéndice 10. Código del script de arranque que inicia el programa de venta.

```
#!/bin/sh
nohup /home/pi/maquinaexpedidora/maquina_expedidora.run &
nohup sh /root/autoip_eth0.sh &
sh /root/firewall.sh;
rm nohup.out;
```

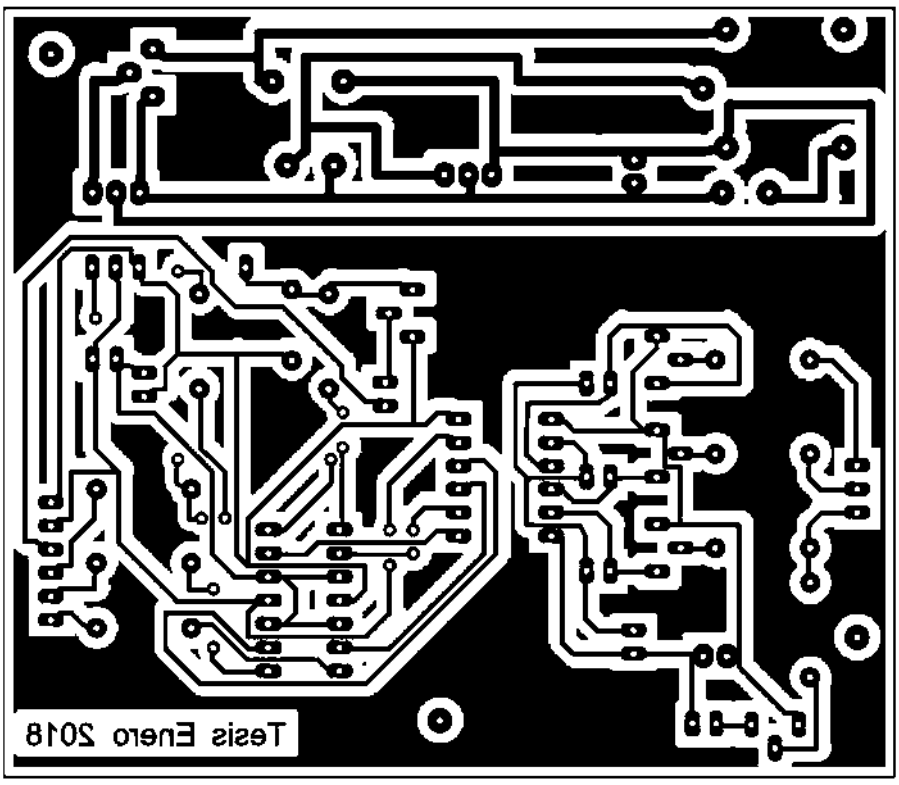
Anexo 1. Perforaciones hechas al gabinete del módulo de acceso inalámbrico.



Anexo 2. Medidas del panel de control del módulo de acceso inalámbrico.



Anexo 3. Circuito PCB de la tarjeta de arranque.



Anexo 4. Contenido del archivo "registrar_planes.sql"

```
insert into
radgroupcheck(groupname,attribute,op,value)\
values('2horas','Session-Timeout',':=','7200');
insert into
radgroupcheck(groupname,attribute,op,value)\
values('2horas','Simultaneous-Use',':=','1');
insert into
radgroupreply(groupname,attribute,op,value)\
values('2horas','WISPr-Bandwidth-Max-
Down',':=','512000');
insert into
radgroupreply(groupname,attribute,op,value)\
values('2horas','WISPr-Bandwidth-Max-Up',':=','256000');
insert into
radgroupreply(groupname,attribute,op,value)\
values('2horas','Session-Timeout',':=','7200');
insert into
radgroupreply(groupname,attribute,op,value)\
values('2horas','Idle-Timeout',':=','300');
insert into
radgroupreply(groupname,attribute,op,value)\
values('2horas','Acct-Interim-Interval',':=','120');
insert into
radgroupcheck(groupname,attribute,op,value)\
values('1dia','Session-Timeout',':=','86400');
insert into
radgroupcheck(groupname,attribute,op,value)\
values('1dia','Simultaneous-Use',':=','1');
insert into
radgroupreply(groupname,attribute,op,value)\
values('1dia','WISPr-Bandwidth-Max-
Down',':=','512000');
insert into
radgroupreply(groupname,attribute,op,value)\
values('1dia','WISPr-Bandwidth-Max-Up',':=','256000');
insert into
radgroupreply(groupname,attribute,op,value)\
values('1dia','Session-Timeout',':=','86400');
insert into
radgroupreply(groupname,attribute,op,value)\
values('1dia','Idle-Timeout',':=','300');
insert into
radgroupreply(groupname,attribute,op,value)\
values('1dia','Acct-Interim-Interval',':=','120');
insert into radgroupcheck(groupname,attribute,op,value)\
values('7dias','Session-Timeout',':=','604800');
insert into radgroupcheck(groupname,attribute,op,value)\
values('7dias','Simultaneous-Use',':=','1');
insert into radgroupreply(groupname,attribute,op,value)\
values('7dias','WISPr-Bandwidth-Max-Down',':=','512000');
insert into radgroupreply(groupname,attribute,op,value)\
values('7dias','WISPr-Bandwidth-Max-Up',':=','256000');
insert into radgroupreply(groupname,attribute,op,value)\
values('7dias','Session-Timeout',':=','604800');
insert into radgroupreply(groupname,attribute,op,value)\
values('7dias','Idle-Timeout',':=','300');
insert into radgroupreply(groupname,attribute,op,value)\
values('7dias','Acct-Interim-Interval',':=','120');
insert into radgroupcheck(groupname,attribute,op,value)\
values('15dias','Session-Timeout',':=','1296000');
insert into radgroupcheck(groupname,attribute,op,value)\
values('15dias','Simultaneous-Use',':=','1');
insert into radgroupreply(groupname,attribute,op,value)\
values('15dias','WISPr-Bandwidth-Max-Down',':=','512000');
insert into radgroupreply(groupname,attribute,op,value)\
values('15dias','WISPr-Bandwidth-Max-Up',':=','256000');
insert into radgroupreply(groupname,attribute,op,value)\
values('15dias','Session-Timeout',':=','1296000');
insert into radgroupreply(groupname,attribute,op,value)\
values('15dias','Idle-Timeout',':=','300');
insert into radgroupreply(groupname,attribute,op,value)\
values('15dias','Acct-Interim-Interval',':=','120');
insert into radgroupcheck(groupname,attribute,op,value)\
values('30dias','Session-Timeout',':=','2592000');
insert into radgroupcheck(groupname,attribute,op,value)\
values('30dias','Simultaneous-Use',':=','1');
insert into radgroupreply(groupname,attribute,op,value)\
values('30dias','WISPr-Bandwidth-Max-Down',':=','512000');
insert into radgroupreply(groupname,attribute,op,value)\
values('30dias','WISPr-Bandwidth-Max-Up',':=','256000');
insert into radgroupreply(groupname,attribute,op,value)\
values('30dias','Session-Timeout',':=','2592000');
insert into radgroupreply(groupname,attribute,op,value)\
values('30dias','Idle-Timeout',':=','300');
insert into radgroupreply(groupname,attribute,op,value)\
values('30dias','Acct-Interim-Interval',':=','120');
```

Anexo 5. Código generador de claves por volumen.

```
#!/bin/sh
num_claves=1000;
url="/opt/wisp";
apg -n $num_claves -m 10 -x 10 -E \'\*\_\!\"\-
\`\"\ç\{\}\,\.\i\!\"\·\$\%\&\/\(\)\=\?\¿\,\#\|\@\#\~\+\º\ª\\\^\<\>\: | sed 's/;/P/g' | sed
's/\[/I/g' | sed 's/\]/D/g' > $url/claves.dat;
cp $url/claves.dat /ramdisk0;
```

Anexo 6. Código del script activador de claves.

```
#!/bin/sh

USRDB="root"; #usuario administrador mysql
CLVDB="toor"; # contraseña de administrador mysql
BSDT="radius"; #base de datos de usuarios
host="127.0.0.1"; # IP del servidor de la base de datos
SQL="use $BSDT;";


#estados
consultando_tiempo_expiracion=1;
verificacion_de_tiempo_y_validacion_de_plan=2;
extraer_clave_de_venta=3;
generando_datos_para_radcheck=4;
generando_datos_para_radreply=5;
generando_datos_para_radusergroup=6;
generando_datos_para_voucher=7;
generando_datos_para_folios_claves=8;
modificando_el_registro_contador=9;
activando_usuario_en_la_base_de_datos=10;
verificando_usuario_en_la_base_de_datos=11;
limpiar_archivos_temporales_de_trabajo=12;
generar_nuevas_claves=25;
clave_activada=99;
terminar=100;
estado_indefinido="*";

#configuracion_inicial;
estado=$consultando_tiempo_expiracion;
prueba="0"; #poner 1 para solo para modo de pruebas y 0 para modo normal
#no modifica tablas en mysql,
#no modifica contador,
#no vacía el archivo activar_cliente.sql

determinar_plan()
{
 plan="";
 case $1 in
  0) vigencia_numero="15"; vigencia_unidades="minutes"; plan="15minutos"; ;;
  1) vigencia_numero="30"; vigencia_unidades="minutes"; plan="30minutos"; ;;
  2) vigencia_numero="60"; vigencia_unidades="minutes"; plan="1hora"; ;;
  3) vigencia_numero="1"; vigencia_unidades="days"; plan="1dia"; ;;
  4) vigencia_numero="7"; vigencia_unidades="days"; plan="7dias"; ;;
  5) vigencia_numero="15"; vigencia_unidades="days"; plan="15dias"; ;;
  6) vigencia_numero="20"; vigencia_unidades="days"; plan="20dias"; ;;
  7) vigencia_numero="30"; vigencia_unidades="days"; plan="30dias"; ;;
  8) vigencia_numero="30"; vigencia_unidades="days"; plan="plan0conexion"; ;;
```

```
 9) vigencia_numero="30"; vigencia_unidades="days"; plan="invitados"; ;;
# 10); ;;
# 11); ;;
# 12); ;;
esac;
if [ "$plan" = "" ]; then echo "0"; else echo "1 $vigencia_numero $vigencia_unidades $plan"; fi;
}

determinar_vigencia()
{
if [ "$1" = "1" ]; then
 fecha_expiracion=$(date -d "$2 $3"); #se calcula la fecha y hora de expiracion
 hh_mm_ss=$(echo $fecha_expiracion | awk '{print $4}'); # se extrae la hora  de terminacion
 dd=$(echo $fecha_expiracion | awk '{print $3}'); #se extrae el día de terminacion
 mm=$(echo $fecha_expiracion | awk '{print $2}'); #se extrae el mes de terminacion
 aa=$(echo $fecha_expiracion | awk '{print $6}'); #se extrae el año de terminacion
 case $mm in #se convierte al formato de fechas de las tablas de mysql que freeradius lee
  Jan) mm=January; mme=Ene; mes=1; ;;   jan) mm=January; mme=Ene; mes=1; ;;
  Ene) mm=January; mme=Ene; mes=1; ;;   ene) mm=January; mme=Ene; mes=1; ;;
  Feb) mm=February; mme=Feb; mes=2; ;;  feb) mm=February; mme=Feb; mes=2; ;;
#  Feb) mm=February; mme=Feb; mes=2; ;;  feb) mm=February; mme=Feb; mes=2; ;;
  Mar) mm=March; mme=Mar; mes=3; ;;     mar) mm=March; mme=Mar; mes=3; ;;
#  Mar) mm=March; mme=Mar; mes=3; ;;     mar) mm=March; mme=Mar; mes=3; ;;
  Apr) mm=April; mme=Abr; mes=4; ;;     apr) mm=April; mme=Abr; mes=4; ;;
  Abr) mm=April; mme=Abr; mes=4; ;;     abr) mm=April; mme=Abr; mes=4; ;;
  May) mm=May; mme=May; mes=5; ;;       may) mm=May; mme=May; mes=5; ;;
#  May) mm=May; mme=May; mes=5; ;;       may) mm=May; mme=May; mes=5; ;;
  Jun) mm=June; mme=Jun; mes=6; ;;      jun) mm=June; mme=Jun; mes=6; ;;
#  Jun) mm=June; mme=Jun; mes=6; ;;      jun) mm=June; mme=Jun; mes=6; ;;
  Jul) mm=July; mme=Jul; mes=7; ;;      jul) mm=July; mme=Jul; mes=7; ;;
#  Jul) mm=July; mme=Jul; mes=7; ;;      jul) mm=July; mme=Jul; mes=7; ;;
  Aug) mm=August; mme=Ago; mes=8; ;;    aug) mm=August; mme=Ago; mes=8; ;;
  Ago) mm=August; mme=Ago; mes=8; ;;    ago) mm=August; mme=Ago; mes=8; ;;
  Sep) mm=September; mme=Sep; mes=9; ;; sep) mm=September; mme=Sep; mes=9; ;;
#  Sep) mm=September; mme=Sep; mes=9; ;; sep) mm=September; mme=Sep; mes=9; ;;
  Oct) mm=October; mme=Oct; mes=10; ;;  oct) mm=October; mme=Oct; mes=10; ;;
#  Oct) mm=October; mme=Oct; mes=10; ;;  oct) mm=October; mme=Oct; mes=10; ;;
  Nov) mm=November; mme=Nov; mes=11; ;; nov) mm=November; mme=Nov; mes=11; ;;
#  Nov) mm=November; mme=Nov; mes=11; ;; nov) mm=November; mme=Nov; mes=11; ;;
  Dec) mm=December; mme=Dec; mes=12; ;; dec) mm=December; mme=Dec; mes=12; ;;
  Dic) mm=December; mme=Dec; mes=12; ;; dic) mm=December; mme=Dec; mes=12; ;;
 esac;
 echo $mm $dd $aa $hh_mm_ss $aa-$mes-$dd\T$hh_mm_ss $dd-$mme-$aa $hh_mm_ss $4; # se retornan los
valores obtenidos
 else echo "0";
 fi;
}
#estado=$terminar;
while [ $estado != $terminar ]
do
 case $estado in
  $consultando_tiempo_expiracion)
   expiracion=$(determinar_vigencia $(determinar_plan $*));
   if [ "$expiracion" != "0" ]; then
    expiracion_forma1=$(echo $expiracion | awk '{print $1,$2,$3,$4}');
    expiracion_forma2=$(echo $expiracion | awk '{print $5}');
    expiracion_forma3=$(echo $expiracion | awk '{print $6}');
    expiracion_forma4=$(echo $expiracion | awk '{print $7}');
    plan=$(echo $expiracion | awk '{print $8}');
    estado=$extraer_clave_de_venta;
   else echo "Error"; break;
   fi;
  ;;
```

```
 $extraer_clave_de_venta) #Extraccion de la clave num $contador
  contador=$(cat contador.dat);
  if [ $contador -lt "1001" ]; then
   usuario=$(cat claves.dat | head -$contador | tail -1);
   contrasenia="metanet";
   estado=$generando_datos_para_radcheck;
  else
   estado=$generar_nuevas_claves;
   echo "1" > contador.dat;
  fi;
 ;;

 $generando_datos_para_radcheck) #radcheck
  SQL=$(echo $SQL"insert into radcheck(username,attribute,op,value) values('$usuario','Cleartext-
Password',':=','$contrasenia');");
  SQL=$(echo $SQL"insert into radcheck(username,attribute,op,value)
values('$usuario','Expiration',':=','$expiracion_forma1');");
  estado=$generando_datos_para_radreply;
 ;;

 $generando_datos_para_radreply) #radreply
  SQL=$(echo $SQL"insert into radreply(username,attribute,op,value) values('$usuario','WISPr-Session-Terminate-
Time',':=','$expiracion_forma2');");
  estado=$generando_datos_para_radusergroup;
 ;;

 $generando_datos_para_radusergroup) #radusergroup
  SQL=$(echo $SQL"insert into radusergroup(username,groupname,priority) values('$usuario','$plan',1);");
  estado=$generando_datos_para_voucher;
 ;;

 $generando_datos_para_voucher) #voucher
  SQL=$(echo $SQL"insert into voucher(username,password,billingplan,isprinted)
values('$usuario','$contrasenia','$plan','1');");
  estado=$generando_datos_para_folios_claves;
 ;;

 $generando_datos_para_folios_claves) #folios_claves
  SQL=$(echo $SQL"insert into folios_claves(clave,plan,minutos,fecha_expiracion)
values('$usuario','$plan','0','$expiracion_forma2');");
  estado=$activando_usuario_en_la_base_de_datos;
 ;;

 $activando_usuario_en_la_base_de_datos)
  echo $SQL > activar_cliente.sql;
  if [ "$prueba" = "0" ]; then mysql -u$USRDB -p$CLVDB -h$host < activar_cliente.sql; fi;
  estado=$verificando_usuario_en_la_base_de_datos;
 ;;

 $verificando_usuario_en_la_base_de_datos) #.....pendiente.....#
#  si_exitoso_imprimir_clave (y continuar) 1)
#  si_no_repetir_el_proceso_de_nuevo 2)
  estado=$modificando_el_registro_contador;
 ;;

 $modificando_el_registro_contador)
  if [ "$prueba" = "0" ]; then echo $(expr $contador + 1) > contador.dat; fi;
  estado=$clave_activada;
 ;;

 $clave_activada)
  echo $usuario $expiracion_forma3 $expiracion_forma4;
  estado=$limpiar_archivos_temporales_de_trabajo;
```

```
 ;;

 $limpiar_archivos_temporales_de_trabajo)
  if [ "$prueba" = "0" ]; then echo "" > activar_cliente.sql; fi;
  estado=$terminar;
  break;
 ;;

 $generar_nuevas_claves)
 sh generador_claves.sh;
 sleep 2;
 estado=$consultando_tiempo_expiracion;
 ;;

 $terminar)
  estado=$terminar;
 ;;

 $estado_indefinido)
  break;
 ;;
esac;
done;
exit;
```

Anexo 7. Plantilla generadora de certificados SSL.

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
RANDFILE                = $ENV::HOME/.rnd

# Extra OBJECT IDENTIFIER info:
#oid_file               = $ENV::HOME/.oid
oid_section             = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions            =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
```

```
[ CA_default ]

dir             = .             # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl      # Where the issued crl are kept
database = $dir/index.txt       # database index file.
#unique_subject = no                    # Set to 'no' to allow creation of
                                        # several ctificates with same subject.
new_certs_dir   = $dir/certificados             # default place for new certs.

certificate     = $dir/certificado_autoridad/cacert.cert        # The CA certificate
serial          = $dir/serial.txt               # The current serial number
crlnumber       = $dir/crlnumber    # the current crl number
                                        # must be commented out to leave a V1 CRL
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/certificado_autoridad/cakey.key# The private key
RANDFILE        = $dir/privado/.rand            # private random number file

x509_extensions = usr_cert              # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt        = ca_default            # Subject Name options
cert_opt  = ca_default          # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions    = crl_ext

default_days    = 3650                  # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = sha1                  # which md to use.
preserve  = no                  # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName  = match
organizationalUnitName      = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName  = optional
organizationalUnitName      = optional
commonName              = supplied
emailAddress            = optional
```

```
################################################################
[ req ]
default_bits            = 1024
default_keyfile      = key.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions    = v3_ca   # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix      : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Mexico
countryName_default             = MX
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = Michoacan de Ocampo
stateOrProvinceName_default     = Michoacan de Ocampo

localityName                    = Morelia
localityName_default = Morelia
0.organizationName              = Internet emonarca
0.organizationName_default = Internet emonarca

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Internet emonarca
organizationalUnitName_default  = Internet emonarca

commonName                       = e-Monarca
commonName_default = e-Monarca

commonName_max                          = 64

emailAddress                    = metarmorfosisnet@yahoo.com
emailAddress_default        = metarmorfosisnet@yahoo.com
emailAddress_max            = 64

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword           = A challenge password
challengePassword_min               = 4
challengePassword_max               = 20

unstructuredName            = www.emonarca.net
```

```
unstructuredName_default = www.emonarca.net
[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                     = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment                      = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl              = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]

# Extensions for a typical CA
# PKIX recommendation.
```

```
subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always

[ proxy_cert_ext ]
# These extensions should be added when creating a proxy certificate

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                    = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment
```

```
# This will be displayed in Netscape's comment listbox.
nsComment                       = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl              = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName
# This really needs to be in place for it to be a proxy certificate.
proxyCertInfo=critical,language:id-ppl-anyLanguage,pathlen:3,policy:foo
```

Anexo 8. Código html de la página de bienvenida del portal captivo.

```
<html xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:w="urn:schemas-microsoft-com:office:word"
xmlns:m="http://schemas.microsoft.com/office/2004/12/omml"
xmlns="http://www.w3.org/TR/REC-html40">

<head>
<meta http-equiv=Content-Type content="text/html; charset=windows-1252">
<meta name=ProgId content=Word.Document>
<meta name=Generator content="Microsoft Word 14">
<meta name=Originator content="Microsoft Word 14">
<link rel=File-List href="index_archivos/filelist.xml">
<link rel=Edit-Time-Data href="index_archivos/editdata.mso">
<!--[if !mso]>
<style>
v\:* {behavior:url(#default#VML);}
o\:* {behavior:url(#default#VML);}
w\:* {behavior:url(#default#VML);}
.shape {behavior:url(#default#VML);}
</style>
<![endif]--><!--[if gte mso 9]><xml>
 <o:DocumentProperties>
  <o:Author>/-/ GP /-/</o:Author>
  <o:LastAuthor>/-/ GP /-/</o:LastAuthor>
  <o:Revision>32</o:Revision>
  <o:TotalTime>149</o:TotalTime>
  <o:Created>2017-11-15T10:36:00Z</o:Created>
  <o:LastSaved>2017-11-20T19:15:00Z</o:LastSaved>
  <o:Pages>1</o:Pages>
  <o:Characters>3</o:Characters>
  <o:Company>WarezMaster®</o:Company>
  <o:Lines>1</o:Lines>
```

```
 <o:Paragraphs>1</o:Paragraphs>
 <o:CharactersWithSpaces>3</o:CharactersWithSpaces>
 <o:Version>14.00</o:Version>
</o:DocumentProperties>
<o:OfficeDocumentSettings>
 <o:AllowPNG/>
</o:OfficeDocumentSettings>
</xml><![endif]-->
<link rel=dataStoreItem href="index_archivos/item0010.xml"
target="index_archivos/props011.xml">
<link rel=themeData href="index_archivos/themedata.thmx">
<link rel=colorSchemeMapping href="index_archivos/colorschememapping.xml">
<!--[if gte mso 9]><xml>
 <w:WordDocument>
  <w:SpellingState>Clean</w:SpellingState>
  <w:GrammarState>Clean</w:GrammarState>
  <w:TrackMoves>false</w:TrackMoves>
  <w:TrackFormatting/>
  <w:HyphenationZone>21</w:HyphenationZone>
  <w:PunctuationKerning/>
  <w:ValidateAgainstSchemas/>
  <w:SaveIfXMLInvalid>false</w:SaveIfXMLInvalid>
  <w:IgnoreMixedContent>false</w:IgnoreMixedContent>
  <w:AlwaysShowPlaceholderText>false</w:AlwaysShowPlaceholderText>
  <w:DoNotPromoteQF/>
  <w:LidThemeOther>ES</w:LidThemeOther>
  <w:LidThemeAsian>X-NONE</w:LidThemeAsian>
  <w:LidThemeComplexScript>X-NONE</w:LidThemeComplexScript>
  <w:Compatibility>
   <w:BreakWrappedTables/>
   <w:SnapToGridInCell/>
   <w:WrapTextWithPunct/>
   <w:UseAsianBreakRules/>
   <w:DontGrowAutofit/>
   <w:SplitPgBreakAndParaMark/>
   <w:EnableOpenTypeKerning/>
   <w:DontFlipMirrorIndents/>
   <w:OverrideTableStyleHps/>
  </w:Compatibility>
  <w:BrowserLevel>MicrosoftInternetExplorer4</w:BrowserLevel>
  <m:mathPr>
   <m:mathFont m:val="Cambria Math"/>
   <m:brkBin m:val="before"/>
   <m:brkBinSub m:val="&#45;-"/>
   <m:smallFrac m:val="off"/>
   <m:dispDef/>
   <m:lMargin m:val="0"/>
   <m:rMargin m:val="0"/>
   <m:defJc m:val="centerGroup"/>
   <m:wrapIndent m:val="1440"/>
   <m:intLim m:val="subSup"/>
   <m:naryLim m:val="undOvr"/>
  </m:mathPr></w:WordDocument>
</xml><![endif]--><!--[if gte mso 9]><xml>
 <w:LatentStyles DefLockedState="false" DefUnhideWhenUsed="true"
  DefSemiHidden="true" DefQFormat="false" DefPriority="99"
  LatentStyleCount="267">
  <w:LsdException Locked="false" Priority="0" SemiHidden="false"
   UnhideWhenUsed="false" QFormat="true" Name="Normal"/>
  <w:LsdException Locked="false" Priority="9" SemiHidden="false"
   UnhideWhenUsed="false" QFormat="true" Name="heading 1"/>
  <w:LsdException Locked="false" Priority="9" QFormat="true" Name="heading 2"/>
  <w:LsdException Locked="false" Priority="9" QFormat="true" Name="heading 3"/>
```

<w:LsdException Locked="false" Priority="9" QFormat="true" Name="heading 4"/>
<w:LsdException Locked="false" Priority="9" QFormat="true" Name="heading 5"/>
<w:LsdException Locked="false" Priority="9" QFormat="true" Name="heading 6"/>
<w:LsdException Locked="false" Priority="9" QFormat="true" Name="heading 7"/>
<w:LsdException Locked="false" Priority="9" QFormat="true" Name="heading 8"/>
<w:LsdException Locked="false" Priority="9" QFormat="true" Name="heading 9"/>
<w:LsdException Locked="false" Priority="39" Name="toc 1"/>
<w:LsdException Locked="false" Priority="39" Name="toc 2"/>
<w:LsdException Locked="false" Priority="39" Name="toc 3"/>
<w:LsdException Locked="false" Priority="39" Name="toc 4"/>
<w:LsdException Locked="false" Priority="39" Name="toc 5"/>
<w:LsdException Locked="false" Priority="39" Name="toc 6"/>
<w:LsdException Locked="false" Priority="39" Name="toc 7"/>
<w:LsdException Locked="false" Priority="39" Name="toc 8"/>
<w:LsdException Locked="false" Priority="39" Name="toc 9"/>
<w:LsdException Locked="false" Priority="35" QFormat="true" Name="caption"/>
<w:LsdException Locked="false" Priority="10" SemiHidden="false"
 UnhideWhenUsed="false" QFormat="true" Name="Title"/>
<w:LsdException Locked="false" Priority="1" Name="Default Paragraph Font"/>
<w:LsdException Locked="false" Priority="11" SemiHidden="false"
 UnhideWhenUsed="false" QFormat="true" Name="Subtitle"/>
<w:LsdException Locked="false" Priority="22" SemiHidden="false"
 UnhideWhenUsed="false" QFormat="true" Name="Strong"/>
<w:LsdException Locked="false" Priority="20" SemiHidden="false"
 UnhideWhenUsed="false" QFormat="true" Name="Emphasis"/>
<w:LsdException Locked="false" Priority="59" SemiHidden="false"
 UnhideWhenUsed="false" Name="Table Grid"/>
<w:LsdException Locked="false" UnhideWhenUsed="false" Name="Placeholder Text"/>
<w:LsdException Locked="false" Priority="1" SemiHidden="false"
 UnhideWhenUsed="false" QFormat="true" Name="No Spacing"/>
<w:LsdException Locked="false" Priority="60" SemiHidden="false"
 UnhideWhenUsed="false" Name="Light Shading"/>
<w:LsdException Locked="false" Priority="61" SemiHidden="false"
 UnhideWhenUsed="false" Name="Light List"/>
<w:LsdException Locked="false" Priority="62" SemiHidden="false"
 UnhideWhenUsed="false" Name="Light Grid"/>
<w:LsdException Locked="false" Priority="63" SemiHidden="false"
 UnhideWhenUsed="false" Name="Medium Shading 1"/>
<w:LsdException Locked="false" Priority="64" SemiHidden="false"
 UnhideWhenUsed="false" Name="Medium Shading 2"/>
<w:LsdException Locked="false" Priority="65" SemiHidden="false"
 UnhideWhenUsed="false" Name="Medium List 1"/>
<w:LsdException Locked="false" Priority="66" SemiHidden="false"
 UnhideWhenUsed="false" Name="Medium List 2"/>
<w:LsdException Locked="false" Priority="67" SemiHidden="false"
 UnhideWhenUsed="false" Name="Medium Grid 1"/>
<w:LsdException Locked="false" Priority="68" SemiHidden="false"
 UnhideWhenUsed="false" Name="Medium Grid 2"/>
<w:LsdException Locked="false" Priority="69" SemiHidden="false"
 UnhideWhenUsed="false" Name="Medium Grid 3"/>
<w:LsdException Locked="false" Priority="70" SemiHidden="false"
 UnhideWhenUsed="false" Name="Dark List"/>
<w:LsdException Locked="false" Priority="71" SemiHidden="false"
 UnhideWhenUsed="false" Name="Colorful Shading"/>
<w:LsdException Locked="false" Priority="72" SemiHidden="false"
 UnhideWhenUsed="false" Name="Colorful List"/>
<w:LsdException Locked="false" Priority="73" SemiHidden="false"
 UnhideWhenUsed="false" Name="Colorful Grid"/>
<w:LsdException Locked="false" Priority="60" SemiHidden="false"
 UnhideWhenUsed="false" Name="Light Shading Accent 1"/>
<w:LsdException Locked="false" Priority="61" SemiHidden="false"
 UnhideWhenUsed="false" Name="Light List Accent 1"/>
<w:LsdException Locked="false" Priority="62" SemiHidden="false"

UnhideWhenUsed="false" Name="Light Grid Accent 1"/>
<w:LsdException Locked="false" Priority="63" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 1 Accent 1"/>
<w:LsdException Locked="false" Priority="64" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 2 Accent 1"/>
<w:LsdException Locked="false" Priority="65" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 1 Accent 1"/>
<w:LsdException Locked="false" UnhideWhenUsed="false" Name="Revision"/>
<w:LsdException Locked="false" Priority="34" SemiHidden="false"
UnhideWhenUsed="false" QFormat="true" Name="List Paragraph"/>
<w:LsdException Locked="false" Priority="29" SemiHidden="false"
UnhideWhenUsed="false" QFormat="true" Name="Quote"/>
<w:LsdException Locked="false" Priority="30" SemiHidden="false"
UnhideWhenUsed="false" QFormat="true" Name="Intense Quote"/>
<w:LsdException Locked="false" Priority="66" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 2 Accent 1"/>
<w:LsdException Locked="false" Priority="67" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 1 Accent 1"/>
<w:LsdException Locked="false" Priority="68" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 2 Accent 1"/>
<w:LsdException Locked="false" Priority="69" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 3 Accent 1"/>
<w:LsdException Locked="false" Priority="70" SemiHidden="false"
UnhideWhenUsed="false" Name="Dark List Accent 1"/>
<w:LsdException Locked="false" Priority="71" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful Shading Accent 1"/>
<w:LsdException Locked="false" Priority="72" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful List Accent 1"/>
<w:LsdException Locked="false" Priority="73" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful Grid Accent 1"/>
<w:LsdException Locked="false" Priority="60" SemiHidden="false"
UnhideWhenUsed="false" Name="Light Shading Accent 2"/>
<w:LsdException Locked="false" Priority="61" SemiHidden="false"
UnhideWhenUsed="false" Name="Light List Accent 2"/>
<w:LsdException Locked="false" Priority="62" SemiHidden="false"
UnhideWhenUsed="false" Name="Light Grid Accent 2"/>
<w:LsdException Locked="false" Priority="63" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 1 Accent 2"/>
<w:LsdException Locked="false" Priority="64" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 2 Accent 2"/>
<w:LsdException Locked="false" Priority="65" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 1 Accent 2"/>
<w:LsdException Locked="false" Priority="66" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 2 Accent 2"/>
<w:LsdException Locked="false" Priority="67" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 1 Accent 2"/>
<w:LsdException Locked="false" Priority="68" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 2 Accent 2"/>
<w:LsdException Locked="false" Priority="69" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 3 Accent 2"/>
<w:LsdException Locked="false" Priority="70" SemiHidden="false"
UnhideWhenUsed="false" Name="Dark List Accent 2"/>
<w:LsdException Locked="false" Priority="71" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful Shading Accent 2"/>
<w:LsdException Locked="false" Priority="72" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful List Accent 2"/>
<w:LsdException Locked="false" Priority="73" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful Grid Accent 2"/>
<w:LsdException Locked="false" Priority="60" SemiHidden="false"
UnhideWhenUsed="false" Name="Light Shading Accent 3"/>
<w:LsdException Locked="false" Priority="61" SemiHidden="false"
UnhideWhenUsed="false" Name="Light List Accent 3"/>
<w:LsdException Locked="false" Priority="62" SemiHidden="false"

UnhideWhenUsed="false" Name="Light Grid Accent 3"/>
<w:LsdException Locked="false" Priority="63" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 1 Accent 3"/>
<w:LsdException Locked="false" Priority="64" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 2 Accent 3"/>
<w:LsdException Locked="false" Priority="65" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 1 Accent 3"/>
<w:LsdException Locked="false" Priority="66" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 2 Accent 3"/>
<w:LsdException Locked="false" Priority="67" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 1 Accent 3"/>
<w:LsdException Locked="false" Priority="68" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 2 Accent 3"/>
<w:LsdException Locked="false" Priority="69" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 3 Accent 3"/>
<w:LsdException Locked="false" Priority="70" SemiHidden="false"
UnhideWhenUsed="false" Name="Dark List Accent 3"/>
<w:LsdException Locked="false" Priority="71" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful Shading Accent 3"/>
<w:LsdException Locked="false" Priority="72" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful List Accent 3"/>
<w:LsdException Locked="false" Priority="73" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful Grid Accent 3"/>
<w:LsdException Locked="false" Priority="60" SemiHidden="false"
UnhideWhenUsed="false" Name="Light Shading Accent 4"/>
<w:LsdException Locked="false" Priority="61" SemiHidden="false"
UnhideWhenUsed="false" Name="Light List Accent 4"/>
<w:LsdException Locked="false" Priority="62" SemiHidden="false"
UnhideWhenUsed="false" Name="Light Grid Accent 4"/>
<w:LsdException Locked="false" Priority="63" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 1 Accent 4"/>
<w:LsdException Locked="false" Priority="64" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 2 Accent 4"/>
<w:LsdException Locked="false" Priority="65" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 1 Accent 4"/>
<w:LsdException Locked="false" Priority="66" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 2 Accent 4"/>
<w:LsdException Locked="false" Priority="67" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 1 Accent 4"/>
<w:LsdException Locked="false" Priority="68" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 2 Accent 4"/>
<w:LsdException Locked="false" Priority="69" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Grid 3 Accent 4"/>
<w:LsdException Locked="false" Priority="70" SemiHidden="false"
UnhideWhenUsed="false" Name="Dark List Accent 4"/>
<w:LsdException Locked="false" Priority="71" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful Shading Accent 4"/>
<w:LsdException Locked="false" Priority="72" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful List Accent 4"/>
<w:LsdException Locked="false" Priority="73" SemiHidden="false"
UnhideWhenUsed="false" Name="Colorful Grid Accent 4"/>
<w:LsdException Locked="false" Priority="60" SemiHidden="false"
UnhideWhenUsed="false" Name="Light Shading Accent 5"/>
<w:LsdException Locked="false" Priority="61" SemiHidden="false"
UnhideWhenUsed="false" Name="Light List Accent 5"/>
<w:LsdException Locked="false" Priority="62" SemiHidden="false"
UnhideWhenUsed="false" Name="Light Grid Accent 5"/>
<w:LsdException Locked="false" Priority="63" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 1 Accent 5"/>
<w:LsdException Locked="false" Priority="64" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium Shading 2 Accent 5"/>
<w:LsdException Locked="false" Priority="65" SemiHidden="false"
UnhideWhenUsed="false" Name="Medium List 1 Accent 5"/>

```
 <w:LsdException Locked="false" Priority="66" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium List 2 Accent 5"/>
 <w:LsdException Locked="false" Priority="67" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium Grid 1 Accent 5"/>
 <w:LsdException Locked="false" Priority="68" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium Grid 2 Accent 5"/>
 <w:LsdException Locked="false" Priority="69" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium Grid 3 Accent 5"/>
 <w:LsdException Locked="false" Priority="70" SemiHidden="false"
  UnhideWhenUsed="false" Name="Dark List Accent 5"/>
 <w:LsdException Locked="false" Priority="71" SemiHidden="false"
  UnhideWhenUsed="false" Name="Colorful Shading Accent 5"/>
 <w:LsdException Locked="false" Priority="72" SemiHidden="false"
  UnhideWhenUsed="false" Name="Colorful List Accent 5"/>
 <w:LsdException Locked="false" Priority="73" SemiHidden="false"
  UnhideWhenUsed="false" Name="Colorful Grid Accent 5"/>
 <w:LsdException Locked="false" Priority="60" SemiHidden="false"
  UnhideWhenUsed="false" Name="Light Shading Accent 6"/>
 <w:LsdException Locked="false" Priority="61" SemiHidden="false"
  UnhideWhenUsed="false" Name="Light List Accent 6"/>
 <w:LsdException Locked="false" Priority="62" SemiHidden="false"
  UnhideWhenUsed="false" Name="Light Grid Accent 6"/>
 <w:LsdException Locked="false" Priority="63" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium Shading 1 Accent 6"/>
 <w:LsdException Locked="false" Priority="64" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium Shading 2 Accent 6"/>
 <w:LsdException Locked="false" Priority="65" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium List 1 Accent 6"/>
 <w:LsdException Locked="false" Priority="66" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium List 2 Accent 6"/>
 <w:LsdException Locked="false" Priority="67" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium Grid 1 Accent 6"/>
 <w:LsdException Locked="false" Priority="68" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium Grid 2 Accent 6"/>
 <w:LsdException Locked="false" Priority="69" SemiHidden="false"
  UnhideWhenUsed="false" Name="Medium Grid 3 Accent 6"/>
 <w:LsdException Locked="false" Priority="70" SemiHidden="false"
  UnhideWhenUsed="false" Name="Dark List Accent 6"/>
 <w:LsdException Locked="false" Priority="71" SemiHidden="false"
  UnhideWhenUsed="false" Name="Colorful Shading Accent 6"/>
 <w:LsdException Locked="false" Priority="72" SemiHidden="false"
  UnhideWhenUsed="false" Name="Colorful List Accent 6"/>
 <w:LsdException Locked="false" Priority="73" SemiHidden="false"
  UnhideWhenUsed="false" Name="Colorful Grid Accent 6"/>
 <w:LsdException Locked="false" Priority="19" SemiHidden="false"
  UnhideWhenUsed="false" QFormat="true" Name="Subtle Emphasis"/>
 <w:LsdException Locked="false" Priority="21" SemiHidden="false"
  UnhideWhenUsed="false" QFormat="true" Name="Intense Emphasis"/>
 <w:LsdException Locked="false" Priority="31" SemiHidden="false"
  UnhideWhenUsed="false" QFormat="true" Name="Subtle Reference"/>
 <w:LsdException Locked="false" Priority="32" SemiHidden="false"
  UnhideWhenUsed="false" QFormat="true" Name="Intense Reference"/>
 <w:LsdException Locked="false" Priority="33" SemiHidden="false"
  UnhideWhenUsed="false" QFormat="true" Name="Book Title"/>
 <w:LsdException Locked="false" Priority="37" Name="Bibliography"/>
 <w:LsdException Locked="false" Priority="39" QFormat="true" Name="TOC Heading"/>
 </w:LatentStyles>
</xml><![endif]-->
<style>
<!--
 /* Font Definitions */
 @font-face
        {font-family:Calibri;
```

```
        panose-1:2 15 5 2 2 2 4 3 2 4;
        mso-font-charset:0;
        mso-generic-font-family:swiss;
        mso-font-pitch:variable;
        mso-font-signature:-520092929 1073786111 9 0 415 0;}
@font-face
        {font-family:Tahoma;
        panose-1:2 11 6 4 3 5 4 4 2 4;
        mso-font-charset:0;
        mso-generic-font-family:swiss;
        mso-font-pitch:variable;
        mso-font-signature:1627400839 -2147483648 8 0 66047 0;}
 /* Style Definitions */
 p.MsoNormal, li.MsoNormal, div.MsoNormal
        {mso-style-unhide:no;
        mso-style-qformat:yes;
        mso-style-parent:"";
        margin-top:0cm;
        margin-right:0cm;
        margin-bottom:10.0pt;
        margin-left:0cm;
        line-height:115%;
        mso-pagination:widow-orphan;
        font-size:11.0pt;
        font-family:"Calibri","sans-serif";
        mso-ascii-font-family:Calibri;
        mso-ascii-theme-font:minor-latin;
        mso-fareast-font-family:Calibri;
        mso-fareast-theme-font:minor-latin;
        mso-hansi-font-family:Calibri;
        mso-hansi-theme-font:minor-latin;
        mso-bidi-font-family:"Times New Roman";
        mso-bidi-theme-font:minor-bidi;
        mso-fareast-language:EN-US;}
a:link, span.MsoHyperlink
        {mso-style-noshow:yes;
        mso-style-priority:99;
        color:blue;
        mso-themecolor:hyperlink;
        text-decoration:underline;
        text-underline:single;}
a:visited, span.MsoHyperlinkFollowed
        {mso-style-noshow:yes;
        mso-style-priority:99;
        color:purple;
        mso-themecolor:followedhyperlink;
        text-decoration:underline;
        text-underline:single;}
p.MsoAcetate, li.MsoAcetate, div.MsoAcetate
        {mso-style-noshow:yes;
        mso-style-priority:99;
        mso-style-link:"Texto de globo Car";
        margin:0cm;
        margin-bottom:.0001pt;
        mso-pagination:widow-orphan;
        font-size:8.0pt;
        font-family:"Tahoma","sans-serif";
        mso-fareast-font-family:Calibri;
        mso-fareast-theme-font:minor-latin;
        mso-fareast-language:EN-US;}
p.MsoNoSpacing, li.MsoNoSpacing, div.MsoNoSpacing
        {mso-style-priority:1;
        mso-style-unhide:no;
```

```
            mso-style-qformat:yes;
            mso-style-parent:"";
            margin:0cm;
            margin-bottom:.0001pt;
            mso-pagination:widow-orphan;
            font-size:11.0pt;
            font-family:"Calibri","sans-serif";
            mso-ascii-font-family:Calibri;
            mso-ascii-theme-font:minor-latin;
            mso-fareast-font-family:Calibri;
            mso-fareast-theme-font:minor-latin;
            mso-hansi-font-family:Calibri;
            mso-hansi-theme-font:minor-latin;
            mso-bidi-font-family:"Times New Roman";
            mso-bidi-theme-font:minor-bidi;
            mso-fareast-language:EN-US;}
span.TextodegloboCar
            {mso-style-name:"Texto de globo Car";
            mso-style-noshow:yes;
            mso-style-priority:99;
            mso-style-unhide:no;
            mso-style-locked:yes;
            mso-style-link:"Texto de globo";
            mso-ansi-font-size:8.0pt;
            mso-bidi-font-size:8.0pt;
            font-family:"Tahoma","sans-serif";
            mso-ascii-font-family:Tahoma;
            mso-hansi-font-family:Tahoma;
            mso-bidi-font-family:Tahoma;}
.MsoChpDefault
            {mso-style-type:export-only;
            mso-default-props:yes;
            font-size:10.0pt;
            mso-ansi-font-size:10.0pt;
            mso-bidi-font-size:10.0pt;
            font-family:"Calibri","sans-serif";
            mso-ascii-font-family:Calibri;
            mso-ascii-theme-font:minor-latin;
            mso-fareast-font-family:Calibri;
            mso-fareast-theme-font:minor-latin;
            mso-hansi-font-family:Calibri;
            mso-hansi-theme-font:minor-latin;
            mso-bidi-font-family:"Times New Roman";
            mso-bidi-theme-font:minor-bidi;
            mso-fareast-language:EN-US;}
@page WordSection1
            {size:595.3pt 841.9pt;
            margin:36.0pt 36.0pt 36.0pt 36.0pt;
            mso-header-margin:35.4pt;
            mso-footer-margin:35.4pt;
            mso-paper-source:0;}
div.WordSection1
            {page:WordSection1;}
-->
</style>
<!--[if gte mso 10]>
<style>
 /* Style Definitions */
 table.MsoNormalTable
            {mso-style-name:"Tabla normal";
            mso-tstyle-rowband-size:0;
            mso-tstyle-colband-size:0;
            mso-style-noshow:yes;
```

```
		mso-style-priority:99;
		mso-style-parent:"";
		mso-padding-alt:0cm 5.4pt 0cm 5.4pt;
		mso-para-margin:0cm;
		mso-para-margin-bottom:.0001pt;
		mso-pagination:widow-orphan;
		font-size:10.0pt;
		font-family:"Calibri","sans-serif";
		mso-ascii-font-family:Calibri;
		mso-ascii-theme-font:minor-latin;
		mso-hansi-font-family:Calibri;
		mso-hansi-theme-font:minor-latin;
		mso-bidi-font-family:"Times New Roman";
		mso-bidi-theme-font:minor-bidi;
		mso-fareast-language:EN-US;}
</style>
<![endif]--><!--[if gte mso 9]><xml>
 <o:shapedefaults v:ext="edit" spidmax="1028"/>
</xml><![endif]--><!--[if gte mso 9]><xml>
 <o:shapelayout v:ext="edit">
  <o:idmap v:ext="edit" data="1"/>
 </o:shapelayout></xml><![endif]-->
</head>

<body lang=ES link=blue vlink=purple style='tab-interval:35.4pt'>

<div class=WordSection1>

<p class=MsoNormal><!--[if gte vml 1]><v:shapetype id="_x0000_t202"
 coordsize="21600,21600" o:spt="202" path="m,l,21600r21600,l21600,xe">
 <v:stroke joinstyle="miter"/>
 <v:path gradientshapeok="t" o:connecttype="rect"/>
</v:shapetype><v:shape id="_x0033__x0020_Cuadro_x0020_de_x0020_texto" o:spid="_x0000_s1027"
 type="#_x0000_t202" href="http://12.0.0.1:3990/prelogin" style='position:absolute;
 margin-left:273.75pt;margin-top:335.25pt;width:111pt;height:27pt;z-index:251660288;
 visibility:visible;mso-wrap-style:square;mso-width-percent:0;
 mso-height-percent:0;mso-wrap-distance-left:9pt;mso-wrap-distance-top:0;
 mso-wrap-distance-right:9pt;mso-wrap-distance-bottom:0;
 mso-position-horizontal:absolute;mso-position-horizontal-relative:text;
 mso-position-vertical:absolute;mso-position-vertical-relative:text;
 mso-width-percent:0;mso-height-percent:0;mso-width-relative:margin;
 mso-height-relative:margin;v-text-anchor:top'
o:gfxdata="UEsDBBQABgAIAAAAIQC2gziS/gAAAOEBAAATAAAAW0NvbnRlbnRfVHlwZXNdLnhtbJSRQU7DMBBF
90jcwfIWJU67QAgl6YK0S0CoHGBkxx5tGhvj5O2O2G0SRWNoz/78nu9wcxkFMGNg6quQqL6RA
0s5Y6ir5vt9lD1JwBDIwOMJKHpHlpr69KfdHjyxSmriSfYz+USnWPY7AufNIadK6MEJMx9ApD/oD
OlTrorhX2lFEilmcO2RdNtjC5xDF9pCuTyYBB5bi6bQ4syoJ3g9WQ0ymaiLzg5KdCXlKLjvcW893
SUOqXwnz5DrgnHtJTxOsQfEKIT7DmDSUCaxw7Rqn8787ZsmRM9e2VmPeBN4uqYvTtW7jvijg9N/y
JsXecLq0q+WD6m8AAAD//wMAUEsDBBQABgAIAAAAIQA4/SH/1gAAAJQBAAALAAAAX3JlbHMvLnJl
bHOkkMFqwzAMhu+DvYPxDvYPiqBAAbgAIAAAAIQC2vig88a1a8/Bvu/d/dMb2JY5uiPbhG/ktn4cqGU/er3pcvwCAAD//wMAUEsDBBQABgAIAAAA
IQCvig88yQIAAB8GAAAOAAAAZHJzL2Uyb0RvYy54bWysVNtOGzEQfa/Uf7D7Uf7D8XnXnZnZgZYA
QIWKZ8frtVrzZbZfWy6tWKZ8fEQfa/uu7yXZgZYGxSCtTLkn5frvuvnuvnn2ZgZeRtTbcp5eeiPv
hRPEIBAB8GAAAOAAAAZHJzL2Uyb0RvYy54bWysVNtOGzEQfa/Uf7D7Uf7D8XnXnenO0KMWbq9Rg6pE
FUfdaDw8Rzk2ItaCzXFm3yb3/9zu82DjbHq6YgS
QlWKZ8frtVrxelzbSZZ+fcf25gKtVFH1xRp7zlzPeC4uu1aRjbBOgi7p4CSnRGgOldTLkn5/uvn0
hRLnma6YAi1K+iIcvZx8/HCxNYUYQgOqEpagE+2KrSlp470psszxRrTMnYARGpU12JZ5vNplVlm2
Re+tyoZ5fpZtwVbGAhfO4et1UtJJ9F/Xgvv7unbCE1VSzM3H08ZzEc5scsGKpWWmkbxPg/1DFi2T
GoPuXV0zz8jayt9ctZJbcFD7Ew5tBnUtuYg1YDWD/E01jw0zItaCzXFm3yb3/9zyu82DJbIq6YgS
zVqkaERma1ZZIJUgXnQeYmGNkno1U5Kv+jSwiX8nKxV4DXzdCu0TY1Yo5nFcXCONo8QWIbq9rQaB
jGxrXBGTChRG8dFghr67gg4HLObizBz4yh3BAyYZuCixoqttGwjBDAgaIvcve76xKMKDt3F+/jlH
FUfdaDw8Rzk4PVgb6/xXAS0JAmaJ8xQzYJu58wm6g4RgDpSsbqRS8WKXi5myZMPC7OVX+enO+yuY
0mRb0rMRaoOVhmCfXCsdXkQc4z5e6EkqMUr+RYmAUfqbqJHGWGkMHj6Q2IdnnGP/U4d7dEDVGOo9
hj3+kNV7jFMdaBEjg/Z741ZqsInZ12lXq13KdcL3A+JS3aEFvlt02K0gLqB6wVGxkH65M/xGIm1z
5vwDs/itkWlcVf4ej1oBdh16iZIG7M8/vQc8DjpqKdnimiip+7FmVlCibjX+w/PBeBz2SryMTz8P
```

8WKPNYtjjV63M8BpGOBSNDyKAe/VTqwttM+40aYhKqqY5hi7pH4nznxaXrgRuZhOIwg3iWF+rh8N
3/2QMJRP3TOzpp/c8JHvYLdQWPFmgBM2EKNhuvZQyzjdh672jcctFP9HvzHDmju+R9Rhr09+AQAA
//8DAFBLAwQUAAYACAAAACEA7sjz8+EAAAALAQAADwAAAGRycy9kb3ducmV2LnhtbEyPTUvDQBCG
74L/YRi9i9iNNR8asymhGAj01CqIt20yJqHZ2ZDdpvHfO5709gzz8s4z2WYxg5hxcr0lBQ+rAARS
bZueWgXvb+X9EwjnNTV6sIQKvtHBJr++ynTa2AvtcT74VnAJuVQr6LwfUyld3aHRbmVHJN592clo
z+PUymbSFy43g1wHQSyN7okvdHrEbYf16XA2CsqPu0/c+nLnq6Ioq7k67cPHV6Vub5biBYTHxf+F
4Vef1SFnp6M9U+PEoCAKk4ijCuIkYOBEEj8zHBnWYQQyz+T/H/IfAAAA//8DAFBLAwQUAAYACAAA
ACEAd5A+LtQAAABIAQAAGQAAAGRycy9fcmVscy9lMm9Eb2MueG1sLnJlbHOEz7FqAzEMANA90H8w
2nv2pVCacL4sbSFDlpJ+gLF1PhOfbA9pZueWgXvb+X9EwjnNTV6sIQKvtHBJr++ynTa2AvtcT74VnAJuVQr6LwfUyld3aHRbmVHJN592clo
UEsDBAoAAAAAAIQC2gziS/gAAAOEBAAAT
AAAAAAAAAAAAAAAAAAAAAAAABbQ29udGVudF9UeXBlc10ueG1sUEsBAi0AFAAGAAgAAAAhADj9If/W
AAAAlAEAAAsAAAAAAAAAAAAAAAAAAAAAALwEAAF9yZWxzLy5yZWxzUEsBAi0AFAAGAAgAAAAhAK+KDzzJ
AgAAHwYAAA4AAAAAAAAAAAAAAAAALgIAAGRycy9lMm9Eb2MueG1sUEsBAi0AFAAGAAgAAAAhAO7I
8/PhAAAACwEAAA8AAAAAAAAAAAAAAAAAIwUAAGRycy9kb3ducmV2LnhtbFBLAQItABQABgAIAAAA
IQB3kD4u1AAAAEgBAAAZAAAAAAAAAAAAAAAAADEGAABkcnMvX3JlbHMvZTJvRG9jLnhtbC5yZWxz
UEsFBgAAAAAFAAUAOgEAADwHAAAAAA==
" o:button="t" fillcolor="#00b050" stroked="f" strokeweight=".5pt">
<v:fill o:detectmouseclick="t"/>
<v:path arrowok="t"/>
<v:textbox>
<![if !mso]>
<table cellpadding=0 cellspacing=0 width="100%">
<tr>
<td><![endif]>

<i
style='mso-bidi-font-style:normal'><span style='font-size:18.0pt;
mso-bidi-font-size:11.0pt;color:windowtext;text-decoration:none;text-underline:
none'>Iniciar Sesión</i>

<![if !mso]></td>
</tr>
</table>
<![endif]></v:textbox>
</v:shape><![endif]--><![if !vml]><span style='mso-ignore:vglayout;position:
absolute;z-index:251660288;margin-left:365px;margin-top:447px;width:152px;
height:40px'><img border=0 width=152
height=40 src="index_archivos/image002.png"
alt="Cuadro de texto: Iniciar Sesión"
v:shapes="_x0033__x0020_Cuadro_x0020_de_x0020_texto"><![endif]><!--[if gte vml 1]><v:roundrect
id="_x0032__x0020_Rectángulo_x0020_redondeado" o:spid="_x0000_s1026"
href="http://12.0.0.1:3990/prelogin" style='position:absolute;margin-left:244.5pt;
margin-top:333pt;width:161.25pt;height:36pt;z-index:251659264;visibility:visible;
mso-wrap-style:square;mso-width-percent:0;mso-height-percent:0;
mso-wrap-distance-left:9pt;mso-wrap-distance-top:0;mso-wrap-distance-right:9pt;
mso-wrap-distance-bottom:0;mso-position-horizontal:absolute;
mso-position-horizontal-relative:text;mso-position-vertical:absolute;
mso-position-vertical-relative:text;mso-width-percent:0;mso-height-percent:0;
mso-width-relative:margin;mso-height-relative:margin;v-text-anchor:middle'
arcsize="10923f"
o:gfxdata="UEsDBBQABgAIAAAAIQC2gziS/gAAAOEBAAATAAAAW0NvbnRlbnRfVHlwZXNdLnhtbJSRQU7DMBBF
90jcwfIWJU67QAgl6YKUHSBUDjCyJ4lFMrZsE9rb46QVG0RZjv7/z9Pb5XYyIzsQRW25hnVZAUOr
0s5Y6ir5vt9lD1JwBDIwOMJKHpHlpr69KfdHjyxSmriSfYz+USnWPY7AufNIadK6MEJMx9ApD/oD
OlTrorhX2lFEilmcO2RdNtjC5xDF9pCuTyYBB5bi6bQ4syoJ3g9WQ0ymaiLzg5KdCXlKLjvcW893
SUOqXwnz5DrgnHtJTxOsQfEKIT7DmDSUCaxw7Rqn8787ZsmRM9e2VmPeBN4uqYvTtW7jvijg9N/y
JsXecLq0q+WD6m8AAAD//wMAUEsDBBQABgAIAAAAIQA4/SH/1gAAAJQBAAALAAAAX3JlbHMvLnJl
bHOkkMFqwzAMhu+DvoPRvnfSwRijTi+l0NvYy2fr2M4PBMnrbUb/Q94/lfHfsHYimAW0Y2/Q2fr2
JVI2sOt6UJgd+ZiDgffL8ekFlFSbvV0oo4EbhzGx4f9GRdb25HMsYhqlCwG5lrLq9biZkxWOiqY
22YiTra2kYMu1l1tQD30/bPm3wwYN0x18gb45AdQl1tp5j/sFB2T0FQ7R0nTNEV3j6o9feQzro1i
OWA14Fm+Q8a1a8+Bvu/d/dMb2JY5uiPbhG/ktn4cqGU/er3pcvwCAAD//wMAUEsDBBQABgAIAAAA
IQAy9ZUy0gIAACUGAAAOAAAAZHJzL2Uyb0RvYy54bWysVNtqGzEQfS/0H8Q+l9rr+BKTdTAOLoGQ
mCYlz7JW8gq0kiLJt359R9pd26SlhVI/rEeamTMzZ0Zzd3+sJdoz64RWBe5ddTFiiupSqG2Bv72u
Powxcp6okkitWIFPzOH7+ft3dwczY7mutCyZRQCi3OxgClx5b2ZZ5mjFauKutGEKlFzbmng42m1W
WnIA9Fpmebc7zA7alsZqypyD24dGiecRn3NG/TPnjnkkCwy5+fi18bsJ32x+R2ZbS0wlaJsG+Ycs
aiIUBD1DPRBP0M6KX6BqQa12mvsrqutMcy4oizVANb3um2peKmJYrAXIceZMk/t/sPRpv7ZIlAXO
RxgpUkOPVkJzNJ2MA3vZ2Ug+fzkwHDJL2Uyb0RvYy54bWysVNtqGzEQfS/0H8Q+l9rr+BKTdTAO
LoGQmCYlz7JW8gq0kiLJt359R9pd26SlhVI/rEeamTMzZ0Zzd3+sJdoz64RWBe5ddTFiiupSqG2B
v72uP

YpKUPMtarVdYq1El+da/6bf0xzrSXuwmhkLpyyLtzJnr0bm+2dWKbIR1EnROB2cpJUJzKKRe5vTb
y92nK0qcZ7pgCrTI6V44ejP++OF6azIxhApUISzBINplW5PTynuTJYnjlaiZOwMjNBpLsDXzeLXL
pLBsi9FrlQzT9CLZgi2MBS6cw7+3jZGOY/yyFNw/lqUTnqicYm0+fm38LsI3GV+zbGmZqSRvy2D/
UEXNpMakfahb5hlZW/kuVC25BQelP+NQJ1CWkovYA3YzSN9081wxI2IvOBxn+jG5/xeWP2zmlsgi
p0NKNKtxRUPyhGP79VMv1wqIFQXoQrACYn+Vkno1VZKv2mpwln/fWdPnLfB1LbRvFmeFYh5Z4ypp
HCU2C0XY+2IQdpJsjctibWGT8fhs5jZM2JkZ8JU7cmoswc21PrvS1sEX85JdXPa+X7bYecLx5zAd
XV5dnlPC0TY6v0Q2xcws69DGOv9VQE3CAWuDtS7CaOIg2GbmfCiCZZ1frA6ULO6kUvFil4upsmTD
AvvSL+l5l8Iduyn9Hhn4L3qs3zVTwe4PQMwckHFUTeNxBH6vRIin9JMocbOh1VhxfFOHmIxz3MVF
23P0DrASa++Bg1NA5btiWt8AE/Gt9cD0FPDPjD0iZgXte3AtNdhTAYpVn7nx77pveg7tL6DYI6Et
NC/dGX4ncXsz5vycWXzaKAIoV/4RP6WCbU6hPVFSgf1x6n/wR5ajlZItSkVO3fc1s4ISda/xLX4e
jEZBW+IlMgnpfGxZHFv0up4C8mGAwmh4PCLYetUdSwv1K6raJGRFE9Mcc+eUe9tdpr6RMNRFLiaT
6IZ6Ypif6WfDQ/Aw1UDMl90rs6alsEfyP0AnKyx7Q+LGNyA1TNYeShkZfphrO2/Uokj8VjeD2B3f
o9dB3ce/AQAA//8DAFBLAwQUAAYACAAAACEAtr4Ks+IAAAALAQAADwAAAGRycy9kb3ducmV2Lnht
bEyPzU7DMBCE70i8g7VIXBBC1w0+ahmyqggQHJKS2tHc3XpJAbEex24S3ZznBbVYzmv2mWE62Eyca
QusdQjJTIMhV3rSuRti9P19nIELUzujOO0L4pgDL8vys0Lnxo9vQaRtrwSUu5BqhibHPpQxVQ1aH
me/JsffhB6sjn0MtzaBHLredvFEqlVa3jj80uqenhqqv7dEiLPZr/VlNL+NOmde31eN6o676CfHy
Ylo9gIg0xb8w/OIzOpTMdPBHZ4LoEO6yBW+JCGmasuBEliT3IA4I89tMgSwL+X9D+QMAAP//AwBQ
SwMEFAAGAAgAAAAhAHeQPi7UAAAASAEAABkAAABkcnMvX3JlbHMvZTJvRG9jLnhtbC5yZWxzhM+x
agMxDADQPdB/MNp7vlyGUuJz2UIgW0lIP0DYOp/JWTaWE5q/j5YGAh01SO89DdtrXNWFigyJDay6
HgTKyYXgo4HP4/75FZRUOWEnIVuzjMP8/MNp79qVQmnC+LG0hQ5aSfoCxdT4Tn2xstyR/Hy0tDRSCJiHpSRp25yWKbyw1JNLQ
dwoEkk0ukNfweXx/fAFRmyFnYiLUcMEKu/FhNXxgNI2H6hxyFaxQ1TC3lrdSVjvjYmqXMhJXplQW
0zgtXmZjT8ajXCv1LMtfA8YbU+ydhrJ3PYjjJfPm+3aapmDxNdmvBan9s0LOLJUY6MSoKR7bL9uv
O8XRb582GyUzX5Z8oJ+2Q3J8wdu5YSETQY6DvPl/vAIAAP//AwBQSwECLQAUAAYACAAAACEAtoM4
kv4AAADhAQAAEwAAAAAAAAAAAAAAAAAAAAAAW0NvbnRlbnRfVHlwZXNdLnhtbFBLAQItABQABgAI
AAAAIQA4/SH/1gAAAJQBAAALAAAAAAAAAAAAAAAAAC8BAABfcmVscy8ucmVsc1BLAQItABQABgAI
AAAAIQAy9ZUy0gIAACUGAAAOAAAAAAAAAAAAAAAAAC4CAABkcnMvZTJvRG9jLnhtbFBLAQItABQA
BgAIAAAAIQC2vgqz4gAAAAsBAAAPAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
LQAUAAYACAAAACEAd5A+LtQAAABIAQAAGQAAAAAAAAAAAAAAAAAA7BgAAZHJzL19yZWxzL2Uyb0Rv
Yy54bWwucmVsc1BLBQYAAAAABQAFADoBAABGBwAAAA=

" o:button="t" fillcolor="#00b050" strokecolor="black [3213]" strokeweight="2pt">
 <v:fill o:detectmouseclick="t"/>
 <v:path arrowok="t"/>
</v:roundrect><![endif]--><![if !vml]><span style='mso-ignore:vglayout;
position:absolute;z-index:251659264;margin-left:324px;margin-top:442px;
width:219px;height:52px'><map name=MicrosoftOfficeMap0><area shape=Polygon
coords="10, 2, 2, 11, 2, 43, 10, 50, 210, 50, 217, 43, 217, 11, 210, 2, 10, 2"
href="http://12.0.0.1:3990/prelogin"></map><img border=0 width=219 height=52
src="index_archivos/image003.png" usemap="#MicrosoftOfficeMap0"
v:shapes="_x0032__x0020_Rectángulo_x0020_redondeado"><![endif]><span
style='mso-fareast-language:ES;mso-no-proof:yes'><!--[if gte vml 1]><v:shapetype
 id="_x0000_t75" coordsize="21600,21600" o:spt="75" o:preferrelative="t"
 path="m@4@5l@4@11@9@11@9@5xe" filled="f" stroked="f">
 <v:stroke joinstyle="miter"/>
 <v:formulas>
  <v:f eqn="if lineDrawn pixelLineWidth 0"/>
  <v:f eqn="sum @0 1 0"/>
  <v:f eqn="sum 0 0 @1"/>
  <v:f eqn="prod @2 1 2"/>
  <v:f eqn="prod @3 21600 pixelWidth"/>
  <v:f eqn="prod @3 21600 pixelHeight"/>
  <v:f eqn="sum @0 0 1"/>
  <v:f eqn="prod @6 1 2"/>
  <v:f eqn="prod @7 21600 pixelWidth"/>
  <v:f eqn="sum @8 21600 0"/>
  <v:f eqn="prod @7 21600 pixelHeight"/>
  <v:f eqn="sum @10 21600 0"/>
 </v:formulas>
 <v:path o:extrusionok="f" gradientshapeok="t" o:connecttype="rect"/>
 <o:lock v:ext="edit" aspectratio="t"/>
</v:shapetype><v:shape id="_x0030__x0020_Imagen" o:spid="_x0000_i1025" type="#_x0000_t75"
 style='width:425.25pt;height:379.5pt;visibility:visible;mso-wrap-style:square'>
 <v:imagedata src="index_archivos/image005.png" o:title=""/>
</v:shape><![endif]--><![if !vml]><img width=567 height=506
src="index_archivos/image006.jpg" v:shapes="_x0030__x0020_Imagen"><![endif]></p>

```
</div>
</body>
</html>
```

Anexo 9. Código fuente del módulo de auto-servicio.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/mman.h>
#include <unistd.h>
#include <pthread.h>
#include </home/pi/maquina_expedidora/font.h>
volatile unsigned *gpio;
// GPIO setup macros. Always use INP_GPIO(x) before using OUT_GPIO(x) or
SET_GPIO_ALT(x,y)
#define BCM2708_PERI_BASE  0x20000000
#define GPIO_BASE  (BCM2708_PERI_BASE + 0x200000) /* GPIO controller */
#define INP_GPIO(g) *(gpio+((g)/10)) &= ~(7<<(((g)%10)*3))
#define OUT_GPIO(g) *(gpio+((g)/10)) |=  (1<<(((g)%10)*3))
#define SET_GPIO_ALT(g,a) *(gpio+(((g)/10))) |= (((a)<=3?(a)+4:(a)==4?3:2)<<(((g)%10)*3))
#define GPIO_SET *(gpio+7)  // sets   bits which are 1 ignores bits which are 0
#define GPIO_CLR *(gpio+10) // clears bits which are 1 ignores bits which are 0
#define GET_GPIO(g) (*(gpio+13)&(1<<g)) // 0 if LOW, (1<<g) if HIGH
#define GPIO_PULL *(gpio+37) // Pull up/pull down
#define GPIO_PULLCLK0 *(gpio+38) // Pull up/pull down clock
#define PAGE_SIZE (4*1024)
#define BLOCK_SIZE (4*1024)
#define IN_PORT(a) *(gpio+((a)/10))&=~(7<<(((a)%10)*3))
#define OUT_PORT(a) *(gpio+((a)/10))|=(1<<(((a)%10)*3))

#define SET_PORT(a) *(gpio+7)=0x01<<a
#define CLR_PORT(a) *(gpio+10)=0x01<<a

#define RSTNC_PORT(a) *(gpio+37)=0x01<<a
#define READ_PORT(a) (*(gpio+13)&(1<<a))?1:0
#define PUERTO_MODO(b,a)
(a==1)?(*(gpio+((b)/10))|=(1<<(((b)%10)*3))):(*(gpio+((b)/10))&=~(7<<(((b)%10)*3)))

#define DELAY for(k=0;k<100;k++)
#define RST 18
#define CE 15
#define DC 4
#define SDA 2
#define SCLK 3
#define LIGHT 14

int  mem_fd;
void *gpio_map;
void setup_io();
```

```
void enviar_byte(unsigned char dato);
void enviar_commando(unsigned char comando);
void enviar_dato(unsigned char dato);
void reset_lcd(void);
void config_lcd(void);
void CLR_lcd(void);
void inicializar_puertos(void);
void inicializar_lcd(void);
void configuraciones_de_inicio(void);

void ponerpixel(volatile unsigned char X, volatile unsigned char Y);
void borrapixel(volatile unsigned char X, volatile unsigned char Y);
void cambiapixel(volatile unsigned char X, volatile unsigned char Y);
unsigned char test_pixel(volatile unsigned char X, volatile unsigned char Y);
void mover_pixel(unsigned char xi, unsigned char yi, unsigned char xf, unsigned char yf);
void deslizar_imagen(unsigned char xi, unsigned char  yi, unsigned char  n, unsigned char  m,
unsigned char k, unsigned char modo);

void imprimir_mensaje(unsigned char mensaje[], unsigned char x, unsigned char y);
void cargar_imagen(unsigned char imagen[], unsigned char posX, unsigned char posY);

void cargar_logo(unsigned char imagen[]);
void imprimir_logo(void);
void animacion_izquierda_logo(void);
void animacion_derecha_logo(void);
void ejecutar_cmd(unsigned char cmd[]);

unsigned char verificar_ip(void);
unsigned char probar_conexion(void);
unsigned char estado_de_la_red(void);
void conectando_a_la_red(void);

unsigned char solicitar_clave(unsigned char planX[]);
void pantalla_inicio(unsigned char imagen[], unsigned char x, unsigned char y);
void solicitando_autorizacion(unsigned char imagen[], unsigned char x, unsigned char y);
void autorizacion_aceptada(unsigned char imagen[], unsigned char x, unsigned char y);
void pago_recibido(unsigned char imagen[], unsigned char x, unsigned char y);
void imprimiendo(unsigned char imagen[], unsigned char x, unsigned char y);
void recoja_su_recibo(unsigned char imagen[], unsigned char x, unsigned char y);
void detalles_recibo(void);
void fin(void);

void pantalla1(void);
void pantalla2(void);
void pantalla3(void);
void pantalla4(void);
void pantalla5(void);
void pantalla6(void);

void limpiar_RAM(void);
void actualizar_pantalla(void);
void pantalla_CPU(void);
```

```
void led_rojo(void);
void esperando_usuario(void);
void coin_selector(void);
void botonA(void);
void botonB(void);
void boton_reinicio(void);
void boton_apagado(void);
void animacion_de_espera(void);

unsigned char numero_decimal_a_numero_ascii(unsigned char n);
unsigned char numero_ascii_a_numero_decimal(unsigned char n);
unsigned char control_de_interfaz(unsigned char *instruccion);

unsigned char RAM_VIDEO[504];
unsigned char logo[85][48], logo_original[85][48];
unsigned char salida_cmd[50][512];
unsigned char interfaz[10];
unsigned char ipservidor[20];
unsigned char checar_ip[100];
unsigned char numero_pings;
unsigned char intervalo_pings;
unsigned char probar_red[200];
unsigned char estado_conexion;
unsigned char num_planes;
unsigned char planes[10][30];
unsigned char folio[10];
unsigned char clave[12];
unsigned char vigencia_fecha[20];
unsigned char vigencia_hora[20];
unsigned char salidaCPU;
unsigned char *pntr0, *pntr1, *pntr2, *pntr3 ;
unsigned char *plan[10];
unsigned char monedas;
unsigned char apagado;
unsigned char boton_a;
unsigned char boton_b;
unsigned char estado;
unsigned char var_tmps[30];
unsigned char opcion_animacion_logo;
unsigned char opcion_animacion_de_espera;

pthread_t hilo[20];
int pid_hilos[20];


#define INTERFAZ pntr0
#define IPSERVIDOR pntr1
#define NUMERO_PINGS numero_pings
#define INTERVALO_PINGS intervalo_pings

#define BTN_A 10
```

```
#define PID_BTN_A 11
#define BTN_B 25
#define PID_BTN_B 12
#define REINICIO 9
#define PID_REINICIO 15
#define APAGADO 11
#define PID_APAGADO 16
#define ENTRADA_COIN_SELECTOR 22
#define COIN_SELECTOR 10 //numero para el PID
#define LED1 23 //verde
#define LED2 24 //rojo
#define LED_VERDE 23
#define LED_ROJO 24

#define BTN_C x
#define BTN_D x
#define BTN_E x

#define ARRIBA 0
#define ABAJO 1
#define DERECHA 2
#define IZQUIERDA 3

#define FOLDER_IMAGENES "/home/pi/maquina_expedidora/imagenes_pantallita/"

enum estados
{
	configurar_pines, //0
	conf_iniciales, //1
	iniciar_subprocesos, //2
	terminar_operaciones, //3
	lanzar_subprocesos, //4
	imprimirl0go, //5
	apagar_cpu, //6
	iniciar_venta, //7
	esperando_monedas, //8
	imprimir_pantalla1, //9
	imprimir_pantalla2, //10
	imprimir_pantalla3, //11
	imprimir_pantalla4, //12
	imprimir_pantalla5, //13
	imprimir_pantalla6, //14
	esperando_al_usuario, //15
	esperando_boton_de_comprar, //16
	pago_realizado, //17
	solicitar_activacion_de_clave, //18
	imprimiendo_clave_en_pantalla, //19
	inactivo, //20
	iniciar_test_red, //21
	terminar_y_reiniciar_proceso_de_compra, //22
	esperando_terminacion_de_compra, //23
	esperar_a_que_se_tenga_conexion, //24
```

```
	verificar_si_hay_conexion, //25
};

unsigned char tecladox, tecladoy;
pthread_t hiloteclado;

void tecladoxy(void)
{
	unsigned char t;
	while(1)
	{
		t=getchar();
//		if(t=='w') tecladoy++;
//		if(t=='s') tecladoy--;
//		if(t=='a') tecladox--;
//		if(t=='d') tecladox++;
		if(t=='c') boton_a++;
		if(t=='m') monedas++;
		if(t=='t') estado=apagar_cpu;
//		printf("t= %c, x = %i, y = %i\n", t, tecladox,tecladoy);
	};
};

void main(unsigned char nargs, char *args[])
{

unsigned int i, j, k;
//unsigned char t;
unsigned char costo_de_plan;
unsigned char plantmp[30];
unsigned char vartmp0=0; //se usa para evitar que se inicie dos veces el mismo hilo con el
mismo subproceso

pthread_t hilo_estado_de_la_red;
pthread_t hilo_conectando_a_la_red;
pthread_t hilo_de_animacion_espera;
pthread_t hilo_esperando_usuario;


INTERFAZ="wlan0";
IPSERVIDOR="192.168.182.1";
//IPSERVIDOR="10.40.27.254";
NUMERO_PINGS=2; //Éste número debe de estar entre 1 y 8
INTERVALO_PINGS=4; //Número en segundos. Es el tiempo que hay entre ping y ping

salidaCPU=0;
apagado=0;
estado_conexion=0;
boton_a=0;
boton_b=0;
estado=configurar_pines;
```

```
monedas=0; //variable que almacena las monedas
k=0; //se usa para guardar las monedas introducidas y compararlas posteriormente.
opcion_animacion_logo=1; //para que el logo de pueda animarse
tecladox=0, tecladoy=0; //solo para pruebas

pthread_create(&hiloteclado,NULL,(void *)tecladoxy, NULL);

setup_io();
configuraciones_de_inicio();

/*
//printf("%s\n",probar_red);
while(1)
{
        //printf("%i\n",0);
//      probar_conexion();
        printf("-------\n");
        ejecutar_cmd("ifconfig wlan0 | cat -n | tail -1 | awk {'print $1'}");
        i=salida_cmd[0][0]-48;
        switch(i)
        {
                case 7: printf("wlan0 apagada\n"); break;
                case 8: printf("wlan0 iniciando\n"); break;
                case 9: printf("wlan0 ip ok\n"); break;
        };


/*
        for(i=0;i<20;i++)
        {
                if(strcmp(salida_cmd[i],"   ")) printf("i=%i, %s\n",i,salida_cmd[i]);
        };
                for(i=0;i<50;i++)
                        for(j=0;j<512;j++)
                                salida_cmd[i][j]=0;

        sleep(INTERVALO_PINGS);
}
//return;
*/

estado=lanzar_subprocesos;
//estado=apagar_cpu;

while(estado!=apagar_cpu)
{
        switch(estado)
        {
//              case configurar_pines: setup_io(); estado=conf_iniciales; break;

//              case conf_iniciales: configuraciones_de_inicio(); estado=lanzar_subprocesos;
break;
```

```
/*

                case imprimirl0go: imprimir_logo(); estado=lanzar_subprocesos; break;
                case conf_iniciales: configuraciones_de_inicio(); estado=imprimirl0go; break;
                case imprimirl0go: imprimir_logo(); estado=lanzar_subprocesos; break;
*/
                case lanzar_subprocesos:
//                      sleep(1);
                        pthread_create(&hilo_estado_de_la_red,NULL,(void
*)estado_de_la_red,NULL);
                        pid_hilos[PID_BTN_A]=pthread_create(&hilo[PID_BTN_A],NULL,(void
*)botonA, NULL);
                        pid_hilos[PID_BTN_B]=pthread_create(&hilo[PID_BTN_B],NULL,(void
*)botonB, NULL);

        pid_hilos[COIN_SELECTOR]=pthread_create(&hilo[COIN_SELECTOR],NULL,(void
*)coin_selector, NULL);
                        pthread_create(&hilo_esperando_usuario,NULL,(void
*)esperando_usuario,NULL);
//                      SET_PORT(LED_VERDE);
                        estado=verificar_si_hay_conexion;
                        break;

                case verificar_si_hay_conexion:
                        if(estado_conexion) estado=esperando_monedas;
                        else estado=esperar_a_que_se_tenga_conexion;
                break;

                case esperando_monedas:
//                      usleep(1000);
//                      printf("entrando a esperando_monedas\n");
                        switch(monedas)
                        {
                                case 0:
                                        boton_a=0;
                                        boton_b=0;
                                        estado=verificar_si_hay_conexion;
//                                      printf("No Hay monedas\n");
                                break;
                                case 1:
                                        opcion_animacion_logo=0;
//                                      printf("animacion = %i\n",0);
                                        estado=iniciar_venta;
                                break;
                        };
                        break;

                case iniciar_venta:
//                      printf("entrando a iniciar_venta\n");
                        pthread_join(hilo_esperando_usuario,NULL);
                        estado=imprimir_pantalla1;
                        break;
```

```
		case imprimir_pantalla1:
			pantalla1();
			sleep(2);
			boton_a=0;
			estado=imprimir_pantalla2;
			break;

		case imprimir_pantalla2:
			pantalla2();
			k=monedas;
			estado=esperando_al_usuario;
			break;

		case imprimir_pantalla3:
			pantalla3();
			estado=esperando_al_usuario;
			break;

		case imprimir_pantalla4:
			pantalla4();
			estado=esperando_al_usuario;
			break;


		case imprimir_pantalla5:
			pantalla5();
			estado=esperando_al_usuario;
			break;


		case imprimir_pantalla6:
			pantalla6();
			estado=esperando_al_usuario;
			break;

		case esperando_al_usuario:
//			printf("Hay %i pesos\n", 5*k);
			if(k!=monedas) //Se revisa si metio mas de una moneda. Si no no se
hace nada de esto y se
			//pasa a esperar el boton de comprar.
			{
				switch(monedas)
				{
//					case 1: estado=imprimir_pantalla1; break; //2horas $5
1*5=5
					case 2: estado=imprimir_pantalla3; break; //1dia $10
2*5=10
					case 4: estado=imprimir_pantalla4; break; //7dias $20
4*5=20
					case 6: estado=imprimir_pantalla5; break; //15dias $30
6*5=30
```

```
                                        case 10: estado=imprimir_pantalla6; break; //30dias
$50 10*5=50
                                };
                                k=monedas;
//                              printf("Hay %i monedas\n", k);
                        };
//                      usleep(100000);
                        if(boton_a) //se revisa que se haya presionado el boton a, hasta que se
presione se sale de
                                                        //esperando_usuario
                        {
                                estado=pago_realizado;
//                              boton_a=0;
                        };

                        break;
                case esperando_boton_de_comprar: break;
                case pago_realizado:


                        costo_de_plan=5*monedas;
                        switch(costo_de_plan)
                        {
                                case 5: strcpy(plantmp,"gacFecDeisjidjutsAkP.php"); break;
//2horas
                                case 10: strcpy(plantmp,"GhovMyWoabgeidryodEt.php");
break; //1dia
                                case 20: strcpy(plantmp,"LophnetnoadKawbarEj9.php"); break;
//7dias
                                case 30: strcpy(plantmp,"muBlarefyeawdOdOjfij.php"); break;
//15dias
                                case 50: strcpy(plantmp,"UshaccyHepciwrefErs1.php"); break;
//30dias
                        };
                        estado=solicitar_activacion_de_clave;
                        break;

                case solicitar_activacion_de_clave:
                        opcion_animacion_de_espera=0; //se enciende la animacion de espera
                        pthread_create(&hilo_de_animacion_espera,NULL,(void
*)animacion_de_espera, NULL); //se lanza
                        //el subproceso de animacion de espere
                        sleep(5); //se espera por 5 seegundos, es el tiempo aproximado que
tarda la informacion en
                        //imprimirse en el boleto, osea los detalles del recibo
                        opcion_animacion_de_espera=solicitar_clave(plantmp);//se solicita la
activacion de la clave
                        estado=imprimiendo_clave_en_pantalla;
                        break;

                case imprimiendo_clave_en_pantalla:
```

```
                        pthread_join(hilo_de_animacion_espera, NULL); //se espera que
termine la animacion de espera
                        detalles_recibo(); //se imprimen detalles del recibo
                        boton_a=0; //para que se pueda esperar al usuario hasta que presione
el boton y evitar que
                        //no se presenten los detalles del recibo prematuramente
                        estado=esperando_terminacion_de_compra;
                        break;

                case esperando_terminacion_de_compra:

                        if(boton_a)       //se revisa si se presiono el boton a, si se presionó se
termina la comprar si no
                        // se espera hasta que se presione.
                        {
                                estado=terminar_y_reiniciar_proceso_de_compra;
                        }
                        else
                        {
                                estado=esperando_terminacion_de_compra;
                        };
                        break;
                case terminar_y_reiniciar_proceso_de_compra:
                        monedas=0; //monedas a cero para nueva compra
                        opcion_animacion_logo=1; // para que se pueda ejecutar la animacion
del logo
                        imprimir_logo(); // se imprime el logo
                        sleep(1); //se presenta el logo por un segundo
                        pthread_create(&hilo_esperando_usuario,NULL,(void
*)esperando_usuario,NULL);  //se inicia la
                        //la animación del logo
                        estado=verificar_si_hay_conexion;
                        break;

                case inactivo:
                        usleep(250000);
                        break;

                case terminar_operaciones:
                        estado=apagar_cpu;
                        break;

                case esperar_a_que_se_tenga_conexion:
//                      printf("%i\n",estado_conexion);
                        if(!estado_conexion)
                        {
                                if(!vartmp0) //Ya lanze el hilo_conectando_a_la_red? si no lo
lanzo, esto impide que se vuelva
                                //a lanzar creando un conflicto entre procesos
                                {
                                        opcion_animacion_logo=0; //variable global con la que
se le indica a la animacion del logo
```

```
                                        //que se apague
                                        pthread_join(hilo_esperando_usuario,NULL); //se
espera a que se termine el proceso
                                        //de animacion del logo, esto evita cosas raras en la
pantalla.
//                                      printf("Lanzando hilo_conectando_a_la_red\n");

        pthread_create(&hilo_conectando_a_la_red,NULL,(void
*)conectando_a_la_red,NULL); //se
                                        //lanza el subproceso de la animacion de espera
                                        vartmp0=1; //con esta variable se conoce si ya se lanzo
el proceso y animacion de
                                        //conectando a la red.
                                        CLR_PORT(LED_VERDE); //se apaga el led verde.
                                };
                                estado=esperar_a_que_se_tenga_conexion;
                        }
                        else
                        {
                                if(vartmp0) //Si no ya lanze el hilo_conectando_a_la_red
espero a que termine
                                {
//                                      printf("Esperando a que termine
hilo_conectando_a_la_red\n");
                                        pthread_join(hilo_conectando_a_la_red,NULL); //
espera a que termine el proceso de animacion
                                        //de conectando a ala red
                                        vartmp0=0;//se pone en cero para que la proxima vez
pueda lanzarse el subproceso de animacion
                                        //de conectando a ala red
                                        SET_PORT(LED_VERDE); //se enciende el led verde
                                        opcion_animacion_logo=1; //se activa la animacion del
logo
                                        pthread_create(&hilo_esperando_usuario,NULL,(void
*)esperando_usuario,NULL); //se lanza la
                                        //animacion del logo en segundo plano.
                                        monedas=0; //para q al regresar pueda volver a cobrar
ya que sin esta linea no puede
                                        //iniciar una nueva venta en dado caso de que el
momento que esten metiendo una moneda
                                        //la red se haya caido.
                                        estado=esperando_monedas;
                                };
                        };
                        break;
//              case apagar_cpu: break;
        };
//      printf("edo=%i\n",estado); //solo para saber en que estado se encuentra la máquina
        usleep(250000); //tiempo de transición entre los estados de la máquina de estados.
}
//pthread_create(&hilo[PID_APAGADO],NULL,(void *)boton_apagado,NULL);
//pthread_create(&hilo[PID_REINICIO],NULL,(void *)boton_reinicio,NULL);
```

```
/**********Acciones que se ejecutan antes de apagarse o reiniciarse**********/
/////usleep(250000);
/////limpiar_RAM();
/////switch(apagado)
/////{
/////     case 1:  imprimir_mensaje("[Reiniciando]",0,0); break;
/////     case 2: imprimir_mensaje("[Apagando]",0,0); break;
/////};
/////actualizar_pantalla();
/////sleep(1);
/////limpiar_RAM();
/////actualizar_pantalla();
/////usleep(250000);
/////CLR_PORT(LED1);
/////CLR_PORT(LED2);
/////reset_lcd();
/////SET_PORT(LIGHT);
/////switch(apagado)
/////{
/////     case 1: ejecutar_cmd("sleep 1; reboot;"); break;
/////     case 2: ejecutar_cmd("sleep 1; poweroff;"); break;
/////};
/*****************************************************************************/
}

void animacion_de_espera(void)
{
        limpiar_RAM();
        cargar_imagen("espere.lcd",0,0);
        while(!opcion_animacion_de_espera)
        {
                deslizar_imagen(4,6,75,13,5,2);
                usleep(100000);
        };
}

void solicitando_autorizacion(unsigned char imagen[], unsigned char x, unsigned char y)
{
        limpiar_RAM();
        imprimir_mensaje("Espere", 9, 40);
//      imprimir_mensaje("autorizacion", 5, 32);
//      cargar_imagen(imagen,x,y);
//
        cargar_imagen("/home/juliocesar/Escritorio/imagenes_pantallita/solicitandoclave.lcd"
, 18, 2);
        actualizar_pantalla();
}

void botonA(void)
{
        unsigned int i=0;
```

```
	while(estado!=terminar_operaciones)
	{
		while(!READ_PORT(BTN_A)) usleep(2000);
		while(READ_PORT(BTN_A))
		{
			i++;
			usleep(1000);
		};
//		printf("i=%i\n", i);
			if(i>75)
			{
				boton_a++;
		//		printf("i=%i\n", boton_a);
//				printf("BotonA\n");
			}
		i=0;
	};
}

void botonB(void)
{
	unsigned int i=0;
	while(estado!=terminar_operaciones)
	{
		while(!READ_PORT(BTN_B)) usleep(2000);
		while(READ_PORT(BTN_B))
		{
			i++;
			usleep(1000);
		};
//		printf("i=%i\n", i);
//			if(i>75) boton_b++;
			if(i>75)
			{
//				SET_PORT(8);
				usleep(100000);
//				CLR_PORT(8);
//				printf("BotonB\n");
			};
		i=0;
	};
}

void coin_selector(void)
{
	unsigned int i=0;
	while(estado!=terminar_operaciones)
	{
		while(!READ_PORT(ENTRADA_COIN_SELECTOR))
		{
			usleep(1000); //para el coin selector
//			usleep(2000); //para emulación con un push boton
```

```
                }

                while(READ_PORT(ENTRADA_COIN_SELECTOR))
                {
                        i++;
                        usleep(1000);
                };
//              printf("i=%i\n", i);
//              if(i>75) //para emulación con un push boton
                if((i>25)&&(i<100)) //para el coin selector
                {
                        monedas++;
//                      printf("Hay %i monedas\n", monedas);
                };
                i=0;
        };
}

void esperando_usuario(void)
{
//      printf("entrando a esperando usuario\n");
        unsigned int retardo=200000;
        unsigned char paso=2;
        unsigned char i, j, k;
        imprimir_logo();
                while(opcion_animacion_logo)
                {
                        for(j=0;j<2;j++)
                        {
                                for(i=0;(opcion_animacion_logo)&&(i<48);i++)
                                {
                                        deslizar_imagen(0,0,84,48,paso,2);
                                        usleep(retardo);
                                };
                        };


                        for(j=0;j<2;j++)
                        {
                                for(i=0;(opcion_animacion_logo)&&(i<48);i++)
                                {
                                        deslizar_imagen(0,0,84,48,paso,3);
                                        usleep(retardo);
                                };
                        };

                        for(j=0;j<2;j++)
                        {
                                for(i=0;(opcion_animacion_logo)&&(i<24);i++)
                                {
                                        deslizar_imagen(0,0,84,48,paso,0);
                                        usleep(retardo);
```

```
                                        };
                                };

                                for(j=0;j<2;j++)
                                {
                                        for(i=0;(opcion_animacion_logo)&&(i<24);i++)
                                        {
                                                deslizar_imagen(0,0,84,48,paso,1);
                                                usleep(retardo);
                                        };
                                };
                        };
//printf("saliendo de esperando usuarop\n");
}

void conectando_a_la_red(void)
{
        limpiar_RAM();
        cargar_imagen("conectando_a_la_red1.lcd",0,0);
        actualizar_pantalla();
        unsigned char i;
        pthread_t hilo_led_rojo;
        pthread_create(&hilo_led_rojo,NULL,(void *)led_rojo,NULL);
        while(!estado_conexion)
        {
                for(i=0;(i<9)&&(!estado_conexion);i++)
                {
                        usleep(200000);
                        deslizar_imagen(4,9,75,8,5,2);
                };

                for(i=0;(i<9)&&(!estado_conexion);i++)
                {
                        usleep(200000);
                        deslizar_imagen(4,9,75,8,5,3);
                };
        };
}

unsigned char probar_conexion(void)
{
        //funcion pendiente implementar submuestreo /***********/
        //esta funcion hace ping a la IP del hostpot o de la estacion PSII

        unsigned char i, x;
//      printf("%s",probar_red); //solo para pruebas
//      printf("iniciando_pruebas\n"); //solo para pruebas
        ejecutar_cmd(probar_red);
//      ejecutar_cmd("ping -W 2 -c 5 10.40.27.254 | awk {'print $1'};"); //solo como ejemplo
//      printf("ok\n"); //solo para pruebas
// numero_pings=5; //solo para pruebas
        for(i=0, x=0; i<(numero_pings+4); i++)
```

```
	{
//			printf("entrando\n"); //solo para pruebas
//			printf("%c%c\n",salida_cmd[i*4][0],salida_cmd[i*4][1]); //solo para pruebas
//			printf("i = %i, %s\n",i,salida_cmd[i]); //solo para pruebas
//			printf("%c%c\n",salida_cmd[i][0],salida_cmd[i][1]); //solo para pruebas
			if(salida_cmd[i][0]=='6' && salida_cmd[i][1]=='4') x++;
//			else x--;
	};
//	printf("----\n"); //solo para pruebas
	for(i=0;i<50;i++) strcpy(salida_cmd[i],"     ");
	if((x>0) && (x<=(numero_pings))) i=1;
	else i=0;
//	printf("%i\n",x); //solo para pruebas
	return i;
}

unsigned char verificar_ip(void)
{
	//Esta funcion verifica que wlan0 tenga solo una IP cualquiera. Lo importante es
determinar si
	//wlan0 esta encendida y se pudo conectar a la estacion PSII.
	unsigned char i, x;
	ejecutar_cmd(checar_ip);

	for(i=0, x=5;(salida_cmd[0][i])&&(i<18);i++)
	{
		switch(salida_cmd[0][i])
		{
			case '0': x++; break;
			case '1': x++; break;
			case '2': x++; break;
			case '3': x++; break;
			case '4': x++; break;
			case '5': x++; break;
			case '6': x++; break;
			case '7': x++; break;
			case '8': x++; break;
			case '9': x++; break;
			default: x--; break;
		};

//		printf("i=%i %c\n", i, salida_cmd[0][i]);
	};
//	printf("\n\nx=%i\n\n",x);
	if((x>6) && (x<18)) i=1; // comprueba que la interfaz tenga una IP
	else i=0;
	return i;
}

unsigned char estado_de_la_red(void)
{
	unsigned char i;
```

```
	while(!apagado)
	{
		i=control_de_interfaz("ifconfig wlan0 | cat -n | tail -1 | awk {'print $1'}");
//devuelve
		//el estado de la interfaz
		switch(i)
		{
			case 0: //interfaz apagada.
//				printf("conexion 0\n"); //solo para pruebas
				estado_conexion=0;
				control_de_interfaz("ifconfig wlan0 down; sleep 3; ifconfig
wlan0 up;"); //se reinicia
				sleep(10);
			break;
			case 1:
//				printf("conexion 1\n"); //solo para pruebas
				sleep(10); //esperar a que obtenga un ip
			break;
			case 2: //se tiene IP, se procede a verificar la IP
//				printf("conexion 2\n"); //solo para pruebas
				if(verificar_ip())
					if(probar_conexion()) estado_conexion=1; //se
procede a hacer ping a la
					//estación PSII.
					else estado_conexion=0;
				sleep(INTERVALO_PINGS);
			break;
		};
	};
}

unsigned char control_de_interfaz(unsigned char *instruccion)
{
	unsigned char i=0;
	unsigned char salida_terminal0[5][5];
	FILE *tuberia_terminal0=popen(instruccion, "r");

	if(!tuberia_terminal0)
	{
		fprintf(stderr, "incorrect parameters or too many files.\n");
		return;
	};
	while(!feof(tuberia_terminal0))
	{
		fscanf(tuberia_terminal0,"%[^\n]\n",salida_terminal0[i++]);
//		printf("i=%i, -> %s\n", i-1, salida_terminal0[i-1]);
	};
	pclose(tuberia_terminal0);

	switch(salida_terminal0[0][0])
	{
		case '7': i=0; break; //está apagada.
```

```
                case '8': i=1; break; //está iniciando.
                case '9': i=2; break; //ya está activa.
        }
        return i;
}

void boton_reinicio(void)
{
        unsigned char i=0;
        while(estado!=terminar_operaciones)
        {
                while((!READ_PORT(REINICIO))&&(!READ_PORT(APAGADO))) usleep(100000);
                while(READ_PORT(REINICIO)&&(i<10))
                {
                        usleep(100000);
                        i++;
                };
//              printf("i=%i\n",i);
//              printf("Reiniciando\n",i);
                if(i>5) apagado=1;
                else    i=0;
        };
}

void boton_apagado(void)
{
        unsigned char i=0;
        while(estado!=terminar_operaciones)
        {
                while((!READ_PORT(REINICIO))&&(!READ_PORT(APAGADO))) usleep(100000);
                while(READ_PORT(APAGADO)&&(i<40))
                {
                        usleep(100000);
                        i++;
                };
//              printf("i=%i\n",i);
//              printf("Apagando\n",i);
                if(i>30) apagado=2;
                else    i=0;
        };
}

void led_rojo(void)
{
        while(!estado_conexion)
        {
                if(!estado_conexion)
                {
                        SET_PORT(LED_ROJO);
                        usleep(500000);
                };
                if(!estado_conexion)
```

```
		{
			CLR_PORT(LED_ROJO);
			usleep(500000);
		};
	};
	CLR_PORT(LED_ROJO);
}

void configuraciones_de_inicio(void)
{
	inicializar_puertos();
	OUT_GPIO(LED1); //OPC
	OUT_GPIO(LED2);//OPC
	CLR_PORT(LED_VERDE);//OPC
	CLR_PORT(LED_ROJO);//OPC
	inicializar_lcd();


	unsigned int i, j, k;
	unsigned char x, y, t;

	for(i=0;i<50;i++)
		for(j=0;j<512;j++)
			salida_cmd[i][j]=0x00;
	t=0; estado_conexion=0;

	limpiar_RAM();
	cargar_logo("logofie.lcd");

	strcpy(interfaz,pntr0);
	strcpy(ipservidor,pntr1);

//	for(i=0;i<NPLANES;i++) strcpy(planes[i],plan[i]);
//	printf("%s\n",FOLDER_IMAGENES);
////revisar aqui por que se atora
	strcpy(checar_ip,"ip addr show "); //se revisa
	strcat(checar_ip,interfaz); // si se
	strcat(checar_ip," | grep inet | head -1 | awk {'print $2'}"); //tiene una IP

	//ping -W 2 -c 2 10.40.27.254 | awk {'print $1'}; //formato como debe de ejecutarse
	//el cmd ping
	//-W tiempo en segundos maximo que debe esperar por el ping
	//para que no cuelge el programa

	strcpy(probar_red,"ping -W 2 -c ");
	x=numero_decimal_a_numero_ascii(numero_pings);
	strcat(probar_red,&x);
	strcat(probar_red," ");
	strcat(probar_red,ipservidor);
	strcat(probar_red," | awk {'print $1'};");
}
```

```
void pantalla1(void)
{
	limpiar_RAM();
	imprimir_mensaje("Bienvenido a",6,25);
	imprimir_mensaje("**e-Monarca**",3,15);
	actualizar_pantalla();
};
void pantalla2(void)
{
	limpiar_RAM();
	imprimir_mensaje("Para comprar",8,40);
	imprimir_mensaje("presione el",11,31);
	imprimir_mensaje("boton",27,22);
	imprimir_mensaje("* 2 horas $5",0,7);
	actualizar_pantalla();
};
void pantalla3(void)
{
	limpiar_RAM();
	imprimir_mensaje("Para comprar",8,40);
	imprimir_mensaje("presione el",11,31);
	imprimir_mensaje("boton",27,22);
	imprimir_mensaje("* 1 dia $10",0,7);
	actualizar_pantalla();
};
void pantalla4(void)
{
	limpiar_RAM();
	imprimir_mensaje("Para comprar",8,40);
	imprimir_mensaje("presione el",11,31);
	imprimir_mensaje("boton",27,22);
	imprimir_mensaje("* 7 dias $20",0,7);
	actualizar_pantalla();
};
void pantalla5(void)
{
	limpiar_RAM();
	imprimir_mensaje("Para comprar",8,40);
	imprimir_mensaje("presione el",11,31);
	imprimir_mensaje("boton",27,22);
	imprimir_mensaje("* 15 dias $30",0,7);
	actualizar_pantalla();
};
void pantalla6(void)
{
	limpiar_RAM();
	imprimir_mensaje("Para comprar",8,40);
	imprimir_mensaje("presione el",11,31);
	imprimir_mensaje("boton",27,22);
	imprimir_mensaje("* 30 dias $50",0,7);
	actualizar_pantalla();
};
```

```
void pantalla_inicio(unsigned char imagen[], unsigned char x, unsigned char y)
{
	limpiar_RAM();
	imprimir_mensaje("Bienvenido a",6,25);
	imprimir_mensaje("**e-Monarca**",3,15);

//	imprimir_mensaje("Para comprar",0,18);
//	imprimir_mensaje("presione",0,10);
//	imprimir_mensaje("el boton",0,2);


//	imprimir_mensaje("Selecione",0,18);
//	imprimir_mensaje("el plan",0,10);
//	imprimir_mensaje("que desea",0,2);

//	cargar_imagen("boton_pres.lcd",58,0);
	actualizar_pantalla();
}

unsigned char solicitar_clave(unsigned char planX[])
{
	unsigned char i, j, k;
	unsigned char peticion[512];
	strcpy(peticion,"wget -cq -t 3 http://");
	strcat(peticion,ipservidor);
	strcat(peticion,"/d693a241-7e13-433f-9158-86ad865b5695/"); //folder de activacion
	strcat(peticion,planX);
	strcat(peticion," && cat ");
	strcat(peticion,planX);

	ejecutar_cmd(peticion);

	i=5; j=0;
	while(salida_cmd[0][i] != ' ' && salida_cmd[0][i])
		clave[j++]=salida_cmd[0][i++];
	clave[j]=0; j=0; i++;
	while(salida_cmd[0][i] != ' ' && salida_cmd[0][i])
		vigencia_fecha[j++]=salida_cmd[0][i++];
	vigencia_fecha[j]=0; j=0; i++;
	while(salida_cmd[0][i] != ' ' && salida_cmd[0][i])
		vigencia_hora[j++]=salida_cmd[0][i++];
	vigencia_hora[j]=0;
	remove(planX);
//	printf("\n%s %s %s\n\n", clave, vigencia_fecha, vigencia_hora);
	sleep(5);
	return 1;
}

void autorizacion_aceptada(unsigned char imagen[], unsigned char x, unsigned char y)
{
	limpiar_RAM();
```

```
	imprimir_mensaje("Autorizacion", 6,40);
	imprimir_mensaje("aceptada", 17,32);
	imprimir_mensaje("Deposite",2,15);
	imprimir_mensaje("el monto",2,5);
	cargar_imagen(imagen,x,y);
//	cargar_imagen("/home/juliocesar/Escritorio/imagenes_pantallita/pague.lcd",57,1);
	actualizar_pantalla();
}

void pago_recibido(unsigned char imagen[], unsigned char x, unsigned char y)
{
	limpiar_RAM();
	cargar_imagen(imagen,x,y);
	imprimir_mensaje("Pago recibido", 3, 40);
	imprimir_mensaje("Imprim.", 3, 22);
	imprimir_mensaje("recibo",3,12);
	imprimir_mensaje("Espere",3,2);
//	cargar_imagen("/home/juliocesar/Escritorio/imagenes_pantallita/pagado.lcd",50,2);
	actualizar_pantalla();
}

void imprimiendo(unsigned char imagen[], unsigned char x, unsigned char y)
{
	limpiar_RAM();
	imprimir_mensaje("Imprimiendo",8,40);
	cargar_imagen(imagen,x,y);
//
	cargar_imagen("/home/juliocesar/Escritorio/imagenes_pantallita/impresora.lcd",25,3)
;
	actualizar_pantalla();
}

void recoja_su_recibo(unsigned char imagen[], unsigned char x, unsigned char y)
{
	limpiar_RAM();
	imprimir_mensaje("Recoja su",15,40);
	imprimir_mensaje("recibo",24,32);
	cargar_imagen(imagen,x,y);
//
	cargar_imagen("/home/juliocesar/Escritorio/imagenes_pantallita/recoger_recibo.lcd",
27,0);
	actualizar_pantalla();
}

void detalles_recibo(void)
{
	limpiar_RAM();
//	strcpy(folio,"000000");
//	imprimir_mensaje("Folio:",0,41);
//	imprimir_mensaje(folio,36,40);

	imprimir_mensaje("Clave:", 0, 41);
```

```
	imprimir_mensaje(clave,24,32);

	imprimir_mensaje("Vig:", 0, 26);
	imprimir_mensaje(vigencia_fecha,18,18);
	imprimir_mensaje(vigencia_hora,36,9);
	imprimir_mensaje("Pulse Terminar",0,0);
	actualizar_pantalla();
}

void fin(void)
{
	limpiar_RAM();
	imprimir_mensaje("Gracias por su",0,32);
	imprimir_mensaje("Preferencia",9,24);
	imprimir_mensaje("** FIE Net **",3,10);
	actualizar_pantalla();
}

void ponerpixel(volatile unsigned char X, volatile unsigned char Y)
{
	switch(Y%8)
	{
		case 0: RAM_VIDEO[84*(5-(int)(Y/8))+X]|=0x80; break;
		case 1: RAM_VIDEO[84*(5-(int)(Y/8))+X]|=0x40; break;
		case 2: RAM_VIDEO[84*(5-(int)(Y/8))+X]|=0x20; break;
		case 3: RAM_VIDEO[84*(5-(int)(Y/8))+X]|=0x10; break;
		case 4: RAM_VIDEO[84*(5-(int)(Y/8))+X]|=0x08; break;
		case 5: RAM_VIDEO[84*(5-(int)(Y/8))+X]|=0x04; break;
		case 6: RAM_VIDEO[84*(5-(int)(Y/8))+X]|=0x02; break;
		case 7: RAM_VIDEO[84*(5-(int)(Y/8))+X]|=0x01; break;
	};
}

void borrapixel(volatile unsigned char X, volatile unsigned char Y)
{
	switch(Y%8)
	{
		case 0: RAM_VIDEO[84*(5-(int)(Y/8))+X]&=~0x80; break;
		case 1: RAM_VIDEO[84*(5-(int)(Y/8))+X]&=~0x40; break;
		case 2: RAM_VIDEO[84*(5-(int)(Y/8))+X]&=~0x20; break;
		case 3: RAM_VIDEO[84*(5-(int)(Y/8))+X]&=~0x10; break;
		case 4: RAM_VIDEO[84*(5-(int)(Y/8))+X]&=~0x08; break;
		case 5: RAM_VIDEO[84*(5-(int)(Y/8))+X]&=~0x04; break;
		case 6: RAM_VIDEO[84*(5-(int)(Y/8))+X]&=~0x02; break;
		case 7: RAM_VIDEO[84*(5-(int)(Y/8))+X]&=~0x01; break;
	};
}

void cambiapixel(volatile unsigned char X, volatile unsigned char Y)
{
	switch(Y%8)
	{
```

```
		case 0: RAM_VIDEO[84*(5-(int)(Y/8))+X]^=0x80; break;
		case 1: RAM_VIDEO[84*(5-(int)(Y/8))+X]^=0x40; break;
		case 2: RAM_VIDEO[84*(5-(int)(Y/8))+X]^=0x20; break;
		case 3: RAM_VIDEO[84*(5-(int)(Y/8))+X]^=0x10; break;
		case 4: RAM_VIDEO[84*(5-(int)(Y/8))+X]^=0x08; break;
		case 5: RAM_VIDEO[84*(5-(int)(Y/8))+X]^=0x04; break;
		case 6: RAM_VIDEO[84*(5-(int)(Y/8))+X]^=0x02; break;
		case 7: RAM_VIDEO[84*(5-(int)(Y/8))+X]^=0x01; break;
	};
}

unsigned char test_pixel(volatile unsigned char X, volatile unsigned char Y)
{
	unsigned char x=0;
	switch(Y%8)
	{
		case 0: if(RAM_VIDEO[84*(5-(int)(Y/8))+X]&0x80) x=1; break;
		case 1: if(RAM_VIDEO[84*(5-(int)(Y/8))+X]&0x40) x=1; break;
		case 2: if(RAM_VIDEO[84*(5-(int)(Y/8))+X]&0x20) x=1; break;
		case 3: if(RAM_VIDEO[84*(5-(int)(Y/8))+X]&0x10) x=1; break;
		case 4: if(RAM_VIDEO[84*(5-(int)(Y/8))+X]&0x08) x=1; break;
		case 5: if(RAM_VIDEO[84*(5-(int)(Y/8))+X]&0x04) x=1; break;
		case 6: if(RAM_VIDEO[84*(5-(int)(Y/8))+X]&0x02) x=1; break;
		case 7: if(RAM_VIDEO[84*(5-(int)(Y/8))+X]&0x01) x=1; break;
	};
	return x;
}

void imprimir_mensaje(unsigned char mensaje[], unsigned char x, unsigned char y)
{
	unsigned int i, j, k;
	k=0;
	while(mensaje[k])
	{
		for(i=5*(mensaje[k]-32);i<(5+5*(mensaje[k]-32));i++)
		{
			for(j=0;j<8;j++)
				if((font[i]<<j)&0x80)
				{
					ponerpixel(i-5*(mensaje[k]-32)+x,j+y-1);
				};
		};
		x=i-5*(mensaje[k]-32)+x+1;
		k++;
	};
}

void cargar_imagen(unsigned char imagen[], unsigned char posX, unsigned char posY)
{
	unsigned char dir_archivo[512];
	unsigned char linea[85];
	strcpy(dir_archivo,FOLDER_IMAGENES);
```

```
	strcat(dir_archivo,imagen);
//	printf("%s\n",dir_archivo);

	int i, j, k;
	FILE *archivo=fopen(dir_archivo,"r");
	j=0;
	while(!feof(archivo))
	{
		fscanf(archivo,"%[^\n]\n",linea);
		j++;
	}
	rewind(archivo);
	k=j-1;
	j=0;
	while(!feof(archivo))
	{
		fscanf(archivo,"%[^\n]\n",linea);
		for(i=0;linea[i];i++)
			if(linea[i]=='1')
				ponerpixel(i+posX,k-j+posY);
//				printf("%i,%i\n",i+posX,k-j+posY);
		j++;
	};
//	free(*pntr0);
	fclose(archivo);
}

void cargar_logo(unsigned char imagen[])
{
	unsigned char dir_archivo[512];
	unsigned char linea[84];
	strcpy(dir_archivo,FOLDER_IMAGENES);
	strcat(dir_archivo,imagen);
//	printf("%s\n",dir_archivo);
	int i, j, k;
	FILE *archivo=fopen(dir_archivo,"r");

	j=0;
	while(!feof(archivo))
	{
		fscanf(archivo,"%[^\n]\n",linea);
		j++;
	}
	rewind(archivo);
	k=j-1;
	j=0;
	while(!feof(archivo))
	{
		fscanf(archivo,"%[^\n]\n",linea);

		for(i=0;linea[i];i++)
			if(linea[i]=='1')
```

```
                        {
                                logo[i][k-j]=1;
//                              logo_original[i][k-j]=1;
//                              printf("%i",logo[i][j]);
                        }

                        else
                        {
                                logo[i][k-j]=0;
//                              logo_original[i][k-j]=0;
//                              printf("%i",logo[i][j]);
                        };
//              printf("\n");
                j++;
        };
        fclose(archivo);

}

void imprimir_logo(void)
{
        unsigned char x, y;
        for(x=0;x<84;x++)
                for(y=0;y<48;y++)
                {
//                      logo[x][y]=logo_original[x][y];
                        if(logo[x][y]) ponerpixel(x,y);
                        else borrapixel(x,y);
                };
        actualizar_pantalla();
}

void animacion_izquierda_logo(void)
{
        unsigned int i, j, k;
        unsigned char matriz_animacion[84][48];
        unsigned char vector_temp_mem[48];

        for(i=0;i<48;i++)
                vector_temp_mem[i]=logo[0][i];

        for(i=0;i<83;i++)
                for(j=0;j<48;j++)
                        logo[i][j]=logo[i+1][j];
        for(i=0;i<48;i++)
                logo[83][i]=vector_temp_mem[i];

        for(i=0,k=0;i<84;i++)
        {
        //              printf("%i ",k++);
                for(j=0;j<48;j++)
        //                      if(logo[i][j]==1)
```

```
                        if(logo[i][j]==1)
        //                              printf("1");
                                ponerpixel(i,j);
                        else
//                              printf("0");
                                borrapixel(i,j);
        //              printf("\n");
        }
        actualizar_pantalla();

}

void animacion_derecha_logo(void)
{
        unsigned int i, j, k;
        unsigned char matriz_animacion[84][48];
        unsigned char vector_temp_mem[48];

        for(i=0;i<48;i++)
                vector_temp_mem[i]=logo[83][i];

        for(i=83;i>0;i--)
                for(j=0;j<48;j++)
                        logo[i][j]=logo[i-1][j];
        for(i=0;i<48;i++)
                logo[0][i]=vector_temp_mem[i];

        for(i=0,k=0;i<84;i++)
        {
        //              printf("%i ",k++);
                for(j=0;j<48;j++)
        //                      if(logo[i][j]==1)
                        if(logo[i][j]==1)
        //                              printf("1");
                                ponerpixel(i,j);
                        else
        //                              printf("0");
                                borrapixel(i,j);
        //              printf("\n");
        }
        actualizar_pantalla();
}

void ejecutar_cmd(unsigned char cmd[])
{
        char *command=cmd;
        unsigned char i=0;
        FILE *tuberia_lectura=popen(command, "r");
//      FILE *tuberia_escritura=popen(command, "w");

        if(!tuberia_lectura)
        {
```

```
                fprintf(stderr, "incorrect parameters or too many files.\n");
                return;
        };
        while(!feof(tuberia_lectura))
        {
                fscanf(tuberia_lectura,"%[^\n]\n",salida_cmd[i++]);
//              printf("i=%i, -> %s\n", i-1, salida_cmd[i-1]);
        };
        pclose(tuberia_lectura);
//      pclose(tuberia_escritura);

}

void deslizar_imagen(unsigned char xi, unsigned char  yi, unsigned char  n, unsigned char  m,
unsigned char k, unsigned char modo)
{
//xi x de inicio
//yi y de inicio
//n numero de pixeles en x
//m numero de pixeles en y
//k numero de lugares que se debe de recorrer un pixel
//m es el modo izq der arri abaj
//      printf("xi = %i, yi = %i, m = %i, n = %i\n",xi, yi, m, n);
        if((k>0) && (m>1) && (n>1))
        {
                unsigned char i, j, x, y, xt[n], yt[m];
                for(j=0;j<k;j++)
                {
                        switch(modo)
                        {
                                case ARRIBA: //arriba
                                        for(x=xi;x<(xi+n);x++) xt[x-xi]=test_pixel(x,yi+m-1);
                                        for(i=0;i<(m-1);i++) for(x=xi;x<(xi+n);x++)
mover_pixel(x,yi+m-2-i,x,yi+m-1-i);
                                        for(x=xi;x<(xi+n);x++) if(xt[x-xi]) ponerpixel(x,yi); else
borrapixel(x,yi);
                                break;
                                case DERECHA: //derecha
                                        for(y=yi;y<(yi+m);y++) yt[y-yi]=test_pixel(xi+n-1,y);
                                        for(i=0;i<(n-1);i++) for(y=yi;y<(yi+m);y++)
mover_pixel(xi+n-2-i,y,xi+n-1-i,y);
                                        for(y=yi;y<(yi+m);y++) if(yt[y-yi]) ponerpixel(xi,y); else
borrapixel(xi,y);
                                break;
                                case ABAJO://abajo
                                        for(x=xi;x<(xi+n);x++)  xt[x-xi]=test_pixel(x,yi);
                                        for(i=0;i<(m-1);i++) for(x=xi;x<(xi+n);x++)
mover_pixel(x,yi+1+i,x,yi+i);
                                        for(x=xi;x<(xi+n);x++) if(xt[x-xi]) ponerpixel(x,yi+m-1);
else borrapixel(x,yi+m-1);
                                break;
                                case IZQUIERDA://izquierda
```

```
                                        for(y=yi;y<(yi+m);y++) yt[y-yi]=test_pixel(xi,y);
                                        for(i=0;i<(n-1);i++) for(y=yi;y<(yi+m);y++)
mover_pixel(xi+1+i,y,xi+i,y);
                                        for(y=yi;y<(yi+m);y++) if(yt[y-yi]) ponerpixel(xi+n-1,y);
else borrapixel(xi+n-1,y);
                                    break;
                                };
                        };
                        actualizar_pantalla();
        };
}

void mover_pixel(unsigned char xi, unsigned char yi, unsigned char xf, unsigned char yf)
{
        if(test_pixel(xi, yi)) ponerpixel(xf, yf);
        else borrapixel(xf, yf);
        borrapixel(xi, yi);
}

void enviar_byte(unsigned char dato)
{
        volatile unsigned char i;
        volatile unsigned int k; //para DELAY por macro!!!
        CLR_PORT(SCLK);
        DELAY;
        CLR_PORT(CE);
        for (i=0;i<8;i++)
        {
                if(dato&0x80) SET_PORT(SDA);
                else CLR_PORT(SDA);
                DELAY;
                SET_PORT(SCLK);
                DELAY;
                DELAY;
                CLR_PORT(SCLK);
                DELAY;
                dato<<=1;
        };
        SET_PORT(CE);
        DELAY;
        SET_PORT(SCLK);
}

void enviar_commando(unsigned char comando)
{
        CLR_PORT(DC);
        enviar_byte(comando);
        SET_PORT(DC);
}

void enviar_dato(unsigned char dato)
{
```

```
//        SET_PORT(DC);
          enviar_byte(dato);
}

void reset_lcd(void)
{
          unsigned char i;
          CLR_PORT(RST);
          for(i=0;i<10;i++);
          SET_PORT(RST);
          CLR_PORT(LIGHT);
}

void config_lcd(void)
{
          enviar_commando(0x21);
          enviar_commando(0xB9);
          enviar_commando(0x04);
          enviar_commando(0x14);
          enviar_commando(0x20);
          enviar_commando(0x0C);
}

void CLR_lcd(void)
{
          unsigned int a;
          enviar_commando(0x40);
          enviar_commando(0x80);
          limpiar_RAM();
          actualizar_pantalla();
}

void actualizar_pantalla(void)
{
          unsigned int i;
          if(salidaCPU) pantalla_CPU();
          else
          {
                    enviar_commando(0x01);
                    enviar_commando(0x00);
                    enviar_commando(0x40);
                    enviar_commando(0x80);
                    for(i=0;i<504;i++) enviar_dato(RAM_VIDEO[i]);
          };
/**/

}

void pantalla_CPU(void)
{
          unsigned char i, j;
          printf("\n ");
```

```
	for(i=0;i<84;i++)		printf("-");
	for(j=48;j>0;j--)
	{
		printf("\n|");
		for(i=0;i<84;i++)
			if(test_pixel(i,j-1)) printf("o");
			else printf(" ");
		printf("|");
	}
	printf("\n ");
	for(i=0;i<84;i++)		printf("-");
printf("\n");
}

void inicializar_puertos(void)
{
	unsigned char i;
	for(i=2;i<28;i++) INP_GPIO(i); // must use INP_GPIO before we can use OUT_GPIO
}

void inicializar_lcd(void)
{
	OUT_GPIO(RST);
	OUT_GPIO(CE);
	OUT_GPIO(DC);
	OUT_GPIO(SDA);
	OUT_GPIO(SCLK);
	OUT_GPIO(LIGHT);
	reset_lcd();
	config_lcd();
	CLR_lcd();
}

void limpiar_RAM(void)
{
	unsigned int i;
	for(i=0;i<504;i++) RAM_VIDEO[i]=0;
}

void setup_io()
{

  /* open /dev/mem */
  if ((mem_fd = open("/dev/mem", O_RDWR|O_SYNC) ) < 0)
  {
    printf("can't open /dev/mem \n");
    exit(-1);
  }

  /* mmap GPIO */
  gpio_map = mmap(
    NULL,          //Any adddress in our space will do
```

```
    BLOCK_SIZE,      //Map length
    PROT_READ|PROT_WRITE,// Enable reading & writting to mapped memory
    MAP_SHARED,      //Shared with other processes
    mem_fd,          //File to map
    GPIO_BASE        //Offset to GPIO peripheral
  );
  close(mem_fd); //No need to keep mem_fd open after mmap
   if (gpio_map == MAP_FAILED)
  {
    printf("mmap error %d\n", (int)gpio_map);//errno also set!
    exit(-1);
  };
   // Always use volatile pointer!
  gpio = (volatile unsigned *)gpio_map;
} // setup_io
unsigned char numero_decimal_a_numero_ascii(unsigned char n)
{
        switch(n)
        {
                case 0: n='0'; break;
                case 1: n='1'; break;
                case 2: n='2'; break;
                case 3: n='3'; break;
                case 4: n='4'; break;
                case 5: n='5'; break;
                case 6: n='6'; break;
                case 7: n='7'; break;
                case 8: n='8'; break;
                case 9: n='9'; break;
        };
        return n;
}
unsigned char numero_ascii_a_numero_decimal(unsigned char n)
{
        switch(n)
        {
                case '0': n=0; break;
                case '1': n=1; break;
                case '2': n=2; break;
                case '3': n=3; break;
                case '4': n=4; break;
                case '5': n=5; break;
                case '6': n=6; break;
                case '7': n=7; break;
                case '8': n=8; break;
                case '9': n=9; break;
        };
        return n;
}
```

Referencias

[1] J. M. De Velasco Farinós, *Los orígenes de Internet en JUNTA DE EXTREMADURA Consejería de Educación, Ciencia y Tecnología Dirección General de Ordenación, Renovación y Centros Mérida*., 2001, pp. 17 - 25.

[2] Cable coaxial. (2017, Accedido el 10 septiembre) [Online]. https://images-na.ssl-images-amazon.com/images/I/71Nh1SAs0uL._SL1347_.jpg

[3] Cable UTP. (2017, Accedido el 10 septiembre) [Online]. https://rukminim1.flixcart.com/image/1664/1664/data-cable/lan-cable/u/f/n/technotech-rj45-cat6-network-3-meter-original-imaejj6h2xszvgpu.jpeg

[4] Fibra óptica. (2017, Accedido el 10 septiembre) [Online]. https://a2sound.com/wp-content/uploads/ST-optical-fiber-connector-hdr-0a.jpg

[5] Antena satelital. (2017, Accedido en septiembre) [Online]. https://www.agenciasinc.es/var/ezwebin_site/storage/images/multimedia/videos/inaugurado-un-nuevo-centro-de-satelites-en-puertollano-y-presentado-deimos-2/2996240-4-esl-MX/Inaugurado-un-nuevo-centro-de-satelites-en-Puertollano-y-presentado-Deimos-2.jpg

[6] Módem USB Wifi. (2017, Accedido el 10 de septiembre) [Online]. https://images.wehkamp.nl/i/wehkamp/16171178_pb_01/tp-link-archer-t2uh-dual-band-wi-fi-adapter-wit-6935364050931.jpg?w=2048

[7] Módem Mi-fi. (2017, Accedido el 10 de septiembre) [Online]. http://cdn1.ntv.com.tr/gorsel/teknoloji/100-teknolojik-harika/novatel-wireless-mifi,pe2Z0RfzJ0uoEfnNP6zLTg.jpg?mode=crop&scale=both&v=20111208110815000&maxWidth=620

[8] Cobertura Wi-Fi de Wicoin en México. (2017, Accedido el 10 de septiembre) [Online]. https://lh3.googleusercontent.com/_sF50qdJKinlDKlboGwBEo_sGMBjxsSbEkI87epIN_I_yzLq6DMY5dkP_F5DfmjkDxYBOA=s125

[9] Mapa de cobertura de Megacable en el país. (2017, Accedido el 11 de septiembre) [Online]. https://megacablecobertura.mx/template/views/convertia/201907-site-mcable-cobertura/assets/svg/mapa.svg

[10] Esquema ilustrativo de una estación PSII. (2017, Accedido el 11 de sptiembre) [Online]. http://www.enlacedirecto.cl/wp-content/uploads/2015/07/sol10.jpg

[11] Chillispot Captive Portal. (2018, Consultado el 5 de agosto) Chillispot. [Online]. www.chillispot.org

[12] Topología general para un servidor NAS. (2017, Accedido el 12 de septiembre) [Online]. https://www.records.asia/wp-content/uploads/2011/03/radius-server.gif

[13] Topología de red típica para chillispot. (2017, Accedido el 11 de septiembre) [Online]. http://www.chillispot.org/images/chilli.png

[14] RADIUS-Wikipedia, la enciclopedia libre.. (2018, Consultado el 10 de agosto) [Online]. https://es.wikipedia.org/wiki/RADIUS

[15] Gabinete metálico. (2018, Imagen descargada el 20 de octubre) [Online]. www.pendiente.com

[16] Presión positiva y presión negativa del aire. (2018, Accedido el 21 de abril) [Online]. https://www.profesionalreview.com/2017/10/29/que-es-presion-positiva-y-negativa-de-aire-pc/

[17] Portal captivo. (2018, Accedido en agosto) [Online]. http://www.zeroshell.net/es/captiveportaldetails/

[18] Raspberry Pi Zero W. (2019, Descargada el 5 de abril) [Online]. https://images-na.ssl-images-amazon.com/images/I/410O%2BX7WgPL._SX425_.jpg

[19] Enumeración de los pines de la raspberry. (2019, Descargada en abril) [Online]. http://www.raspberry-pi-geek.com/var/rpi/storage/images/media/images/raspib-gpio/12356-1-eng-US/RasPiB-GPIO_lightbox.png

[20] Parte frontal de la pantalla LCD-10168. (2019, Descargada el 8 de abril) [Online]. https://cdn.sparkfun.com//assets/parts/4/4/7/3/10168-01a.jpg

[21] Parte posterior de la pantalla LCD-10168. (2019, Descargada el 5 de abril) [Online]. https://electronilab.co/wp-content/uploads/2014/05/lcd_nokia5110_3_Electronilab-e1400699820698.jpg

[22] Diagrama de tiempos del protocolo SPI. (2019, Descargada el 5 de abril) [Online]. https://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf

[23] Botón de presión o push boton. (2019, Descargada el 20 de abril) [Online]. https://http2.mlstatic.com/boton-pulsador-maquinita-15-ampere-D_NQ_NP_888442-MLM28529718677_102018-F.jpg

[24] Led RGB. (2019, Descargada el 25 de abril) [Online]. https://http2.mlstatic.com/10-led-rgb-5mm-anodo-catodo-comun-ultrabrillante-D_NQ_NP_647890-MLM29848531754_042019-O.jpg

[25] Impresora QUIAN ANJET80 QIT801701. (2019, Descargada el 25 de abril) [Online]. https://www.qian.mx/wp-content/uploads/2017/07/0W3A7917-1030x687.jpg

[26] Protocolo IPP. (2018, Accedido el 16 de septiembre) [Online]. https://support.usr.com/support/5436/5436-es-ug/twelve.html

[27] Iptables_1. (2018, Accedido en septiembre) [Online]. https://blog.desdelinux.net/redireccionar-trafico-iptables/

[28] Iptables_2. (2018, Accedido en septiembre) [Online]. https://www.enlinux.org/puertos-y-servicios-en-gnulinux-centos/

[29] Iptables_3. (2018, Accedido en septiembre) [Online]. https://blog.desdelinux.net/comandos-para-comprobar-si-un-puerto-esta-abierto-en-nuestra-pcservidor-u-otro-remoto/

[30] Iptables_4. (2018, Accedido en septiembre) [Online]. https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/s1-iptables-options.html

[31] Administración de impresoras en Linux_1. (2019, Accedido en abril) [Online]. http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m5/administracin_del_servidor_de_impresin.html

[32] Administración de impresoras en Linux_2. (2019, Accedido en abril) [Online]. http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m1/administracin_de_usuarios_y_grupos.html

[33] Administración de impresoras en Linux_3. (2019, Accedido en abril) [Online]. https://www.garron.me/es/gnu-linux/agregar-usuario-grupo-linux.html

[34] Moneda de 5 pesos mexicanos. (2019, Descargada en abril) [Online]. https://i.colnect.net/f/2921/784/5-Pesos.jpg

[35] Run levels en linux_1. (2019, Accedido en marzo) [Online]. https://elbauldelprogramador.com/entendiendo-los-runlevel-en-debian-y-ubuntu/

[36] Run levels en linux_2. (2019, Accedido en marzo) [Online]. https://linuxpractico.wordpress.com/2009/01/01/agregar-eliminar-scripts-de-inicio-los-runlevels-de-system-v/

[37] Plan Nacional de desarrollo 2018 - 2024. (2019, Descargado en junio) Pdte. Andrés Manuel López Obrador. https://www.gob.mx/cms/uploads/attachment/file/458250/20190430-XVIII-1.pdf pág. 172.

[38] EasyHotspot. (2017, septiembre) https://sourceforge.net/projects/easyhotspot/. [Online]. https://sourceforge.net/projects/easyhotspot/

[39] Manual de Mysql. (2018, septiembre) [Online].

https://downloads.mysql.com/docs/refman-5.0-es.pdf