DESARROLLO DE UNA APLICACIÓN WEB PARA PORTAL CAUTIVO QUE PERMITA EL DESPLIEGUE DE DIFERENTES CONTENIDOS PUBLICITARIOS

JUAN PABLO AGUIRRE MARTÍNEZ
JULIANA CASTELLANOS GARCÍA

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE OPERACIONES Y SISTEMAS
PROGRAMA INGENIERÍA MULTIMEDIA
SANTIAGO DE CALI
2017

DESARROLLO DE UNA APLICACIÓN WEB PARA PORTAL CAUTIVO QUE PERMITA EL DESPLIEGUE DE DIFERENTES CONTENIDOS PUBLICITARIOS

JUAN PABLO AGUIRRE MARTÍNEZ

JULIANA CASTELLANOS GARCÍA

Proyecto de Grado para optar al título de Ingeniero Multimedia

Director
HELMUT ALEXANDER RUBIO WILSON
Ingeniero en electrónica y telecomunicaciones
Magíster en sistemas inalámbricos

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE OPERACIONES Y SISTEMAS
PROGRAMA INGENIERÍA MULTIMEDIA
SANTIAGO DE CALI
2017

Nota de aceptación:

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar al título de Ingeniero Multimedia

JUAN JOSÉ CARDONA

Jurado

OSCAR HERNAN MONDRAGÓN

Jurado

Santiago de Cali, 1 de Junio de 2017

CONTENIDO

LISTA DE CUADROS

LISTA DE FIGURAS

LISTA DE ANEXOS

Pág.

GLOSARIO

ACCESS POINT: es un dispositivo de red que interconecta equipos de comunicación inalámbrica para formar una red.

AUTENTICACIÓN: proceso para confirmar que algo o alguien es quien dice ser.

BACK-END: área de la programación que se dedica a la parte lógica de un sistema. El resultado del "back-end" en un proyecto no será visible ya que no involucra diseño, se trata de programar funcionalidades y comportamientos.

BALANCEO DE CARGAS: técnica usada para balancear o compartir el trabajo a realizar entre diferentes servidores mediante un algoritmo que asigna de la forma más conveniente dicho trabajo.

CÓDIGO FUENTE: son los archivos o "scripts" con las instrucciones necesarias, escritas en uno o varios lenguajes de programación para ejecutar un programa.

DIRECCIÓN "URL": identificador de recursos uniforme por sus siglas en inglés, es una secuencia de caracteres que permite denominar recursos en "Internet" con el fin de ser localizados.

DIRECCION IP: es un conjunto de cuatro números del 0 al 255 separado por puntos, esta dirección es única e irrepetible con la cual se identifica un dispositivo conectado a una red.

DIRECCIÓN MAC: también conocida como dirección física, es un identificador de 48 bits que corresponde únicamente a una tarjeta o dispositivo de red.

DISPOSITIVO DE RED: dispositivo hardware que se conecta a un segmento de red para transmitir datos a otros dispositivos de red.

FIRMWARE: es un programa informático que establece la lógica de más bajo nivel para controlar directamente la parte física o electrónica de un dispositivo.

FRAMEWORK: es una estructura que sirve de base para el desarrollo de software.

FRONT-END: área del desarrollo de “Software” que se encarga de la parte visual, como por ejemplo, colores, tipografías, fondos, tamaños, formas etc.

HARDWARE: partes tangibles de un sistema informático.

INTERFAZ DE RED: componente hardware que permite conectar un dispositivo a una red.

LOCALHOST: nombre reservado que tienen todos los dispositivos que disponga de una tarjeta de red para referirse a sí mismo.

SISTEMA OPERATIVO: es el “Software” principal de un sistema informático que gestiona los recursos tanto “Software” como “Hardware”.

SOFTWARE DE CÓDIGO ABIERTO: es el software cuyo código fuente es publicado, esto permite a los usuarios utilizar, cambiar, mejorar y redistribuir el “Software” ya sea en su forma modificada u original.

SOFTWARE: conjunto de componentes lógicos que hacen posible la realización de tareas específicas.

VERSION BETA: primera versión completa, pero con posibles errores de un programa informático.

RESUMEN

En el presente trabajo de grado se documenta todo el proceso seguido para desarrollar una aplicación web que permita la administración de un portal cautivo con la capacidad de desplegar distintos contenidos publicitarios, dependiendo del Access Point desde el cual se intenta el ingreso a una red pública inalámbrica; esto con el objetivo de brindar a estas redes, ampliamente utilizadas en espacios cotidianos de la vida citadina, un valor agregado, al darles la posibilidad de desplegar publicidad digital de forma dinámica e inteligente para el mercado.

Este documento contiene el conjunto de información necesaria para entender todos aquellos conocimientos técnicos que son fundamentales para soportar el desarrollo del proyecto y describe todas las etapas que hicieron parte del mismo, iniciando por el estudio de diferentes portales cautivos de código abierto, completamente funcionales, que se pueden encontrar en internet, para posteriormente seleccionar “GraseHotSpot” como el proyecto de portal cautivo que mejor se adapta al escenario planteado. Sobre este portal cautivo se realiza una descripción de sus componentes, a la vez que se exploran distintos métodos mediante los cuales se hace posible identificar e individualizar el tráfico de acuerdo al “AccessPoint”, para finalmente realizar un filtro de direcciones “IP” utilizando scripts de “PHP”, junto con su correspondiente aplicación “web” de administración, que permite indicar los rangos de direcciones “IP” que corresponden a un contenido particular.

En la sección final se presenta el diseño de una prueba, que especifica una situación particular bajo la cual se puede verificar que las modificaciones realizadas al portal cautivo permiten individualizar el contenido dependiendo de la dirección “IP” del cliente. De esta forma se documenta la implementación de dicha prueba y se exponen los resultados obtenidos, para finalmente extraer conclusiones que versan sobre distintos aspectos implicados en el desarrollo del proyecto.

Palabras clave: portal cautivo, código abierto, filtrado de trafico de red, ip, mac, ssid, aplicación web, red pública inalámbrica, servidor, publicidad, protocolos de red, dispositivos de red.

INTRODUCCIÓN

La masificación del uso de internet es una realidad global desde hace muchos años. Según la CRC (Comisión de Regulación de Comunicaciones) [1], en el 2014, en Colombia, el 52,6% de los individuos usaron Internet, la cual es una cifra que está por encima de los porcentajes del mundo y de los países en desarrollo.

Los avances tecnológicos y el furor de los dispositivos móviles han logrado que el acceso a internet haya migrado a una forma móvil también, de manera que la conexión a internet ahora es fácilmente implementada fuera de los hogares, y sobre todo, fuera de las redes privadas. Actualmente, es muy fácil conectarse a internet fuera de casa mediante un dispositivo móvil, y en diferentes ubicaciones como instituciones educativas, lugares de trabajo y puntos de acceso públicos, de manera gratuita.

El acceso a redes públicas gratuitas a través de medios inalámbricos toma una serie de consideraciones con respecto a procesos de autenticación, autorización y registros de los usuarios que ingresan a estas redes. Comúnmente, estos procesos se administran mediante un portal cautivo, que es una aplicación que intercepta el tráfico en la red e impide a los usuarios conectarse a internet hasta que pasen por una página web determinada.

Con el fin de explorar las posibilidades de explotación de los portales cautivos e intentar añadir más versatilidad y funcionalidad a su uso actual, se plantea el desarrollo de un portal cautivo que pueda administrar varias páginas web, para ser desplegadas dependiendo a la zona física de cobertura en la que se encuentre el usuario que está intentando acceder a internet por medio de la red inalámbrica pública.

Al tener la posibilidad de ofrecer una página web específica, basándose en la locación, será posible entregar contenidos diseñados particularmente para ser útiles o valiosos en función del entorno inmediato, y de esta forma se pueden idear nuevas formas de aprovechar las aplicaciones de portal cautivo, dentro de espacios variados, ya sea instalaciones educativas, centros comerciales, bibliotecas públicas, áreas de trabajo, aeropuertos, hoteles, entre muchas otros.

[1] Reporte de industria del sector TIC [en línea]. Comisión de Regulación de Comunicaciones, 2015. [Consultado 03 de marzo de 2016]. Disponible en Internet: http://colombiatic.mintic.gov.co/602/articles-13464_archivo_pdf.pdf

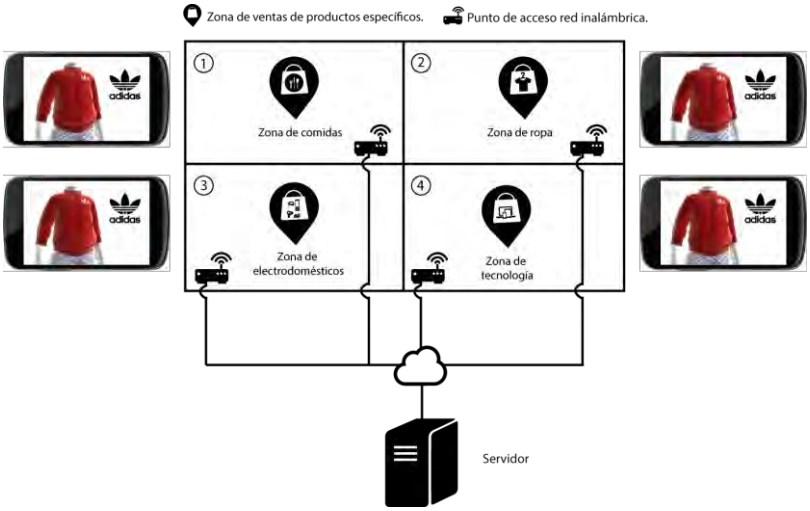
1. PLANTEAMIENTO DEL PROBLEMA

En años recientes, el Internet se ha convertido en un servicio de consumo masivo, tanto así que el acceso a la red se ha vuelto una prioridad para las personas, que desean estar conectadas en cualquier momento y lugar, razón por la cual, muchos establecimientos y espacios públicos han optado por brindar servicio de internet gratuito dentro de su propiedad.

Comúnmente, el acceso a internet se logra mediante la conexión del cliente a la red pública inalámbrica del establecimiento. Dicha conexión es gratuita, sin embargo, muchos espacios públicos buscan beneficiarse de ésta conexión al implementar portales cautivos que impiden al usuario hacer uso del internet o navegar por la web, a menos que ingresen a una página web predeterminada que es administrada por la aplicación de portal cautivo. Como estrategia de venta y marketing, se publicita en esta página, de manera que se encuentra una forma digital de desplegar publicidad y captar clientes a cambio del uso de un servicio con vasta demanda.

Esta estrategia particular representa, como es obvio, grandes beneficios, sin embargo, no está siendo aprovechada al máximo. En espacios públicos de gran tamaño, se utilizan varias zonas “wi-fi” para poder ofrecer suficiente cobertura en toda la extensión del lugar y debido a la manera en que los portales cautivos están diseñados, la misma página, con el mismo contenido publicitario, es desplegada en cada una de estas zonas sin diferenciación alguna, como se muestra en la Figura 1.

Figura 1. Funcionamiento portales cautivos tradicionales



Fuente: Elaboración propia.

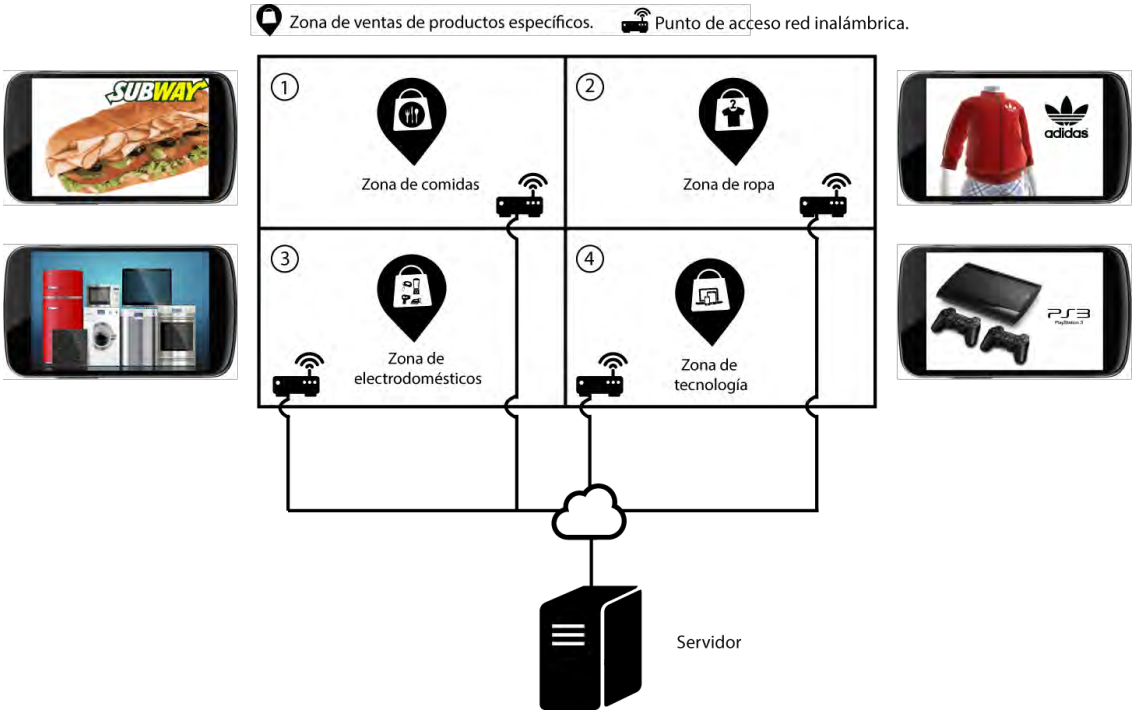
Lo anterior se debe a que los portales cautivos están configurados por defecto para redirigir a una única página web, a pesar de que en la práctica terminan interceptando tráfico “http” de múltiples usuarios que están ubicados en diferentes espacios. En ciertos contextos, esta configuración no representa un problema alguno, puesto que, si se desea simplemente que el usuario ingrese un nombre de usuario y una contraseña, no hay necesidad de variar el contenido de la página, por el contrario, una única página que atienda todos los usuarios por igual es la mejor solución.

Sin embargo, se presentan numerosos casos en los que los portales son utilizados principalmente para aceptar políticas de uso del servicio de “Internet” y al mismo tiempo, mostrar publicidad que puede ser de interés para la persona que está intentando acceder a la red. Bajo esta modalidad de uso, presentar una sola página web que no tenga en cuenta el punto desde el cual el usuario se conecta representa un problema, puesto que generaliza el contenido para todos los usuarios, cuando la información sobre la ubicación de los mismos, es esencial para poder incluir publicidad dirigida que pueda ser más precisa y efectiva para captar la atención de las personas, lo cual es un elemento de extremo valor, ya que el principal objetivo de la publicidad en los portales cautivos es poder monetizar el servicio de internet, aunque se esté ofreciendo gratuitamente.

Es de esta manera como se identifica entonces la necesidad de aprovechar al máximo la tecnología disponible para plantear estrategias publicitarias más eficientes, a la vez que se plantea la problemática de cómo adecuar un portal cautivo para el despliegue de diferentes contenidos publicitarios individualizados por áreas dentro de una red pública inalámbrica.

Frente a la problemática, se propone un modelo de distribución de contenidos personalizables a través de los portales cautivos, por zonas de la red inalámbrica, sin necesidad de emplear múltiples servidores, es decir, cada área “wi-fi” tendrá la posibilidad de mostrar al cliente una publicidad diferente según sea conveniente y pertinente en dicho espacio físico, de manera que el administrador de la red tendrá el poder para entregar publicidad dirigida específicamente a una zona particular de toda la propiedad, como se ilustra en la Figura 2.

Figura 2. Propuesta portal cautivo personalizado por áreas



Fuente: Elaboración propia.

2. JUSTIFICACIÓN

Colombia es un país que ha invertido esfuerzo, tiempo y dinero en el desarrollo tecnológico, a través de diferentes proyectos implementados como parte del plan "Vive Digital", una estrategia del gobierno para impulsar el salto tecnológico a través de la masificación del uso de internet. El plan vive digital se lanzó en 2010, y hasta la fecha ha recibido 3 reconocimientos internacionales, entre ellos, el de mejor plan de tecnología del mundo, otorgado en la feria más importante de la industria de telecomunicaciones, el GSMA Mobile World Congress de Barcelona [2].

Para la segunda versión del plan vive digital (2014-2018) se plantea un objetivo particular correspondiente a el acceso a internet desde redes públicas, dónde se espera tener por lo menos 1500 zonas de wifi público gratuito en todo el país [3].

Esta clase de estrategias nacionales son un reflejo de las tendencias que se observan globalmente, puesto que en el 2009 sólo existían 0.5 millones de "hotspots" para el acceso a internet a través de wifi público en todo el mundo,[4] para el 2015 se estimaba que habría un total de 7.8 millones, a la vez que se espera que esta cifra continúe creciendo rápidamente, de manera que en el 2020 se desplieguen aproximadamente 13.3 millones de estos "hotspots" [5].

Es de esta forma que se identifica claramente un entorno de auge tecnológico ampliamente apoyado y soportado por diferentes naciones, incluyendo el gobierno de la república de Colombia, en compañía de diferentes empresas del sector privado que se han sumado a la iniciativa. En un panorama preparado para la masificación del internet, y el acceso al mismo a través de redes públicas

---

[2] Premios [en línea]. Galardones otorgados al plan Vive Digital. Ministerio de tecnologías de la información y las comunicaciones. [Consultado 03 de Marzo de 2016] Disponible en Internet: http://micrositios.mintic.gov.co/vivedigital/logros-plan/logro.php?lg=27

[3] Acceso a Internet [en línea]. Metas. Ministerio de tecnologías de la información y las comunicaciones. [Consultado 03 de Marzo de 2016] Disponible en Internet: http://micrositios.mintic.gov.co/vivedigital/2014-2018/proposito.php?lg=18

[4] WBA Industry Report 2011 [en línea]. Global Developments In Public Wi-Fi. London: Wireless Broadband Alliance, 2011. [consultado 29 de Marzo de 2016]. Disponible en Internet: http://www.wballiance.com/wba/wp-content/uploads/downloads/2012/07/16_WBA-Industry-Report-2011-_Global-Developments-in-Public-Wi-Fi-1.00.pdf

[5] Global Public Wi-Fi Hotspots Will Reach 7.8 Million in 2015 and Continue To Grow at a CAGR of 11.2% through 2020 [en línea]. Singapore: ABI Research, 2015. [consultado 29 de Marzo de 2016]. Disponible en Internet : https://www.abiresearch.com/press/global-public-wi-fi-hotspots-will-reach-78-million/

inalámbricas, es importante pensar en desarrollar aplicaciones y componentes para la red, que permitan aprovecharla al máximo la demanda del recurso.

Es por tal razón, que mediante el desarrollo de este proyecto se pretende aportar un granito de arena en la construcción de nuevas formas de ofrecer contenidos digitales, a través de los portales cautivos que son implementados como sistema de autenticación en las redes públicas inalámbricas. Este proyecto encuentra un amplio campo de acción y por lo tanto su carácter es mayormente práctico, en concordancia con su intención de dar solución a situaciones de la vida real y el entorno cotidiano.

Actualmente, muchas de las redes públicas inalámbricas de acceso gratuito implementan portales cautivos como un medio para el control de acceso de usuarios a la red, pero sólo despliegan una página web, la misma página web, para el alcance total del área inalámbrica. Al introducir diferentes páginas web con diversos contenidos, dirigidos a espacios físicos específicos, se puede aprovechar en mayor medida la funcionalidad del portal cautivo para adaptarlo a las necesidades particulares de los diferentes espacios dentro de la cobertura de la red.

Una de las aplicaciones más evidentes para el despliegue de contenidos personalizados por áreas es la publicidad. Con la implementación de múltiples portales cautivos, se crea la posibilidad de entregar diferentes mensajes a los usuarios conectados a la red, que corresponden a lo que pueden encontrar o hacer en su entorno inmediato. Es de esta manera como el conectarse a internet podría significar ver una publicidad de su restaurante preferido, que coincidencialmente tiene promociones, y se encuentra a una distancia relativamente corta de su posición actual. O tal vez, podría significar que el acceso a internet desde un parque público permite conocer las distintas actividades de entrenamiento físico que se ofrecen en los alrededores, o incluso, podría dar indicaciones e información sobre lugares turísticos relevantes a un extranjero que se conecta a internet desde una red pública en Colombia por primera vez. Y los ejemplos siguen sobre la misma línea, cuantos sea que se puedan identificar.

Es entonces claro, que la implementación de múltiples portales cautivos podría masificar el marketing digital y establecer nuevas vías de comunicación con usuarios de toda clase de productos o servicios, razón por la cual el proyecto tiene capacidad para impactar en el área comercial desde una dimensión tecnológica, afectando directamente a las empresas dueñas de espacios de alto movimiento comercial, o diferente entidades que ofrezcan servicio wifi dentro de su propiedad, así como a los usuario finales, que reciben servicio gratuito de internet, a la vez que son informados sobre aspectos determinados de los lugares circundantes.

3. ANTECEDENTES

3.1 TEXTBLUE

En primer lugar se tiene que, en Newcastle Inglaterra, la empresa “TextBlue” conformada por especialistas en mercadeo móvil y desarrollo de aplicaciones, lanza al mercado un proyecto llamado “Internet Sharing Devices”, el cual consiste en la implementación de un dispositivo que permite acceso a una red inalámbrica privada “Wi-Fi” a usuarios anónimos, sin embargo una vez los usuarios acceden a la red, deberán completar acciones rápidas para mantener la conexión en dicha red.

Las acciones rápidas que deben completar los usuarios, pueden ser configuradas por el cliente que adquiere el servicio de “TextBlue”, estas van desde subscripciones a listas de difusión, un me gusta en “FaceBook”, seguir una cuenta de “Twitter”, ver algún video publicitario, descargar una aplicación, etc.

Para el correcto funcionamiento de este sistema de mercadeo digital, el cliente debe adquirir un dispositivo, el cual se encarga de generar la cobertura inalámbrica y a su vez, ejecutar y validar todos los relacionados con las acciones rápidas. “TextBlue” ofrece dos tipos de dispositivos que se adaptan de manera muy general a las necesidades del cliente y se enseñan en la Figura **3**.

Figura 3. Dispositivos para compartir Internet de TextBlue: Para interiores de forma circular, y antenas para exteriores

Fuente: Internet Sharing Devices. Newcastle: TextBlue [en línea]. [Consultado 01 de Marzo de 2016]. Disponible en Internet: http://www.textblue.net/?page=textblue_internet_sharing_devices

El primer dispositivo es para usar en interiores, cuenta con un rango de cobertura de aproximadamente 500 pies y tiene un costo de £199. El segundo dispositivo es para exteriores, cuenta con un rango de cobertura de aproximadamente 183 metros y su precio es de £229. Adicionalmente, si el cliente adquiere más de 3 dispositivos deberá pagar una cuota de £29 mensuales por concepto de administración y gestión continua.

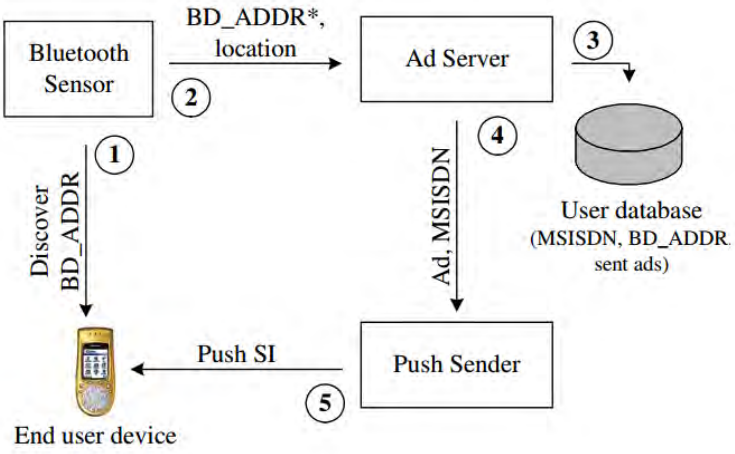
Mediante los productos y servicios de la empresa "TextBlue", es posible implementar estrategias de mercadeo y difusión de información, aprovechando al máximo un recurso que actualmente se encuentra disponible en la mayor parte de lugares públicos, el "Wi-Fi". Sin embargo, para el correcto funcionamiento de este servicio, es necesario adquirir la parte hardware que únicamente es ofrecida por la empresa a un precio relativamente elevado, por lo tanto, el servicio ofrecido por "TextBlue" se diferencia del propósito planteado en esta investigación, debido a que existe una dependencia con los dispositivos necesarios para la implementación del servicio, adicionalmente, estos son ofrecidos únicamente por la empresa.

3.2 SISTEMA DE PUBLICIDAD MÓVIL CON RECONOCIMIENTO DE UBICACIÓN PARA BLUETOOTH Y WAP PUSH

Por otra parte, en el campo de las redes personales inalámbricas "WPAN", se encuentra el proyecto "Bluetooth and WAP Push Based Location-Aware Mobile Advertising System" desarrollado por la empresa "Media Team". El proyecto consiste básicamente en la implementación de un servidor de "Bluetooth" para el despliegue de publicidad en teléfonos móviles, mediante el uso del protocolo "Wireless Application Protocol Push", el cual provee el servicio "Service Loading" con el propósito de enviar contenido a teléfonos móviles mediante "Bluetooth" sin la confirmación del usuario.

El funcionamiento del sistema al cual "Media Team" llamó "B-MAD" o "Bluetooth Mobile Advertising", consta de 5 macro etapas ilustradas en la Figura 4.

Figura 4. Arquitectura del sistema "B-MAD"



Fuente: AALTO, Lauri; *et al*. Bluetooth and WAP Push Based Location-Aware Mobile Advertising System [en línea].[Consultado 08 de Marzo de 2016]. Disponible en Internet: http://www.mediateam.oulu.fi/publications/pdf/496.pdf

- Un sensor de "Bluetooth" descubre la dirección de "Bluetooth" o "BD-Addrs" de los dispositivos cercanos.

- El sensor de "Bluetooth" envía la "BD-Addrs" mediante el protocolo WAP para establecer la conexión con el servidor de publicidad.

- El servidor de publicidad mediante una base de datos de direcciones de "Bluetooth", valida constantemente si algún registro existente se encuentra disponible para el envío de contenidos, es decir valida si el dispositivo se encuentra cerca y con el "Bluetooth" encendido.

- Los contenidos son enviados al "Push Sender", el cual se encarga de transmitir directamente la información al usuario, sin que este tenga la posibilidad de autorizar la recepción.

- Finalmente, el "Push Sender" entrega el contenido publicitario mediante "WAP Push". En la Figura 5, se puede observar el contenido publicitario.

Figura 5. Ejemplo de publicidad móvil en un Nokia 3650

Fuente: AALTO, Lauri; GOTHLIN, Nicklas; KORHONEN, Jani; OJALA, Timo. Bluetooth and WAP Push Based Location-Aware Mobile Advertising System [en línea]. [Consultado 08 de Marzo de 2016]. Disponible en Internet: http://www.mediateam.oulu.fi/publications/pdf/496.pdf

En la etapa de evaluación, el proyecto fue testeado en un ambiente ficticio y posteriormente en las calles del centro de la ciudad “Oulu” en el norte de Finlandia en ocho locales comerciales, los resultados de las pruebas fueron desalentadores.

Debido a que el servidor de “Bluetooth” debe establecer la conexión con los dispositivos cercanos mediante un proceso llamado “inquiri” o descubrimiento, al momento de múltiples peticiones de descubrimientos, el servidor de “Bluetooth” espera un tiempo pseudoaleatorio con el fin de ser un sistema libre de errores, y atender una a una todas las peticiones disponibles, el servidor tarda un lapso de tiempo considerable para finalizar este proceso. Para los teléfonos testeados Nokia 3650, este proceso tardaba en promedio 5 segundos, tiempo necesario para que un peatón se aleje a una distancia considerable de la tienda emisora del contenido publicitario.

“Bluetooth and WAP Push Based Location-Aware Mobile Advertising System” se diferencia del propósito planteado en esta investigación, debido a que implementa distintas tecnologías para el despliegue de contenidos, dichas tecnologías específicamente el “Bluetooth” está en una etapa de decrecimiento, debido a las características que presenta, como por ejemplo, el alto consumo de energía, el cual tiene un impacto considerable en telefonía móvil, y finalmente, la baja velocidad de transmisión.

3.3 IMPLEMENTACIÓN DE UN PORTAL CAUTIVO EN EL COLEGIO SAN LUIS GONZAGA

En la Universidad Politécnica Salesiana ubicada en Quito, Ecuador, un estudiante realizó un proyecto como tesis de grado para la carrera de ingeniería informática, en la cual implementaba un portal cautivo para el colegio San Luis Gonzaga.

En el proyecto, se identifica la necesidad de la institución educativa para dar cobertura de wifi principalmente en el área de la biblioteca, para uso exclusivo de profesores y alumnos, razón por la cual se controlará el acceso a la red a través de un portal cautivo, en el que los usuarios deberán identificarse con un nombre de usuario y contraseña. Para tal desarrollo se realizó un estudio de la red inalámbrica que ya estaba implementada, teniendo en cuenta factores como la cobertura e intensidad de la señal inalámbrica a distintas distancias del Access Point. Posteriormente se describen los elementos de hardware y software a usar y finalmente se implementa el portal cautivo de “Chillispot”, junto con una base de datos en “MySQL” y un servidor “freeRadius”, todos instalados en una máquina con DEBIAN como sistema operativo.

Luego de realizar las configuraciones apropiadas y unas cuantas pruebas, se obtiene como resultado el sitio web de bienvenida del portal cautivo, en la Figura 6, que redirige a una página de “login”, como se muestra en la Figura 7.

Figura 6. Página de bienvenida del portal cautivo del colegio San Luis Gonzaga

Fuente: MALDONADO, Angel. Implementación de un portal cautivo que permita el control de acceso al servicio de Internet a los estudiantes del colegio San Luis Gonzaga a través de una autenticación de los usuarios mediante un servicio AAA implementado en un servidor que trabaje con protocolo RADIUS. Tesis ingeniero de sistemas. Quito: Universidad Politécnica Salesiana. Facultad de Ingeniería, 2012. 130 p.

Figura 7. Página de "login" del portal cautivo del colegio San Luis Gonzaga

Fuente: MALDONADO, Angel. Implementación de un portal cautivo que permita el control de acceso al servicio de Internet a los estudiantes del colegio San Luis Gonzaga a través de una autenticación de los usuarios mediante un servicio AAA implementado en un servidor que trabaje con protocolo RADIUS. Tesis ingeniero de sistemas. Quito: Universidad Politécnica Salesiana. Facultad de Ingeniería, 2012. 111 p.

Este proyecto tiene la particularidad de mostrar una aplicación directa del portal cautivo, puesto que se muestra su implementación en el entorno en el que funcionará en la vida real. Adicionalmente, tiene en cuenta los requisitos del colegio, como por ejemplo que debe haber una clara identificación del usuario para poder saber quién está usando el internet y cómo, a demás no tiene en cuenta tiempos de uso para desconexión automática de la red, puesto que la institución deseaba ofrecer el servicio de manera permanente una vez que el usuario se identificara, y finalmente, el servicio era totalmente gratuito. Sin embargo, hay que tener en cuenta que en este proyecto, a diferencia del planteado en este documento, se realiza una implementación sencilla que si bien requiere de conocimientos teóricos y prácticos, no implica nuevas funcionalidades ni servicios fuera de lo común, y además, está pensado para su uso en un solo lugar, el cual era la biblioteca del colegio anteriormente mencionado.

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Desarrollar una aplicación Web que se integre a un portal cautivo para el despliegue de contenidos publicitarios, propios de un área inalámbrica definida dentro de una red pública.

4.2 OBJETIVOS ESPECÍFICOS

- Realizar un análisis comparativo de diferentes aplicaciones de portales cautivos completamente funcionales, modificables y de código abierto que se pueden encontrar actualmente en Internet.

- Seleccionar la aplicación de portal cautivo más adecuada a partir de la comparación de las alternativas presentes.

- Modificar el código fuente de la aplicación de portal cautivo para que pueda reconocer e individualizar el tráfico por “Accespoint” y permita gestionar diversos contenidos web personalizables por cada zona inalámbrica.

- Realizar una prueba de funcionamiento de la aplicación de portal cautivo con el servidor y mínimo dos zonas “wi-fi” diferentes.

5. MARCO TEÓRICO

5.1 ARQUITECURAS WLAN

Las especificaciones en el estándar “wi-fi” definen dos modos de operación para las redes inalámbricas de área local: Ad-Hoc e infraestructura. En el modo Ad-hoc no se utilizan APs, ya que los únicos actores en el proceso de comunicación inalámbrica son las estaciones o dispositivos finales. En el modo infraestructura se utilizan APs para comunicar a los clientes o estaciones con el resto de la red.

5.1.1 Ad-Hoc. También conocido como el modo “peer to peer” o IBSS (Independent Basic Service Set), donde la comunicación se realiza directamente entre diferentes estaciones que se conectan inalámbricamente a través de sus tarjetas de red, de manera que no se necesita ningún tipo de infraestructura inalámbrica o conexión cableada, como se enseña en la Figura 8.

Figura 8. WLAN en modo ad-hoc

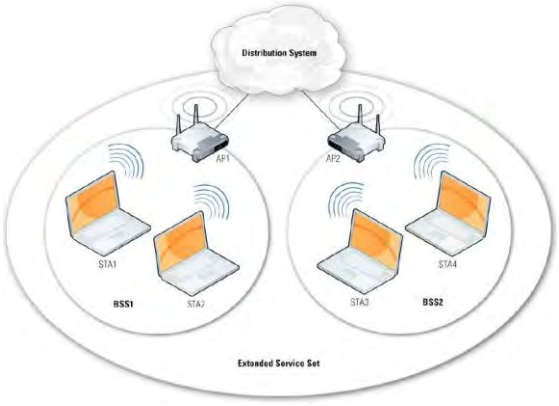
Fuente: SCARFONE, Karen; DICOI, Derrick; SEXTON, Matthew, TIBBS, Cyrus. Guide to Security Legacy IEEE 802.11 Wireless Networks [en Línea]: Recommendations of the National Institute of Standars and Technology. Gaithersburg: National Institute of Standars and Technology, 2008. [consultado el 17 de Marzo de 2016]. Disponible en Internet: http://csrc.nist.gov/publications/nistpubs/800-48-rev1/SP800-48r1.pdf

Este tipo particular de arquitectura presenta la ventaja de que es muy simple, rápido y barato de implementar, no necesita de infraestructura adicional ni mantenimiento y permite ser estructurado en diferentes topologías. En contraste, evidencia otro tipo de dificultades que interfieren negativamente en la comunicación, como por ejemplo, un volumen alto de estaciones en la misma red ad-hoc, o una distancia considerable entre las estaciones, así como obstáculos y objetos entre las mismas.

5.1.2 Infraestructura. Bajo este modo se implementan comúnmente las WLAN. En esta arquitectura, los clientes deben comunicarse con un "AccesPoint" para conectarse al sistema de distribución que permite hacer uso de los recursos de la red y conectarse a internet. El AP posibilita la comunicación entre todos los dispositivos en la red, pero también asegura que los paquetes enviados llegaran a los respectivos destinos y no necesariamente a todos los dispositivos.

En el modo infraestructura se pueden presentar los BSS (Basic Service Set) y los ESS (Extended Service Set). En BSS, se tiene un solo AP y una o varias estaciones, mientras que en ESS, se tiene un conjunto de BSS, de manera que existen múltiples "APs" que se conectan al sistema de distribución, que suele ser una red cableada. Este modo se ilustra en la Figura 9.

Figura 9. WLAN en modo infraestructura



Fuente: SCARFONE, Karen; DICOI, Derrick; SEXTON, Matthew, TIBBS, Cyrus. Guide to Security Legacy IEEE 802.11 Wireless Networks [en línea] Recommendations of the National Institute of Standars and Technology. Gaithersburg: National Institute of Standars and Technology, 2008. [consultado el 17 de Marzo de 2016]. Disponible en Internet: http://csrc.nist.gov/publications/nistpubs/800-48-rev1/SP800-48r1.pdf

5.2 PORTAL CAUTIVO

Un portal cautivo es una aplicación que se encarga de vigilar e interceptar el tráfico de paquetes con protocolo “HTTP”, para que ninguna solicitud sea aceptada hasta que el usuario se autentique, para lo cual es re direccionado a un sitio web permitido por el portal cautivo para tal fin, o hasta que complete alguna acción deseada por el administrador del portal.

Es así como se pueden definir al menos dos tipos de portales cautivos, uno que requiere de autenticación con usuario y contraseña para proceder con la navegación, y otro que requiere de la finalización de alguna tarea, ya sea esta dar click a un botón de aceptación de términos y condiciones, o ingresar a un enlace publicitario, o llenar una encuesta, etc. Algunos ejemplos de portales cautivos se exponen en la Figura 10 y Figura 11.

Figura 10. Portales cautivos con acceso mediante “login”

Fuente: elaboración propia

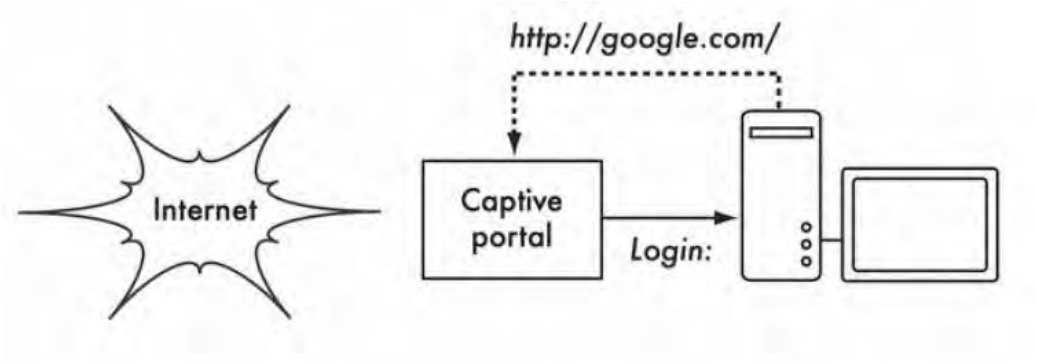
Figura 11. Portal cautivo de Cali Digital en las estaciones del MIO, con acceso mediante encuesta

Fuente: elaboración propia

El funcionamiento básico de los portales cautivos es bastante simple. Cuando un usuario desea hacer uso de los recursos de una determinada red debe disponer de una estación que pueda comunicarse y conectarse con esa red, tal como un computador, un portátil o un Smartphone.

Para poder hacer uso del servicio de internet que provee la red, el usuario debe proceder a ingresar a su navegador e intentar ir a algún sitio web, por ejemplo, “Google”. En este momento la aplicación de portal cautivo intercepta la petición “HTTP”, y responde al usuario con su propia página de internet, que le exige un registro mediante usuario y contraseña, como se observa en la Figura 12.

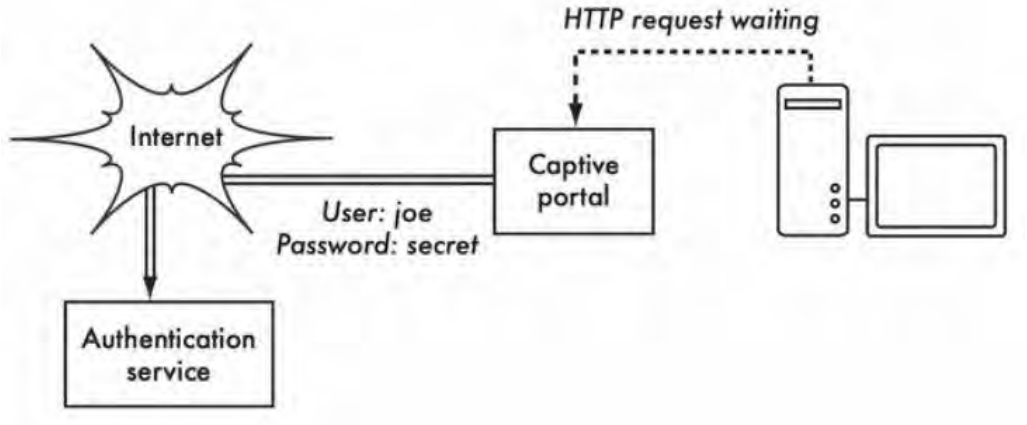
Figura 12.Solicitud HTTP de una estación para acceder a una pagina web



Fuente: Redes inalámbricas en los países en desarrollo [en línea]. 4 ed. Hacker Friendly LLC., 2013. 530 p. [consultado 04 de marzo de 2016]. Disponible en Internet: http://wndw.net/pdf/wndw3-es/wndw3-es-ebook.pdf

Posteriormente la aplicación de portal cautivo verifica las credenciales entregadas por el usuario, haciendo uso de un servidor de autenticación, que puede estar integrado en el "access point", o puede ser una máquina dentro de la red, o incluso, puede ser un servidor remoto accesible a través de internet, como es el caso de la Figura 13.

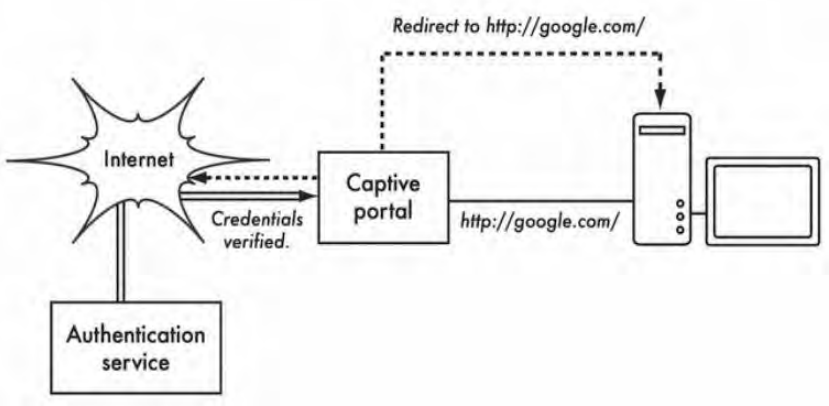
Figura 13.Verificación de credenciales mediante un servidor de autenticación



Fuente: Redes inalámbricas en los países en desarrollo [en línea]. 4 ed. Hacker Friendly LLC., 2013. 530 p. [consultado 04 de marzo de 2016]. Disponible en Internet: http://wndw.net/pdf/wndw3-es/wndw3-es-ebook.pdf

Para finalizar el proceso, si las credenciales son correctas el servidor de autenticación se lo confirma al portal cautivo, que se encarga de redireccionar al usuario a la página que había solicitado inicialmente, y libera los recursos de la red para que el usuario pueda navegar libremente, como se ilustra en la Figura 14.

Figura 14. Aceptación de credenciales y acceso a Internet



Fuente: Redes inalámbricas en los países en desarrollo [en línea]. 4 ed. Hacker Friendly LLC., 2013. 530 p. [consultado 04 de marzo de 2016]. Disponible en Internet: http://wndw.net/pdf/wndw3-es/wndw3-es-ebook.pdf

Cabe anotar que los portales cautivos están pensados para permitir o denegar el acceso a internet, pero no implementan ningún tipo de encriptación para los usuarios, luego, no pueden garantizar la protección contra ataques de suplantación, y a la larga, tampoco son un método efectivo para asegurar que sólo los usuarios legítimos están haciendo uso de los recursos de la red. Sin embargo, el portal cautivo ha resultado ser una solución óptima para necesidades de autenticación en redes públicas, donde otra clase de mecanismos de seguridad como WEP y WPA resultan ser inviables, puesto que la distribución de claves en un contexto público compromete en gran manera la efectividad de las mismas, por obvias razones.

Es así como los portales cautivos terminan entregando el nivel adecuado de seguridad, donde el servicio no es totalmente abierto, aunque tampoco lo sea totalmente impenetrable, pero funciona efectivamente en redes públicas presentes en lugares como aeropuertos, hoteles, cafés, bibliotecas, entre muchos otros; donde se espera recibir usuarios casuales, que utilicen los servicios de internet por periodos cortos de tiempo.

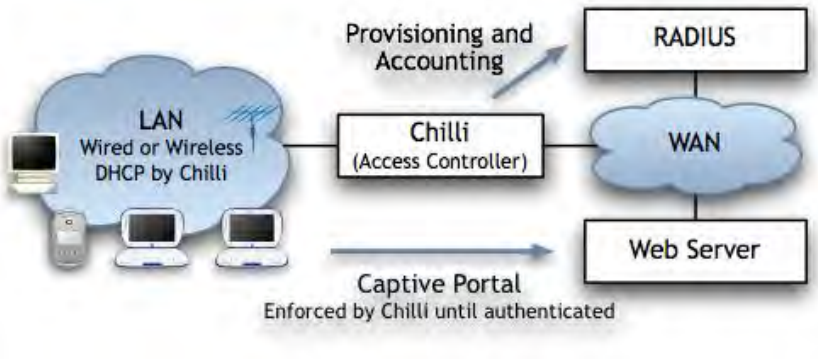
A continuación, se analizan algunas de las alternativas más relevantes de portal cautivo “open source” disponibles en “Internet”.

5.2.1 Covachilli. Es un software de código abierto basado en el popular proyecto ChilliSpot que actualmente no existe, a pesar de esto, los contribuidores y desarrolladores que mantienen “CoovaChilli” son los mismos del proyecto “ChilliSpot”. Su principal funcionalidad es controlar el acceso a una “LAN” mediante un portal cautivo en el cual se exige ingresar credenciales para acceder al servicio.

Este proyecto cuenta con herramientas insuficientes para la comunicación entre los miembros de la comunidad, en donde solo disponen de un correo electrónico, en el que exclusivamente se hace reporte de errores o preguntas puntuales acerca del funcionamiento del proyecto.

Para el correcto funcionamiento del portal cautivo, se debe contar con conexión a “Internet”, por lo menos un “Access Point”, un servidor de autenticación y un servidor web, los cuales pueden coexistir en el mismo servidor o ser usados de forma remota como se observa en la Figura 15.

Figura 15. Arquitectura de red para la implementación de un portal cautivo usando “CoovaChilli”



Fuente: CoovaChilli.[en línea] [Consultado 14 de agosto de 2016]. Disponible en Internet: https://coova.github.io/CoovaChilli/

El proyecto funciona en tres grandes módulos llamados “downLink”, “server radius” y “upLink”.

El modulo “downLink” gestiona las direcciones “IP” de los clientes mediante el protocolo “DHCP”, posteriormente valida si el usuario se encuentra autenticado en el “server radius”, en caso de no estarlo, es re direccionado al servidor de autentificación, por consiguiente, es llevado al portal cautivo.

Seguidamente se encuentra el “server radius”, el cual gestiona la autenticación de los clientes mediante credenciales, y valida el estado de cada usuario para notificar al módulo “downLink”. Si el proceso de autenticación fue valido, cambia el estado del cliente a autenticado y viceversa.

Finalmente, el modulo “upLink” se encarga de reenviar el tráfico de los clientes autenticados a otras redes, es decir provee el servicio de internet.

“CoovaChilli” otorga una documentación detallada en la sección “man pages” sobre el proceso de instalación de los módulos más relevantes, así como de los parámetros configurables por el administrador de la red, pudiendo ser ejecutado y gestionado desde el “command line” del servidor, finalmente, hace uso del demonio “syslogd” de Linux, en donde se almacenan y despliegan todos los errores y advertencias que ocurren durante la ejecución de “CoovaChilli”, con el fin de facilitar el proceso de depurado al momento de operar, e incluso de agregar nuevas funcionalidades [6].

5.2.2 EasyHotSpot. Es un portal cautivo de código abierto, creado con el fin de administrar redes mediante un sistema de facturación, control de tiempo y ancho de banda para los usuarios que necesiten acceder a una red tanto cableada como inalámbrica, siendo todos estos parámetros fácilmente configurables por el administrador de la red desde una interfaz desplegada en el navegador.

En su página web, “EasyHotSpot” cuenta con un blog en donde los usuarios plantean preguntas, y así mismo responden a estas, sin embargo, se evidencia la falta de participación por parte de la comunidad, debido a que entre cada publicación existe un lapso de tiempo considerable, incluso hasta de 4 años. Además de esto, también cuenta con un foro social, el cual se encuentra totalmente abandonado por parte de los desarrolladores de este proyecto, desplegando el error 404 en el navegador, es decir, el recurso solicitado ya no se encuentra disponible.

[6] CoovaChilli - man pages [en línea]. Coova.github.io. [Consultado 14 de agosto de 2016]. Disponible en Internet: http://coova.github.io/CoovaChilli/man-pages.html

Al momento de la instalación, se cuenta con una guía en donde se explica paso a paso el proceso necesario para instalar, configurar y ejecutar el portal cautivo. En esta guía, se especifican los siguientes requerimientos hardware mínimos para el correcto funcionamiento del proyecto: El servidor debe contar con un procesador “Pentium” 3, 512MB RAM, 5GB libres en el disco duro, 2 interfaces de red y los dispositivos convencionales de la arquitectura de red, “AP”, “switchs”, “hubs”, etc.

Para instalar el proyecto, inicialmente se debe tener instalado el sistema operativo Ubuntu, el cual se configura para arrancar desde el “CD-ROM” con la imagen .iso de “EasyHotSpot”, una vez hecho esto, se despliega una interfaz gráfica con las instrucciones de instalación.

Posteriormente, se configuran las interfaces de red, en donde “eth0” es la interfaz con conexión a internet, y “eth1” es la interfaz de salida a la subred protegida por el portal cautivo, tanto para clientes con conexión cableada como inalámbrica. En la Figura 16 se ilustra esta arquitectura.

Figura 16. Arquitectura de red para la implementación de un portal cautivo usando “EasyHotSpot”

Fuente: EasyHotSpot [en línea]. [Consultado 08 de agosto de 2016]. Disponible en Internet: http://easyhotspot.inov.asia/index.php/documentation#introduction

Una vez realizada la instalación, el portal cautivo funciona correctamente, los siguientes pasos serán configurar los parámetros existentes tales como; la dirección “url” del portal cautivo, la dirección “url” que se despliega después de la correcta autenticación del usuario, la dirección del servidor de autenticación o “server raduis”, los nombres de las interfaces de red, el precio por minuto que

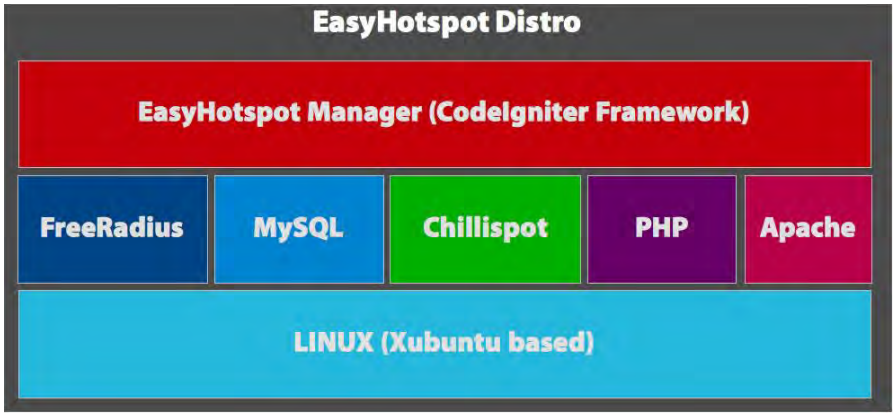
debe pagar un usuario al consumir el servicio, la velocidad de carga y descarga y los nombres de usuario que tendrán acceso a la red sin necesidad de autenticarse.

Finalmente, al momento que un usuario acceda a la red protegida por el portal cautivo, se despliega una interfaz en la cual el usuario debe ingresar sus credenciales para consumir el servicio normalmente.

A pesar de esta guía, actualmente no se cuenta con la documentación completa ni la "API" del proyecto, no obstante, se hace mención sobre la arquitectura software usada en la implementación de "EasyHotSpot", en donde se usan diferentes herramientas igualmente de código abierto para cada funcionalidad específica, presentando esta información por medio de la Figura 17.

El sistema operativo como se mencionó anteriormente es Ubuntu, una distribución de Linux, implementa "FreeRadius" para el servidor de autenticación, "MySql" como sistema de gestión de base de datos para lograr la persistencia de la información en el sistema, "Chillispot" para las funcionalidades del portal cautivo y controlador de los "AP", "PHP" como lenguaje de programación del "back-end", es decir, de lado del servidor, "Apache" como servidor web "HTTP", finalmente, se hace uso del "framework" "CodeIgniter", con el fin de gestionar y mantener de un modo más eficiente el código fuente de todo el sistema [7].

Figura 17. Arquitectura software de "EasyHotSpot"



Fuente: EasyHotSpot [en línea] [Consultado 09 de agosto de 2016]. Disponible en Internet: http://easyhotspot.inov.asia/index.php/documentation#introduction

[7] EasyHotspot - Documentation [en línea]. Easyhotspot.inov.asia, 2010. [Consultado 14 de agosto de 2016]. Disponible en Internet: http://coova.github.io/CoovaChilli/man-pages.html

5.2.3 GraseHotSpot. Es un proyecto gratuito y de código abierto para implementar un servidor “grase”, cuya principal funcionalidad es integrar un portal cautivo a la red. Este software es básicamente una integración de “CoovaChilli”, “FreeRadius” y “MySQL”, junto con otros paquetes personalizados por el creador de la solución, que permite implementar un portal cautivo de la manera más sencilla posible, sin tener que instalar o configurar cada componente individualmente.

Debido a su sencillez, no posee tantas funcionalidades como otros proyectos revisados anteriormente, ya que se concentra en la funcionalidad del portal cautivo, y se vuelve muy fácil de instalar y manejar a través de la interfaz web, desde la cual se pueden configurar, de ser necesario, parámetros para “Coova”, direcciones IP, plantillas para HTML que además son responsivas, entre otros; además de permitir la visualización de usuarios conectados y monitorear las sesiones.

Por esta misma razón, el proyecto no cuenta con una documentación muy extensa, pero si la necesaria para instalar correctamente el portal cautivo, configurar el hardware, instalar Ubuntu, y si se desea, configurar directamente “Coova” y “Freeradius”, puesto que, aunque la solución está diseñada para venir empaquetada, también es posible instalar cada uno de los paquetes por separado y configurarlos al antojo del usuario. Adicionalmente, también existe una página “Wiki” en “Github”, donde hay mucha más información sobre el proyecto, instalación y configuración de ciertos parámetros. En general parece una documentación suficiente para implementar el portal cautivo básico exitosamente.

Por otro lado, si resultan problemas o se quieren editar ciertos componentes, el proyecto cuenta con una comunidad disponible, para resolver los problemas y comentar las dudas. Los usuarios se pueden suscribir a las listas de correo, pero el principal medio de comunicación se da a través de un grupo el Google+, donde se puede observar que si bien no hay un volumen gigante de usuarios, como el caso de “PacketFence” o “Pfsense”, sí hay continuidad en el envío y respuesta de mensajes, por lo que se podría decir que el grupo esta satisfactoriamente activo. Vale recalcar que en muchas ocasiones, las dudas son resueltas por el propio creador del proyecto, que dedica tiempo a estar pendiente de los usuarios y ha comentado gran cantidad de “posts”.

Como información técnica, es importante saber que el proyecto está escrito mayormente en PHP, y posee algunos archivos “JavaScript” y “HTML”, naturalmente. La última versión es la 3.8, lanzada el 10 de enero de 2016, y esta soportada por al menos, Ubuntu 12.04 y “Debian 6”, tanto para máquinas de 32

como de 64 bits. No se menciona ningún requerimiento de hardware adicional, excepto por el hecho de que el servidor debe tener mínimo dos interfaces de red [8].

5.2.4 OpenWisp. Es una plataforma de software cuyo objetivo es la implementación de un servicio "wi-fi" completo y centralizado. Esta plataforma está siendo utilizada actualmente por alrededor de 15 regiones o provincias en Italia, siendo gestionada por las administraciones públicas de dichas localidades para brindar un servicio de "wi-fi" público y gratuito.

La versión estable de "OpenWisp" es en realidad un sistema con cinco aplicaciones, y cada una de ellas cumple funciones específicas. El primer módulo, es el de administración de usuarios y es el que se encarga de brindar credenciales a los usuarios y contabilizar el uso de los datos, además, permite visualizar gráficos y tablas relacionadas al historial del usuario. El segundo módulo es el Firmware, que debe instalarse en los "APs" que brindan el servicio de "wi-fi" público, para permitir la integración de los mismos con toda la plataforma, y el tercero es un módulo de administración que permite gestionar los "APs" de manera centralizada. El cuarto es un sistema de monitoreo geográfico, que verifica constantemente el estado de los "APs", para informar si están funcionando normalmente, a la vez que los despliega en un mapa de "Google Maps" de acuerdo a su ubicación física real. Por último, el quinto módulo es el administrador del portal cautivo, que se basa en "netfilter" de "Linux". Actualmente el equipo de trabajo de "OpenWisp" se encuentra desarrollando otra serie de módulos para agregar más funcionalidades a la plataforma, pero aún no han sido lanzados de forma estable.

La plataforma cuenta con documentación escasa. Los aportes más significativos están en "Github", en el apartado de "Wiki", donde se explica vagamente el proceso de instalación y algunas configuraciones, sin embargo, se apoyan en el uso de links para redirigir a otras páginas donde supuestamente hay información más detallada, pero las peticiones a estos sitios web devuelven un error 404, "Not Found". Adicionalmente, se encuentra disponible una página orientada únicamente a la lectura de documentación relevante sobre el proyecto, pero igualmente, es muy poca información y algunos apartados están incompletos. La comunidad es igualmente escasa, su único medio de comunicación es un grupo en Google+ donde apenas hay 18 temas.

La versión 1.3.4 de "OpenWisp" es la más reciente, y fue lanzada el 4 de noviembre de 2015. Debido a la cantidad de módulos disponibles para usar con

[8] About GRASE Hotspot [en línea]. GRASE Hotspot, 2011. [Consultado 5 de noviembre de 2016]. Disponible en Internet: https://grasehotspot.org/about/

“OpenWisp”, el proyecto en conjunto utiliza varios lenguajes de programación, principalmente “RubyOnRails”, y de forma secundaria, “Python”, “JavaScript”, y “Lua”. Debido a la poca documentación e información de instalación disponible, es difícil saber que requerimientos necesita la plataforma para funcionar, el único elemento que se puede destacar es que debe ser implementado sobre “OpenWRT”, una distribución de “Linux” para dispositivos como “routers” y “Access points” [9].

5.2.5 Packet Fence. Es una solución integral de control de acceso a la red, que implementa todo un sistema para realizar las tareas necesarias al momento de administrar una red y gestionar el acceso de los usuarios a la misma, además es gratis y de código abierto.

“Packet Fence” cuenta con una documentación bastante amplia, basada en guías por temáticas. De esta forma, se encuentran disponibles varias guías de longitudes variables que tratan tópicos específicos, como por ejemplo, la guía de administración, que es una de las más largas, explica el proceso de instalación, los modos de funcionamiento de “Packet Fence” y las opciones de configuración necesarias para correr el sistema correctamente; la guía de desarrollador, que incluye variables a tener en cuenta en el momento de que se presente la necesidad de modificar alguna funcionalidad o componente del sistema, para que se ajuste a las necesidades particulares del propietario; las guías de instalación rápida, que explican procesos de instalación para versiones de “Packe tFence” pre configuradas, y distintas guías cortas para integrar el sistema con otros módulos, como cortafuegos de otros proveedores.

Cuando la documentación no es suficiente, y existen dudas o casos particulares que requieren de mayor información, “Packet Fence” dispone de una comunidad altamente activa y grande, cuya comunicación se soporta en el uso de listas de correo (Mailing Lists), que se componen de tres direcciones de correo diferentes, especializadas en tres temas principales: Nuevos lanzamientos y anuncios públicos, desarrollo, y uso. Los usuarios pueden entonces suscribirse a las listas y mandar su mensaje, y adicionalmente, “Packet Fence” guarda todos los correos y los hace públicos, a través de un directorio de archivos online o visibles por fecha en “SourceForge”.

“Packet Fence” está escrito principalmente en “Perl” y “Javascript”, actualmente se encuentra en su versión 6.2.1, lanzada el 8 de Julio del 2016, y puede ser descargada de forma normal, o en versión “ZEN”, una versión fácil de instalar que

[9] OpenWISP: public wifi [en línea]. openWISP.org, 2016. [Consultado 6 de noviembre de 2016]. Disponible en Internet: http://openwisp.org/whatis.html

ya viene pre configurada. El sistema soporta instalación únicamente en Linux, en sus distribuciones “Red Hat Enterpise server” 6 y 7, “CentOS” 6 y 7, y “Debian” 7 y 8. También se ha podido instalar en “Fedora” y “Gentoo”, pero no hay documentación disponible para la configuración en dichos sistemas operativos. Como requisitos mínimos de instalación se necesita una máquina con un procesador Intel o AMD a 3GHz, 8GB de RAM, 100 GB de espacio en disco duro y una tarjeta de red, aunque se recomienda tener dos. En cuanto a dispositivos de red, el sistema necesita, naturalmente, una infraestructura de red funcional con todos sus artefactos correspondientes, como lo son los “router”, “switches” o “APs”.

“Packet Fence” tiene tres modos de implementación: en línea, fuera de banda, e híbrido. En el primer modo, “Packet Fence” funciona como “Gateway” para todo el tráfico de la red, luego no hay necesidad de realizar configuraciones complejas para las “AccessPoint” o “switches”, mientras que en el segundo modo, la red se segmenta utilizando “VLANs” y cada “AP” debe ofrecer una técnica para asignamiento de “VLAN”, mientras que “Packet Fence” gestiona todos los “APs”. El modo hibrido permite que los “APs” asignen “VLAN” con autenticación de “MAC” o a través del protocolo 802.1x, mientras que “Packet Fence” sigue funcionando en modo en Línea.

Este sistema tiene múltiples características, entre las que se pueden destacar: soporte para autenticación 802.1x a través del módulo de RADIUS tanto en redes cableadas como inalámbricas, soporte para voz sobre IP en “switches” de distintos proveedores, detección de actividades anormales y peligrosas en la red mediante la integración con sistemas de detección de intrusos como “Suricata” o “Snort”, implementación de portal cautivo para autenticación y remediación, aislamiento de red mediante la asignación de “VLANs”, control de acceso basado en roles, autenticación mediante cuentas de “Facebook”, “Google”, “GitHub”, y confirmación por correo, confirmación por mensaje de texto; contabilidad del ancho de bando consumido por los dispositivos conectados a la red, entre otras [10].

5.2.6 PepperSpot. Al igual que “CoovaChilli”, “PepperSpot” es una subversión mejorada y de código abierto de “ChilliSpot”, provee un portal cautivo para el control de acceso a una “LAN” con la capacidad de manejar direcciones “IPv4” e “IPv6”.

[10] Packet Fence: About [en línea]. Inverse, 2016. [Consultado 6 de noviembre de 2016]. Disponible en Internet: https://packetfence.org/about.html

Las herramientas que dispone “PepperSpot” para la comunidad y los desarrolladores son bastantes limitadas, debido a que no cuentan con; documentación, “API”, foros, blogs ni “FAQs”. En la página solo existe una guía de instalación en donde explican de manera básica las funcionalidades y sus respectivas configuraciones.

Para instalar correctamente “PepperSpot”, se debe tener un servidor web, un servidor de autenticación y un servicio de enrutamiento, los cuales pueden convivir en el mismo equipo o de manera remota, adicionalmente, solo funciona en servidores con sistema operativo basado en “Linux” con versiones del “Kernel” mayores a la 2.6.24.

La versión actual de “PepperSpot” es la 0.4 liberada el 22 de octubre de 2015, a pesar de ser reciente, notablemente hace falta documentación y herramientas para la comunicación entre la comunidad [11].

5.2.7 PfSence. Es una solución de software para seguridad en redes, donde se proveen todas las funcionalidades necesarias para tener un control avanzado sobre la gestión y seguridad de la red, de manera sencilla, estable, confiable y con un alto nivel de rendimiento. “PfSense” es, de forma más específica, una distribución de firewall gratis y de código abierto, basado en “FreeBSD”.

En cuanto a documentación, el proyecto se encuentra muy bien dotado de información, a través de una página “wiki” que posee distintas secciones para abordar variedades de temas, que van desde instalación hasta funciones avanzadas, explicaciones de “How-To” y preguntas frecuentes. Adicionalmente, existe un libro de “PfSense”, que se supone es la mejor fuente de información sobre el proyecto, ya que es una “guía definitiva” pero sólo puede ser adquirido al comprar una membresía de oro como cliente.

Por otro lado, se observa que el proyecto tiene una comunidad activa y grande, que utiliza el foro de “PfSense” como principal medio de comunicación. En este se pueden observar diferentes categorías dependiendo del tema principal, y cada categoría es básicamente una funcionalidad del software. Naturalmente, algunas categorías muestran más actividad que otras, pero de forma general, todas tienen

[11] PepperSpot – The next generation captive portal [en línea]. Thibault Vançon, 2012. [Consultado 7 de noviembre de 2016]. Disponible en Internet: http://pepperspot.sourceforge.net/index.php?n=Doc.UserDocumentation

un buen volumen de participación, y se han mantenido activas a través del tiempo. Otro medio de comunicación disponible, son las listas de correo, que muestran una actividad notablemente menor en relación al foro, pero que de igual forma, sigue activo, aunque con muchos menos participantes. La lista de correos dispone de 4 correos, cada uno para un tema específico: anuncios del proyecto, para informar sobre nuevas versiones o cambios significativos; soporte y discusiones, para resolver dudas o discutir problemas; desarrollo, para dar soporte a aquellas personas que escriben código y/o modifican directamente el software; recomendaciones de seguridad, para publicar distintos temas relacionados a cuestiones de seguridad.

En adición, el proyecto cuenta con un canal de IRC ("Internet Relay Chat") soportado a través de la plataforma de "Freenode", para comunicación instantánea entre distintos usuarios. Finalmente, el proyecto también hace uso de varias plataformas o redes sociales para poner en contacto a sus usuarios. Actualmente tienen grupos o páginas en: "Twitter", "Google+," "LinkedIn", "Spiceworks", "Reddit" y "Facebook".

El 25 de Julio de 2016 fue lanzada la última versión estable de "PfSense", la 2.3.2, lo cual prueba que tanto usuarios como desarrolladores aún están activos creando nuevas funciones y mejorando u actualizando las ya existentes. Los componentes de portal cautivo están escritos en "PHP", uno de los lenguajes de "back-end" más utilizados en el mundo. En cuanto a requerimientos, "PfSense" puede ser instalado tanto en arquitecturas x86 como x86-64, sobre servidores con un procesador con velocidad mínima de 500 MHz y RAM de 256 MB, aunque las especificaciones recomendadas son: 1 GHz de velocidad de procesador y 1 GB de memoria RAM, y adicionalmente, 1 GB de espacio en disco duro, únicamente para la instalación completa del software. Debido a que "PfSesnse" está basado en "FreeBSD", otras especificaciones diversas, como compatibilidad con interfaces de Ethernet, inalámbricas, seriales, dispositivos de sonido, video, USB, entre otros, son exactamente las mismas que las indicadas para "FreeBSD".

"PfSense" es muy fácil de configurar a través de la interfaz gráfica web, desde la cual se pueden cambiar todos los parámetros disponibles para una configuración personalizada que se adapte a las necesidades del administrador, sin necesidad de utilizar la línea de comandos en ningún momento.

Como cortafuegos, "PfSense" tiene varias características, entre las que destacan su capacidad de filtrar de acuerdo a la IP de origen o destino, de acuerdo al protocolo IP, de acuerdo al puerto de origen o destino para el protocolo TCP y UDP, o incluso, de acuerdo al sistema operativo del dispositivo cliente, con la ayuda de una herramienta llamada p0f. También permite asignar alias sobre IPs,

redes o puertos; desactivar todas las funcionalidades del firewall, también es posible para lograr que el proyecto funcione sólo como “router”.

Otras características interesantes incluyen: soporte “NAT” (Network Address Translation); “Multi-WAN”, para permitir el uso de varias conexiones a Internet; balance de cargas en el servidor; soporte para VPNs; DNS dinámico; servidor DHCP; monitoreo, através de gráficos RRD que despliegan información histórica sobre distintas variables, como uso de cpu, estados del firewall, tiempos de respuesta por parte de la interfaz WAN, entre otros.

El portal cautivo de este software permite realizar autenticación por medio de una base de datos local configurada previamente o a través de un servidor “RADIUS”, o puede simplemente no pedir ninguna credencial y desplegar el portal. Adicionalmente, permite configurar el número máximo de conexiones que puede tener un cliente por IP, la cantidad de tiempo que puede estar un cliente inactivo antes de ser desconectado por inactividad, y un tiempo límite después del cual todos los clientes serán desconectados. Otras características relevantes incluyen la opción de desplegar una ventana “pop-up” con un botón de desconectarse, para que el usuario puede salir voluntariamente de la red; la opción de definir una página de redirección para ser mostrada inmediatamente después de acceder al portal cautivo; el filtrado por “MAC”, las listas blancas por “MAC” o dirección IP, la posibilidad de mostrar el portal cautivo utilizando HTTP o HTTPS, según se desee, y un gestor de archivos para cargar imágenes que deseen ser utilizadas en el portal [12].

5.2.8 WifiDog. Es una alternativa de solución de código abierto para el control de acceso a una red inalámbrica pública, mediante el despliegue de un portal cautivo para la correcta autenticación de los usuarios. Este proyecto se creó con el propósito de remplazar los portales cautivos existentes en el año 2004, los cuales no cumplían las características necesarias para un óptimo desempeño tanto para los usuarios finales como para los administradores de la red. Los desarrolladores de “WifiDog”, se concentraron principalmente en la creación de un sistema personalizable, el cual pueda ser configurado de acuerdo a las necesidades del administrador de la red, en eliminar por completo los “pop ups” y contenidos invasivos, como en el caso de los portales cautivos implementados con “noCat”, finalmente y una de las características más relevantes en ese entonces, brindar un servicio que se ejecute correctamente sin ninguna dependencia software y sin importar el navegador en el lado del cliente.

[12] PfSense: Main page [en línea]. Rubicon Communications, 2015. [Consultado 8 noviembre de 2016]. Disponible en Internet: https://doc.pfsense.org/index.php/Main_Page

En la página web de “WifiDog”, existe una sección donde listan todos los usuarios de este proyecto, tanto personas particulares como empresas y organizaciones, en donde Europa y Norte América registran la mayor cantidad de usuarios. Además, cuentan con un correo electrónico el cual está disponible para hacer consultas y reportar errores, finalmente en el contexto de comunidad y soporte, disponen de una sección de “FAQs” en donde se encuentra gran cantidad de preguntas con sus respectivas respuestas acerca del proyecto, tales como; desarrolladores, términos y condiciones de uso, instalación, configuración y funcionamiento del proyecto. Sin embargo, no poseen espacios fundamentales para la correcta comunicación de la comunidad como por ejemplo foros y blogs.

“WifiDog” está diseñado para correr sobre sistemas operativos basado en Linux e incluso sobre “routers” de marca “Linksys”. Cuenta con un archivo de configuración llamado “.po”, en donde se puede especificar el lenguaje en el cual será desplegada toda la información de la interfaz tanto del administrador, como del consumidor del servicio tal como se muestra en la esquina superior izquierda de la Figura 18.

Figura 18. Interfaz gráfica de un portal cautivo implementado con “WifiDog” por “Pizzédélic”

Fuente: WifiDog [en línea]. [Consultado 10 de Agosto de 2016]. Disponible en Internet: http://dev.wifidog.org/wiki/Screenshots

Adicionalmente, cuenta con un sistema de dialogo cliente-servidor, en donde la comunicación se ejecuta mediante un “ping” en lugar de usar “JavaScript” como en la mayoría de portales cautivos, permitiendo que la información sea desplegada correctamente sin importar el tipo de dispositivo ni sistema operativo que acceda a la red protegida. “WifiDog” posee dos modos de uso en cuanto a la autenticación del usuario. El primer modo de uso llamado “Splash only mode”, el usuario es re direccionado al portal cautivo, pero no debe introducir credenciales para acceder

al servicio, y en el segundo modo de uso llamado “Normal mode”, ver Figura 18, para que el usuario pueda acceder normalmente a la red, debe introducir correctamente las credenciales que le hayan sido asignadas por el administrador de la red.

“WifiDog” permite desplegar estadísticas en tiempo real sobre el consumo de ancho de banda de cada cliente, los usuarios que más han frecuentado el servicio, la cantidad de clientes concurrentes, grafico de consumo de datos por hora, semana y mes, la información de cada nodo y creación de una lista blanca para permitir el acceso sin autentificación a clientes específicos.

La arquitectura de este proyecto se divide principalmente en dos módulos; el cliente o “Gateway” y el servidor de autenticación o “auth server”.

El cliente es un demonio, es decir, un proceso que se ejecuta en segundo plano esperando a que ocurran determinados eventos para ofrecer servicios, una de sus características es que no cuenta con una interfaz gráfica, por lo tanto, se usa una consola de comandos para realizar la respectiva instalación, configuración y administración [13].

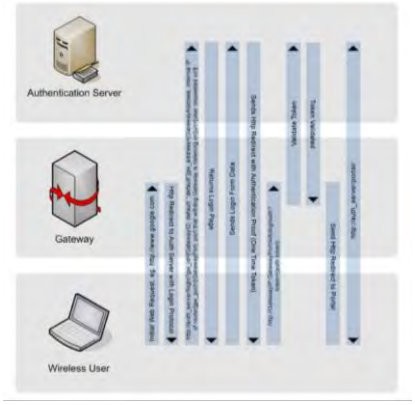
Para que el cliente funcione correctamente, debe correr sobre el sistema operativo “Linux”, tener instalado y compilado “NetFilter” en el “kernel” con el fin de interceptar y manipular los paquetes de red, y por último, debe tener instalado el “firewall” llamado “iptables”, debido que el cliente con base en la información extraída del servidor de autenticación se comporta como un orquestador, jugando con las reglas del “firewall” para permitir o denegar a un usuario el acceso a la red.

El segundo módulo es un servidor de autenticación, el cual se encarga de gestionar y validar las credenciales de cada usuario mediante una aplicación web montada sobre “Apache” que puede ser configurada de acuerdo a las necesidades del administrador de la red, o mediante diferentes plantillas que la comunidad ha desarrollado y compartido. Este módulo está codificado en “PHP” usando la versión 8.0 de “PostgreSQL” como gestor de la base de datos para persistencia del sistema. Dichos módulos se ilustran en la Figura 19.

[13] Doc – WiFiDog [en línea]. WiFiDog, 2010. [Consultado 10 de Agosto de 2016]. Disponible en Internet: http://dev.wifidog.org/wiki/doc

“WifiDog” cuenta con la documentación necesaria para el proceso de instalación e integración de los módulos mencionados anteriormente, sin embargo, no existe la “API” del proyecto.

Figura 19. Diagrama de flujo del proceso de autenticación implementado por “WifiDog”



Fuente: WifiDog [en línea]. [Consultado 10 de Agosto de 2016]. Disponible en Internet: http://dev.wifidog.org/wiki/doc/developer/FlowDiagram

5.2.9 ZeroShell. Es una distribución con cantidad de funcionalidades que se ejecutan sobre “Linux”, dichas funcionalidades están orientadas hacia servidores y dispositivos integrados destinados a proporcionar servicios en una “LAN”.

“Zeroshell” está disponible en forma de “CD” o de imagen de “Compact Flash”, es decir, para su ejecución no hace falta ser instalado en un disco duro, debido a que se puede operar directamente desde el “CD-ROM” o desde un puerto “USB”, sin perder ninguna de sus funcionalidades. Además, puede ser configurado y administrado utilizando el navegador web o directamente desde sus archivos.

Gran parte de sus funcionalidades trabajan de manera nativa incluyendo el portal cautivo, sin embargo, la parte visual en el navegador, está escrita usando dos herramientas de desarrollo web descritas anteriormente, “HTML” y “JavaScript”.

Esta distribución de “Linux” cuenta con un foro en su página web, el cual dispone de diferentes herramientas para la comunicación entre usuarios.

Si el usuario simplemente desea leer las publicaciones existentes, lo podrá hacer sin necesidad de crear una cuenta, no obstante, un usuario puede registrarse y contar con más herramientas para la participación en el foro tales como: búsqueda de información avanzada, mensajes privados entre miembros de la comunidad, postulación de temas en el foro, capacidad de comentar publicaciones existentes; finalmente, ser parte de un grupo de usuarios, en donde se notifica la participación de cada uno de los miembros del grupo.

En cuanto a las actualizaciones de “Zeroshell”, sus desarrolladores crearon un sistema de actualización automático, el cual permite al usuario actualizar cualquier versión de forma sencilla. La última actualización de esta distribución se realizó el día 4 de julio de 2016, “Zeroshell” 3.6.0. Esta versión tiene ajustes de problemas con en las funcionalidades del portal cautivo, “VPN” y equilibrio de tráfico o “Net Balancer”.

Para el correcto funcionamiento de “Zeroshell”, es necesario contar un sistema operativo de x86 o x64 bits, en cuanto a las características del hardware, por lo menos debe tener un procesador que trabaje a una frecuencia de 233 MHz, 96MB de RAM, dos tarjetas de red y un lector de “CD-ROM ATA” o adaptador “ATA”. En cuanto a la arquitectura de red, esta distribución funciona con dispositivos convencionales como “routers”, “swtiches” y “Access points”.

Se mencionan las principales funcionalidades generales de “Zeroshell” y las de su portal cautivo.

- Servidor RADIUS para proporcionar una autenticación segura y la gestión automática de las claves de cifrado para el Wireless. El software usado para la implementación de esta funcionalidad es “FreeRadius”.

- Portal Cautivo para apoyar el inicio de sesión web, en redes cableadas e inalámbricas, implementado de forma nativa, sin necesidad de utilizar software como “NoCat” o “Chillispot”.

- QoS (Quality of Service) y de gestión de tráfico para controlar el tráfico en una red congestionada, con la posibilidad de administrar el ancho de banda mínimo, máximo y asignar una prioridad a una clase de tráfico.

• Servidor proxy de “HTTP” que es capaz de bloquear las páginas web que contienen virus. Esta característica se implementa con la solución antivirus “ClamAV” y el servidor proxy “HAVP”.

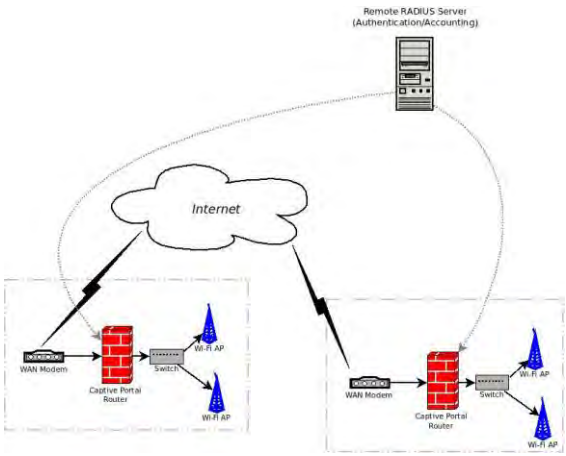
• Enrutador con rutas estáticas y dinámicas (RIPv2).

• Servidor “DHCP” con la posibilidad de asignar una “IP” fija en función de la dirección “MAC” del cliente.

• “X509”, entidad emisora de certificados para la emisión y gestión de certificados electrónicos.

El portal cautivo implementado por “ZeroShell” consiste en una puerta de enlace, que en este caso será el “router” de una subred, el cual redirige las peticiones “HTTP” y “HTTPS” a un servidor de autenticación, en cual despliega al usuario una página en donde deberá introducir las credenciales correctas, finalmente, permite al usuario hacer uso del servicio normalmente. En la Figura 20 se ilustra la arquitectura de red para la implementación del portal cautivo.

Figura 20. Arquitectura de red para la implementación de un portal cautivo usando “ZeroShell”



Fuente: ZeroShell [en línea]. [Consultado 07 de agosto de 2016]. Disponible en Internet: http://www.zeroshell.org/hotspot-router/

Cabe resaltar, que con respecto a la Figura 20, el servidor de autenticación y la puerta de enlace pueden coexistir en el mismo equipo.

La puerta de enlace del portal cautivo puede trabajar en dos modos distintos:

- **Modo enrutado:** La puerta de enlace debe ser necesariamente el "router" de la subred protegida por el portal cautivo.

- **Modo Bridge:** La puerta de enlace protegida funciona como puente entre dos o más interfaces, esto permite al servidor "DHCP" que un cliente tenga siempre la misma dirección IP en el área protegida y en la red sin protección.

"Zeroshell" implementa una capa de seguridad, en donde todas las interacciones entre el navegador web del usuario y el servidor de autenticación se cifran utilizando https para evitar que las credenciales se pueden capturar en la red. Además, los clientes son identificados por sus direcciones IP y "MAC", sin embargo, estos dos parámetros pueden ser fácilmente falsificados. Con el fin de resolver este problema, el servidor de autenticación renueva periódicamente las credenciales, usando el algoritmo de cifrado "AES256" el cual no pueden ser fácilmente falsificados antes de su expiración, todo esto, teniendo en cuenta que se ejecuta de manera transparente para el usuario final del portal cautivo.
A Partir de la versión 1.0.beta6 de "ZeroShell", el Portal Cautivo es capaz de autenticar a los usuarios mediante el uso de certificados X.509. La última característica permite utilizar las "SmartCard" para el inicio de sesión de red con Portal Cautivo.

Es posible declarar una lista blanca de clientes, es decir, no se les solicitaran credenciales de autenticación para acceder a la red. También, es posible definir una lista de servicios gratuitos proporcionados por servidores externos que los clientes pueden utilizar sin necesidad de autenticación.

La página de acceso web del portal cautivo y el lenguaje a utilizar durante la fase de autenticación se puede configurar por el administrador.

Finalmente, no se cuenta con la documentación ni la "API" de la distribución, sin embargo, los desarrolladores asignaron un espacio en donde los usuarios voluntariamente ayudan con la documentación del proyecto, asimismo, existe una sección llamada "FAQs" o preguntas frecuentes por sus siglas en inglés, en donde la comunidad postula preguntas y respuestas a temas específicos. A pesar de

esto, la comunidad no se ha pronunciado considerablemente en la ayuda para la documentación del proyecto, por lo que en la actualidad, se encuentra notablemente incompleto y desordenado [14].

5.3 PUBLICIDAD ONLINE

El internet ha sido una herramienta que ha revolucionado radicalmente la forma en que las personas se relacionan con la información, y ha causado especial impacto en los esquemas de comunicación, permitiendo conectar personas que se encuentran a distancias muy lejanas, en tiempo real. En el contexto publicitario, Internet ha posibilitado la democratización de la publicidad, ya que se permite publicidad de todo tipo de anunciantes sin importar su tamaño, su relevancia o su naturaleza, alcanzando las mismas posibilidades de efectividad para todos. Es por ello, que actualmente Internet es uno de los medios más explotados para el despliegue de publicidad que tiene unas características particulares al ser visualizado en páginas web.

5.3.1 Formatos de publicidad en Internet.

- Pop - ups. Es publicidad digital que se despliega en una ventana independiente a la del sitio web accedido. Busca llamar la atención mediante el factor sorpresa, pues este tipo de formato “salta” de la pantalla y suele tener colores muy llamativos o efectos de animación.

- Banners. Es la típica publicidad estática, que sólo es una imagen ubicada en algún lugar altamente visible dentro del sitio web, generalmente en la parte superior, aunque se pueden desplegar en diferentes ubicaciones y en variados tamaños y proporciones.

- Interstitial. Es la publicidad que se despliega al ingresar a una página web mientras que esta carga. Suele ubicarse en primer plano y ocultar el contenido del sitio web, y generalmente debe cerrarse mediante una acción.

- Botones. Son utilizados para acceder a otra clase de contenido que anunciante quiere que el usuario visualice. Suelen ser los menos intrusivos porque ocupan

[14] Documentación y procedimientos sobre Zeroshell [en línea]. Fulvio Ricciardi, 2005. [Consultado 7 de agosto de 2016]. Disponible en Internet: http://www.zeroshell.net/es/documentation/

muy poco espacio y se suelen adaptar efectivamente al diseño de las páginas, pero no otorgan mucha información.

- Enlaces de texto. Son el tipo de publicidad más simple, compuesta únicamente por palabras y un enlace, sin ninguna clase de estilo, diseño, o fondo en especial.

- Spots online. Básicamente son anuncios de publicidad comunicados de forma audiovisual, en formato de vídeo. Suelen ser de tamaños relativamente pequeños y de corta duración. Pueden presentarse incrustados en la página web o de forma flotante en una ventana independiente.

5.3.2 Estrategias de mercadeo digital.

- Behavioral targeting. Es una metodología de marketing que se concentra en estudiar al usuario a partir de su comportamiento en línea, para poder brindarle anuncios que sean de su interés real.

Esta metodología cuenta con cuatro fases. La primera es la de creación de perfiles, donde se obtiene gran cantidad de información con respecto al usuario y su comportamiento en internet, qué le gusta, cuáles son sus intereses, qué cosas busca, a qué le da click, etc. En la segunda fase, se realiza una segmentación, para agrupar estos perfiles de usuario que presentan similitudes. En la tercera etapa se entregan los contenidos digitales especializados para el tipo de usuario y en la última, se realiza un monitoreo constante de la relación entre el usuario y en contenido que se le entrega, para evaluar la efectividad del mismo y realizar mejoras.

- Redes sociales. El auge de las redes sociales le ha permitido al marketing explorar posibilidades de publicidad en un ambiente que es percibido como seguro y confiable, permitiendo a los anunciantes reforzar y promover su producto y marca, en un entorno social, donde los usuarios tienen la posibilidad de indicar directamente qué es lo que les gusta, a la vez que son capaces de observar lo que les agrada a sus amistades, de manera que es más fácil generar efectos en cadena a través de estos medios.

- Advertainment. Es una nueva estrategia basada en la combinación de contenidos publicitarios con actividades de entretenimiento, de forma que el

producto pueda estar en la mira del público objetivo y pueda ser presentado de una manera que terminara enganchando al usuario. En esta estrategia, se intenta vender un juego, que es percibido como entretenimiento y no publicidad, luego el usuario difícilmente mostrará actitudes reacias, y la parte de marketing y publicidad se articula al juego en forma de narrativa o contenidos subliminales.

- Marketing Viral. Esta estrategia pretende que los mismos usuarios se tomen la tarea de difundir la publicidad, pues esta tiene algún contenido adicional que ha sido diseñado para ofrecer un valor agregado. De esta forma, un pequeño grupo de usuarios conforman un núcleo emisor que reenvía la publicidad, y luego los receptores también colaboran en los procesos de transmisión y difusión de la información, de modo que la publicidad termina llegando exponencialmente a muchos usuarios.

5.4 TENDENCIAS WEB

En una época en donde el usuario no presta atención a los mensajes publicitarios debido al bombardeo constante y no solicitado de miles de marcas que luchan por un espacio en su mente. En una época en donde los hábitos de vida se han transformado por la incursión de nuevas tecnologías –computadores, móviles, consolas, reproductores de audio- y en una época en donde el usuario ha tomado el control absoluto de los mensajes que crea y recibe; era apenas lógico que la publicidad y el mercadeo comenzaran a moverse en nuevos terrenos que conectaran a través de experiencias interactivas, al esquivo consumidor de la generación C –The connected generation

Como menciona la autora Silvia Angélica Vargas su tesis “Nuevas formas de publicidad y mercadeo en la era digital”, es necesario crear diferentes estrategias de publicidad en la web, debido a que este medio está saturado. Para lograr con este propósito, se requiere conocer las tendencias en cuanto a las herramientas usadas para implementar estrategias de publicidad interactivas en la web, tales como navegadores, plataformas.

Por la parte de los navegadores web, según los datos tomados en enero y febrero de 2016 por “w3schools”, los navegadores más usados son; “Google Chrome”, “FireFox”, “Internet Explorer”, “Safari”, finalmente, “Opera”. Las cifras son ordenadas en la

Figura **21**.

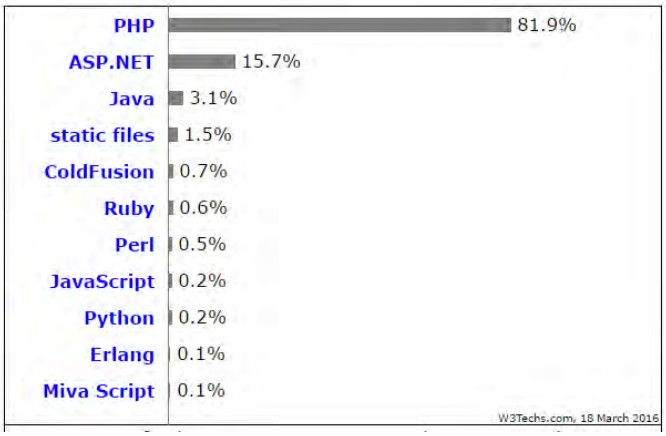
Figura 21. Navegadores más usados en los primeros dos meses del 2016

2016	Chrome	IE	Firefox	Safari	Opera
February	69.0 %	6.2 %	18.6 %	3.7 %	1.3 %
January	68.4 %	6.2 %	18.8 %	3.7 %	1.4 %

Fuente: Browser Statistics [en línea]. W3Schools. [Consultado 11 de marzo de 2016]. Disponible en Internet: http://www.w3schools.com/browsers/browsers_stats.asp

Adicionalmente, es de vital importancia conocer los lenguajes de programación más usados por el lado del servidor, en este caso, según las estadísticas de "w3techs", las cuales son actualizadas diariamente, "php" y "asp.net", están en los primeros lugares, como se enseña en la Figura 22.

Figura 22. Estadísticas de los lenguajes de programación más usados para montar servicios web



Fuente: Usage of server-side programming languages for websites [en línea]. W3Techs. Web Technology Surveys. [Consultado 11 de marzo de 2016]. Disponible en Internet: http://w3techs.com/technologies/overview/programming_language/all

6. METODOLOGÍA

Para el desarrollo de este proyecto se caracteriza la investigación como tecnológica aplicada, puesto que pretende transformar conocimientos teóricos para generar conocimiento útil, que tendrá como finalidad su aplicación en la solución de un problema dentro de un contexto manifestado en la vida real, a través de mediaciones tecnológicas que permitirán el desarrollo o diseño de un producto, o la reconfiguración de algún proceso, mediante el cual se optimizan procesos empleados en situaciones actuales, o se generan nuevas formas de competitividad al interior de un mercado.

6.1 ETAPA DE EXPLORACIÓN

Durante la etapa de exploración se pretende adquirir conocimientos muy generales sobre los portales cautivos y realizar un listado de las aplicaciones de portales cautivos de código abierto disponibles para descarga gratuita en Internet.

Sobre cada uno de los elementos de esta lista se realizará una caracterización, mediante la recolección de datos procedentes de la documentación disponible en la web. Posteriormente se realizará un análisis comparativo de las características encontradas, con el objetivo de determinar cuál elemento se adapta mejor al propósito y escala de este proyecto.

6.2 ETAPA DE ESTUDIO Y ANÁLISIS

A partir de los resultados del análisis comparativo, se seleccionará una sola aplicación de portal cautivo para utilizar durante el resto del proyecto.

Mediante la documentación previamente obtenida, se instalará dicha aplicación de portal cautivo en un computador, que hará las veces de servido, cuyo sistema operativo dependerá de las características propias del portal. Una vez instalado y probado su correcto funcionamiento, se iniciará un proceso de estudio en el cual se examinará detenidamente el código fuente, con el objetivo de entender apropiadamente el diseño y la implementación técnica del mismo.

6.3 ETAPA DE MODIFICACIÓN

Dado el conocimiento de las funciones del portal cautivo por defecto, así como la estructura del código, se diseña una estrategia para modificar el código de tal manera que se le otorgue a la aplicación de portal cautivo las nuevas características que le permitirán ejecutar funciones de reconocimiento de "AccesPoint", así como de administración de contenidos personalizables, propios para cada zona inalámbrica definida por la cobertura del AP.

6.4 ETAPA DE IMPLEMENTACIÓN Y PRUEBAS

Una vez ejecutadas las modificaciones pertinentes, se realizará una prueba para verificar que el sistema en conjunto funciona como está proyectado.

Para tal fin, se hace necesario contar con mínimo dos dispositivos móviles que se puedan conectar a la red, dos "AccessPoint" y un "switch". En esta parte, se justificará la selección de los dispositivos utilizados, basándose en las características de los mismos y su utilidad dentro de la prueba planteada.

Adicionalmente, es necesario disponer de diferentes contenidos web para ser desplegados. Estos serán diseñados y creados por los ejecutores del proyecto, de manera que se puedan explorar diferentes posibilidades y estrategias para la visualización e interacción con las publicidades digitales.

Finalmente se sacarán conclusiones a partir del desarrollo de la prueba, que permitirán identificar tanto errores como aciertos, en el diseño e implementación de las nuevas funcionalidades del portal cautivo, así como la utilidad y alcance del nuevo conocimiento generado.

7. IDENTIFICACIÓN DEL ESCENARIO Y POTENCIALES USUARIOS

En este apartado del documento, se hace énfasis en la identificación de un posible escenario en el cual estén implicados tanto los usuarios potenciales del portal cautivo como el administrador del mismo.

7.1 ESCENARIO

Uno de los posibles escenarios reales en el cual la solución propuesta en este proyecto puede ser aplicada, es en el aeropuerto internacional Alfonso Bonilla Aragón localizado en el corregimiento de Palmaseca del municipio de Palmira. Este aeropuerto es gestionado por la sociedad concesionaria llamada Aerocali S.A, la cual se encarga de la administración de los recursos, infraestructura y servicios aeroportuarios con el fin de garantizar su operación de forma continua, eficiente e innovadora [15].

En este aeropuerto, existe un lugar llamado zonas comunes de muelles nacionales e internacionales, en donde diariamente transita un elevado número de personas, adicionalmente, en dichas zonas hay una gran cantidad de establecimientos en los que se comercializan diferentes tipos de productos y servicios tales como alimentos, recuerdos, productos de belleza, equipajes, cambios de moneda, entre muchos otros. Asimismo, Aerocali ofrece el servicio de “Wi-Fi” de manera gratuita [16], convirtiendo este espacio en un escenario con las características ideales para la implementación de la solución propuesta.

Finalmente, la persona encargada de esta sociedad gestora de recursos es el gerente de servicios aeroportuarios, el cual puede presentar particular interés hacia la solución de portales cautivos múltiples, ya que es una buena estrategia de monetizar el recurso del “Wi-Fi” añadiendo valor y cambiando la experiencia de usuario al momento de acceder a “Internet” desde los muelles nacionales e internacionales, ya que el contenido los portales cautivos son aplicaciones “web” de toda índole que pueden ser programadas para tener un alto grado de interactividad con el usuario y ofrecer de una forma no convencional los mismo

[15] Nuestra empresa [en línea]. Aerocali, 2017. [Consultado 05 de junio de 2017]. Disponible en Internet: http://www.aerocali.com.co/aerocali/nuestra-empresa/

[16] Telecomunicaciones [en línea]. Aerocali, 2017. [Consultado 05 de junio de 2017]. Disponible en Internet: http://www.aerocali.com.co/telecomunicaciones/

productos y servicios que actualmente existen en los establecimientos comerciales.

7.2 USUARIOS DEL PORTAL CAUTIVO

Posterior a la identificación del escenario, se procede a reconocer las características más relevantes de los usuarios del portal cautivo, con el fin de sintetizar dicha información en un perfil de usuario.

Tal como se ilustra en la Figura 23, se destaca que los usuarios del portal cautivo deben tener contacto previo con la tecnología y contar mínimo con un dispositivo móvil bien sea “Smartphone” o “Tablet”, debido a que es necesario tener habilidades y conocimientos básicos para acceder al servicio de “Internet”, adicionalmente, los usuarios del portal cautivo carecen del servicio de datos móviles a causa de diversos motivos, como por ejemplo, cuestiones económicas o simplemente preferencia. Además, los usuarios del portal cautivo son personas que frecuentan lugares en donde predomine el comercio. Finalmente, se enmarca la edad de los potenciales usuarios en un amplio rango comprendido desde los 13 hasta los 65 años de edad, puesto que, en este rango las personas usan frecuentemente dispositivos móviles en espacios públicos.

Figura 23. Perfil del usuario del portal cautivo

Fuente: Elaboración propia.

7.3 ADMINISTRADOR DEL PORTAL CAUTIVO

Finalmente, se procede a identificar el perfil de usuario del administrador del portal cautivo. Esta persona es la encargada de gestar los contenidos web publicitarios que se despliegan en los diferentes portales cautivos de cada área inalámbrica, por lo tanto, es necesario realizar la distinción de dicho perfil.

Como se muestra en la Figura 24, se detalla que el administrador del portal cautivo debe ser una persona con conocimientos básicos de tecnología y debe contar con un computador, debido a que, si bien el proceso de asignación de contenidos de los portales cautivos es relativamente sencillo, es necesario tener conocimiento sobre los formatos básicos que imperan en la "web" para así mismo tener la capacidad de realizar dicha configuración sin romper los enlaces existentes entre los respectivos "index.html" y sus recursos, adicionalmente, deben tener nociones básicas de conceptos de red, especialmente acerca de direccionamiento "IP" para asignar los rangos de cada contenido. Además, debe ser una persona con la suficiente disponibilidad de tiempo para realizar los cambios del contenido de cada portal cautivo si así se requiere. Finalmente, se enmarca la edad del potencial administrador del portal cautivo en el rango comprendido desde los 25 hasta los 40 años de edad, puesto que, en este rango existe una variada cantidad de perfiles bien sean tecnólogos o técnicos que cumplen con los rasgos esenciales del administrador del portal cautivo.

Figura 24. Perfil del administrador del portal cautivo

Fuente: Elaboración propia.

8. ANÁLISIS DE ALTERNATIVAS DE PORTAL CAUTIVO

En esta sección, se analizan las distintas alternativas de portal cautivo, teniendo en cuenta una lista de criterios de selección con su respectiva importancia, fundamentados en las características que influyen considerablemente en la elección de la alternativa con los atributos más favorables que encajen con el propósito y desarrollo del proyecto.

8.1 DEFINICIÓN DE CRITERIOS DE SELECCIÓN

A continuación, se hará mención de los criterios de selección en orden descendente de acuerdo a su importancia, la cual es representada por un valor porcentual que varía en función del nivel de significancia sobre las características deseables en el portal cautivo a seleccionar. Cabe mencionar que los criterios de selección son cualitativos, y serán transformados a variables cuantitativas, asignándoles un valor en una escala del 1 al 10, de acuerdo a la comparación del mismo atributo entre cada alternativa. Una vez realizado esto, se organizan los valores de cada criterio de selección para cada alternativa, con el fin de ser comparados. Posteriormente, se observa el valor absoluto de cada criterio, es decir, el valor escalado de acuerdo a su peso. Finalmente, los valores de los criterios para cada alternativa serán ponderados, permitiendo conocer el portal cautivo que cumple en mayor medida con los atributos necesarios para dar paso al siguiente objetivo específico.

8.1.1 Documentación. En primer lugar con un peso del 30%, se tiene la documentación del portal cautivo. El objetivo general de este proyecto, es añadir una funcionalidad específica a un portal cautivo mediante la intervención de su código fuente, de acuerdo a la naturaleza modular, numero de protocolos, entidades y atributos implicados, es necesario partir de un proyecto que se encuentre documentado en la mayor medida, puesto que al contar con una descripción de cada módulo, se posibilita la intervención, análisis y modificación de este; por tal razón, es el criterio de selección con mayor peso.

8.1.2 Comunidad. Posteriormente, se encuentra la comunidad con un peso del 20%. La comunidad es un atributo clave al momento de elegir, ya que, por medio de herramientas de comunicación, por ejemplo blogs, foros, videos, entre otras, se informa a los demás usuarios acerca de toda clase de situaciones involucradas como errores, explicaciones, configuraciones, etc., permitiendo efectuar un trabajo colaborativo entre grupos de personas trabajando para el mismo fin. Adicionalmente, permite agregar valor en cuanto a la ayuda y soporte técnico del proyecto entre los mismos usuarios.

8.1.3 Simplicidad y modularidad del software. El siguiente criterio de selección es la simplicidad del software con un 20%. Por conveniencia del desarrollo de este proyecto, el ideal es partir de un "software" con la menor cantidad de módulos y funcionalidades, es decir, que únicamente cuente con el portal cautivo, ya que al momento de intervenir el código fuente, es de gran ayuda evitar dependencias entre servicios y funcionalidades tal como lo plantea "Uncle Bob" en su libro "Clean Code", alta cohesión y bajo acoplamiento significa que un programa debe estar dividido en la mayor cantidad de módulos, pero evitar al máximo las dependencias entre estos mismos. En conclusión, entre mayor sea la cantidad de funcionalidades, mayor es la cantidad de módulos y dependencias, en consecuencia, es más compleja la intervención y modificación del código.

8.1.4 Lenguaje del código fuente. Se ha asignado un peso del 15% al lenguaje del código fuente. Debido al gran número y diversidad de lenguajes de programación que pueden ser implementados para desarrollar un proyecto, es común encontrar lenguajes de programación y estructuras inusuales, que requieren un nivel mucho más técnico y minucioso para realizar una instrucción que otros, por lo tanto, es necesario el establecimiento de este criterio, debido a que evita la selección de una alternativa desarrollada con herramientas que cuenten con poca documentación, desconocidas por la comunidad y que no encajen con nuestros conocimientos previos. Por esta razón, los lenguajes de programación como "Java", "C#", "PHP", "JavaScript" y tecnologías de desarrollo web como "CSS" y "HTML", tendrán mayor peso al momento de seleccionar una alternativa.

8.1.5 Requerimientos adicionales. Los requerimientos adicionales se encuentran en el siguiente lugar con un 10%. En este campo, se validan todos aquellos requerimientos de orden hardware y arquitectura de red, en donde se analizan las características mínimas necesarias para el correcto funcionamiento de las alternativas, frecuencia del procesador, memoria “RAM”, memoria “ROM”, cantidad de tarjetas de red y sistema operativo. En cuanto a la arquitectura de red, se valida si es necesaria la implementación de dispositivos de red no convencionales, como en el caso de “TextBlue” o “ZeroTruth”, y adicionalmente, si se establecen requerimientos en cuanto a la distribución de cada módulo, es decir, si permite la convivencia de la totalidad del proyecto en el mismo equipo, o si es necesario el uso de diferentes ordenadores.

8.1.6 Última actualización. Finalmente, el criterio de selección con menor peso es la fecha de la última liberación con un 5%. Este criterio de selección permite establecer la actividad que los desarrolladores ejercen sobre el proyecto, debido a que existen alternativas que no han sido actualizadas en un tiempo considerable, este mismo factor, influye en la contemporaneidad de las herramientas empleadas para el desarrollo del portal cautivo. Por lo tanto, es el criterio de selección con menor peso debido a que la compatibilidad de las herramientas para el desarrollo web es relativamente alta permitiendo integrar tecnologías recientes.

8.2 SELECCIÓN DE ALTERNATIVA DE PORTAL CAUTIVO

Una vez identificados los criterios de selección, se procede a realizar la respectiva asignación de valores para cada alternativa de portal cautivo, y posteriormente, se ponderan estos valores de acuerdo al peso de cada criterio. De este modo, se obtiene el portal cautivo con las características más adecuadas para la continuación de este proyecto.

Cuadro 1. Alternativas "vs" criterios de selección con su respectivo valor

Criterios de selección / Alternativas	Documentación	Comunidad	Simplicidad y modularidad	Lenguaje del código fuente	Requerimientos adicionales	Última actualización
Easyhotspot	4	3	8	8	7	4
Zeroshell	6	5	3	3	7	10
Wifidog	5	2	4	7	8	3
Packetfence	9	7	2	5	2	10
CoovaChilli	2	2	8	3	9	10
OpenWisp	3	2	6	4	4	9
Pfsense	8	9	2	8	5	10
Pepperspot	2	1	8	2	6	9
Grase hostspot	7	6	8	9	8	10

Fuente: Elaboración propia.

Cuadro 2. Alternativas "vs" criterios de selección con su respectivo valor absoluto, y total de cada alternativa

Criterios de selección / Alternativas	Documentación	Comunidad	Simplicidad y modularidad	Lenguaje del código fuente	Requerimientos adicionales	Última actualización	Total
Easyhotspot	1.2	0.6	1.6	1.2	0.7	0.2	5.5
Zeroshell	1.8	1.0	0.6	0.45	0.7	0.5	5.05
Wifidog	1.5	0.4	0.8	1.05	0.8	0.15	4.7
Packetfence	2.7	1.4	0.4	0.75	0.2	0.5	5.95
CoovaChilli	0.6	0.4	1.6	0.45	0.9	0.5	4.45
OpenWisp	0.9	0.4	1.2	0.6	0.4	0.45	3.95
Pfsense	2.4	1.8	0.4	1.2	0.5	0.5	6.8
Pepperspot	0.6	0.2	1.6	0.3	0.6	0.45	3.75
Grase hostspot	2.1	1.2	1.6	1.35	0.8	0.5	7.55

Fuente: Elaboración propia.

Después de realizar una comparación cuantitativa sobre las características de cada opción de portal cautivo, se observa que "GraseHotSpot" obtuvo el puntaje más alto, con un total de 7.55. No obstante, se puede percibir que la diferencia del total entre la alternativa seleccionada y "Pfsense" no es significativa, por lo tanto, vale la pena aclarar que los motivos principales que condujeron a la elección de

“GraseHotSpot”, fue la simplicidad y modularidad del software, ya que “Pfsense” cuenta con una mayor cantidad de componentes y funcionalidades aumentando considerablemente la complejidad al momento de realizar modificaciones, aunque este cuente con una mayor comunidad y mejor documentación.

Este resultado se obtiene de acuerdo a los criterios definidos y a la forma en que este portal cautivo, en particular, se ajusta a las necesidades del proyecto. Su principal ventaja es que es una solución que viene empaquetada y prácticamente pre configurada, lo cual garantiza un significativo ahorro de tiempo, puesto que no es necesario instalar y configurar individualmente cada módulo requerido para el funcionamiento del portal, ahorrando también posibles fallas que se puedan presentar en el futuro debido a configuraciones inconsistentes para el caso de aplicación que se quiere tratar. Acompañada a esta característica, se debe resaltar el módulo de administrador del portal cautivo, el cual a través de una interfaz web, permite gestionar fácilmente la mayor parte de las funcionalidades del software, haciéndolo ideal para aprender, explorar y modificar, teniendo en cuenta que un gran volumen de información por aprender y de archivos por modificar representan una significativa carga cognitiva que puede no resultar favorable para el desarrollo del proyecto, cuando el objetivo de este no es en ningún momento especializarse en portales cautivos y las numerosas funcionalidades que se le pueden agregar.

Otra ventaja identificada, fue el uso de PHP como principal lenguaje de programación, lo cual facilita el entendimiento del código fuente a los desarrolladores del proyecto, y en general, a cualquier persona familiarizada con el desarrollo de “back-end”, puesto que este es el lenguaje que se utiliza en la mayoría de servicios web, por ende, su manipulación no será una tarea que implique intentar aprender todo un lenguaje desde cero ni buscar demasiados recursos complejos. De la misma forma, la contemporaneidad de la solución provee seguridad al momento de trabajar con sus componentes, que estarán en su gran mayoría actualizados, bien documentados y compatibles con las tecnologías actuales.

Finalmente, se tiene una solución que es fácilmente instalable sin necesidad de utilizar recursos de máquina de forma significativa, como en otros proyectos más complejos, y se cuenta con una documentación y comunidad acorde a la proporción y alcance del proyecto, lo cual debe ser suficiente para llegar a instalar exitosamente el portal cautivo, y obtener respuestas de otros internautas en caso de que sea necesaria una consulta para solucionar un problema bastante especifico, que no haya sido tratado puntualmente en la documentación.

9. ARQUITECTURA Y COMPONENTES DE “GRASEHOTSPOT”

9.1 INSTALACIÓN DE GRASEHOTSPOT

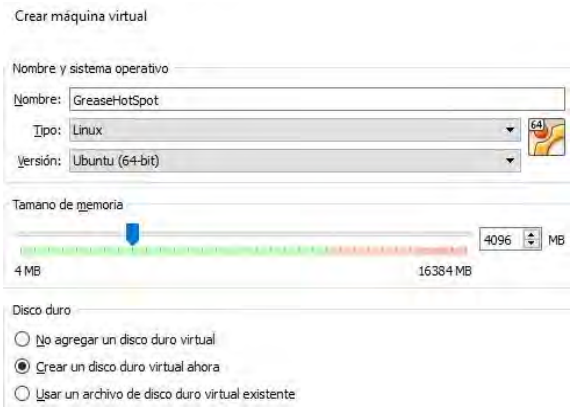
Una vez seleccionado el portal cautivo de código abierto más favorable para este proyecto, en función de los criterios de selección, se describe a grandes rasgos el proceso de instalación del portal cautivo.

Por factores como la versatilidad, recursos y eficiencia, se realiza la instalación del servidor del portal cautivo en un entorno de virtualización de máquinas llamado “VirtualBox”, en el cual es posible instalar e instanciar máquinas en un amplio espectro de sistemas operativos invitados, al interior del ambiente virtual del sistema operativo anfitrión, en donde cada uno de estos, cuenta con su propio ambiente virtual. En este caso particular, en una maquina con sistema operativo “Windows 10”, se virtualiza la maquina servidor con el sistema operativo “Ubuntu 14.04 LTS Desktop”.

Con el fin de realizar la virtualización correctamente, es necesario conocer los requisitos de instalación del servidor “GraseHotSpot” para prestar los recursos mínimos que demande, tales como; procesador x86 con una frecuencia de 700 MHz, memoria RAM de 512 MB, disco duro de 5 GB y dos tarjetas de red. Vale la pena mencionar que para lograr mejor desempeño, es recomendable asignar una mayor cantidad de recursos de los mínimos establecidos. Por otro lado, el sistema de “GraseHotSpot” garantiza su correcto funcionamiento sobre el sistema operativo “Ubuntu 14.04 LTS” o “Debian 7”, sin necesidad de instalar componentes adicionales, diferentes a los paquetes preconfigurados de “Grase”.

Para esto, se crea una nueva máquina virtual como se muestra en la Figura 25, en donde se ingresa el nombre de la misma, el tipo y versión del sistema operativo, y se asigna la cantidad de memoria RAM máxima que la máquina virtual puede consumir, en este caso particular, 4096 MB.

Figura 25. Creación e ingreso de datos básicos de la máquina virtual



Fuente: Elaboración propia.

Posteriormente, se define la ubicación en donde serán almacenados los datos de la máquina virtual en la maquina anfitrión, el tamaño y tipo de almacenamiento de la memoria, en este caso, 10 GB reservadas dinámicamente, es decir, el tamaño del archivo de la máquina virtual dependerá de lo que se encuentre almacenado actualmente hasta que llegue al límite de 10GB.

Una vez definidos estos parámetros, se procede a asignar la imagen del sistema operativo Ubuntu 14.04 LTS Desktop para iniciar la máquina virtual. Desde este punto, se continua con la instalación normalmente como si se tratase de una maquina física, en la cual se indican variables como el idioma, ubicación geográfica, nombre de usuario y contraseña.

El resultado de este proceso, es una máquina virtual con sistema operativo Ubuntu que posee las prestaciones necesarias para comportarse como servidor de este proyecto.

Finalmente, se configura “GraseHotSpot” ejecutando los siguientes comandos a través del terminal.

- Se descarga el último paquete del repositorio con el comando wget y el enlace.

$wget http://packages.grasehotspot.org/pool/main/g/grase-repo/grase-repo_1.6_all.deb

- Se instala el paquete descargado. $ sudo dpkg -i grase-repo_1.6_all.deb

- Se instalan los componentes del portal cautivo, freeRadius y openVpn con las configuraciones por defecto. $ sudo apt-get install grase-www-portal grase-conf-freeradius grase-conf-openvpn

Para comprobar que el proceso de instalación fue exitoso, ingresamos a través del navegador a la dirección http://10.1.0.1/grase/radmin, la cual debe desplegar la interfaz de administrador por defecto.

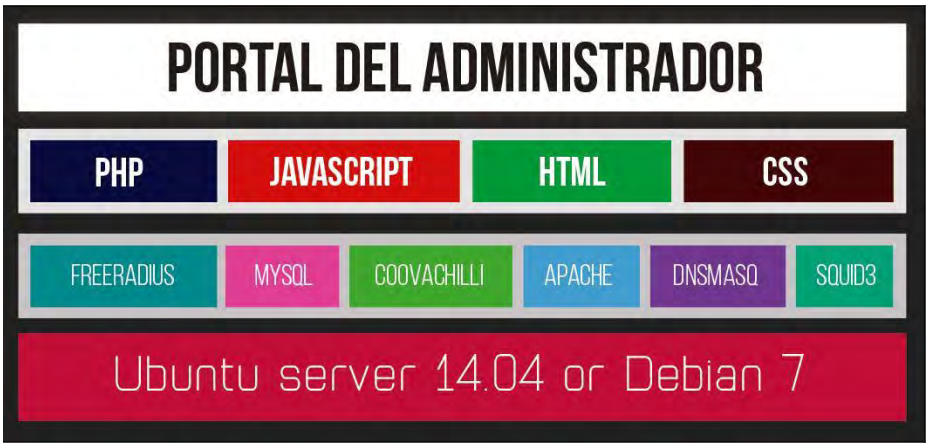
9.2 DESCRIPCIÓN DE FUNCIONAMIENTO DE "GRASEHOTSPOT"

"GraseHotSpot, al igual que la mayoría de alternativas de portal cautivo, está compuesto de múltiples módulos, los cuales se caracterizan por estar bajo la licencia de uso "GNU" o licencia publica general [17], en donde el uso del software es libre, incluso para proyectos de carácter comercial.

Cada uno de estos módulos, se encarga de ejecutar diferentes funcionalidades dedicadas, y al ser integrados, permiten crear un ecosistema de aplicaciones software para un portal cautivo complejo con múltiples funciones y configuraciones tal como se observa en la Figura 26.

[17] El sistema operativo GNU [en línea]. Free Software Foundation, Inc, 2016. [Consultado 30 de enero de 2017]. Disponible en Internet: https://www.gnu.org/licenses/licenses.es.html#GPL

Figura 26. Modulos de "GraseHotSpot"



Fuente: Elaboración propia.

9.2.1 FreeRadius. Es un software que implementa un servicio de autenticación, basado en el protocolo de red llamado “RADIUS” o “Remote Authentication Dial-in User Server”, el cual permite autenticar, autorizar y gestionar las cuentas de los usuarios. Por lo tanto, “FreeRadius” permite proteger una red bien sea inalámbrica o cableada, mediante el uso de credenciales de usuario, controlando por completo quien accede a la red, que permisos tiene sobre esta y gestionar los recursos consumidos[18].

9.2.2 MySql. Es un motor para la gestión de bases de datos “SQL” o “Structured Query Language”, el cual se caracteriza por estar compuesto de tablas relacionadas entre sí de forma jerárquica, en donde cada tabla almacena sus propios registros.

MySql cuenta con una amplia gama de herramientas para la administración de bases de datos, brindando diferentes beneficios, tales como; rapidez, administración simple, gran escalabilidad, posibilidad de trabajar en diferentes plataformas, personalización en los formatos de las tablas, entre otras. Estos beneficios son la razón de su alta popularidad, contando con grandes clientes en la industria de tecnología de la información, como por ejemplo “GitHub”, “Facebook”, “PayPal”, “YouTube”, “Wikipedia”, “Twitter” etc. [19]

[18] MARQUÉS, Guillermo. AAA y FreeRadius: Lulu, 2016. p 38.
[19] MySql Customers [en línea]. Oracle Corporation, 2017. [Consultado 30 de enero de 2017]. Disponible en Internet: https://www.mysql.com/customers/

9.2.3 CoovaChilli. Es un “software" de portal cautivo basado en el proyecto ChilliSpot. Su principal funcionalidad, es interceptar el tráfico de red “http”, para controlar el acceso a una red mediante un portal cautivo, en el cual se exige ingresar credenciales para acceder al servicio normalmente. Para más detalle, ver la sección 5.2.1.

9.2.4 Apache. Es un servidor web muy robusto y de alto desempeño, que implementa el protocolo de transferencia “HTTP”, para la creación y despliegue de páginas y servicios web. Para esto, apache crea un proceso que está constantemente a la espera de peticiones hechas por los clientes, y responde con el contenido solicitado, el cual, es finalmente interpretado y desplegado por los navegadores web. [20]

9.2.5 DnsMasq. Es un paquete de “Linux” que permite implementar fácilmente un servidor DNS y DHCP.

9.2.5.1 Servidor DNS. Se encarga de establecer la relación entre los nombres de dominio y las direcciones IP de los equipos de una red [21]. Los servicios de DnsMasq, permiten resolver nombres de dominio tanto de forma directa como inversa, es decir, conoce la IP de un nombre de dominio y conoce el nombre de dominio de una IP.

9.2.5.2 Servidor DHCP. Se encarga de asignar de forma dinámica las direcciones IP de una red a los clientes que la soliciten, usando un rango de direcciones determinado [22].

[20] About the Apache HTTP server Project [en línea]. The Apache Software Foundation, 1997. [Consultado 30 de enero de 2017]. Disponible en Internet: https://httpd.apache.org/ABOUT_APACHE.html

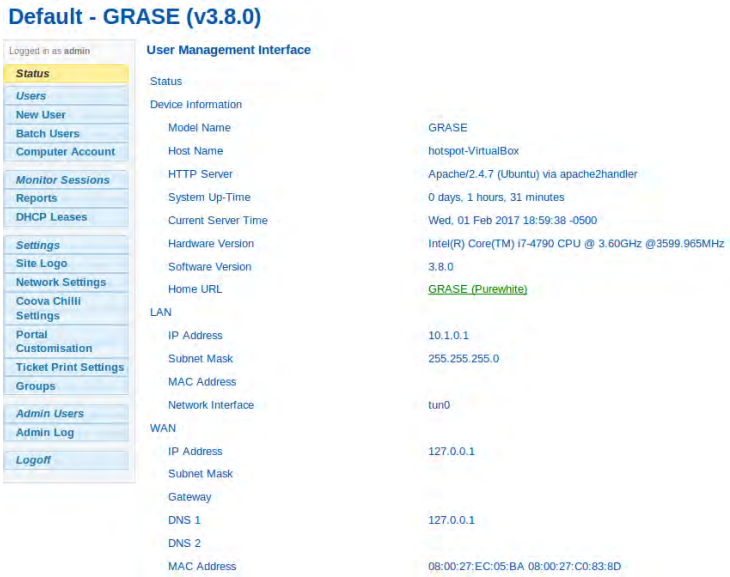
[21] ANDREU, Joaquín. Servicios en red. Madrid: Editex, 2011. P. 32.

[22] Ibid. P.25.

9.2.6 Squid3. Es un servidor proxy y web caché que se ejecuta sobre entornos basados en Unix. El funcionamiento de estos servicios consiste en actuar como un intermediario entre un cliente (navegador web) e internet, para limitar el acceso a ciertos recursos potencialmente maliciosos o no deseados por el administrador de la red, esta funcionalidad, la ejecuta un subcomponente específico llamado “Squid guard”. Además, Squid3 almacena los contenidos más solicitados en memoria, con el fin de desplegarlos rápidamente al momento de ser requeridos dentro de la red “LAN” [23].

9.2.7 Portal del administrador. Es una aplicación web servida por “Apache” en la dirección 10.1.0.1/grase/radmin/loginconfig, ver Figura 27, en la cual, el administrador de la red puede gestionar las múltiples configuraciones del portal cautivo y monitorear los recursos consumidos por los usuarios activos.

Figura 27. Portal del administrador de "GraseHotSpot"



Fuente: Elaboración propia.

Las características más relevantes del portal administrativo son:

• Creación y monitoreo de los usuarios. La aplicación permite desplegar gran variedad de información, incluyendo el estado de cuenta (válida o expirada), el

[23] SAWANT, Uday. Ubuntu Server Cookbook. Birmingham: Packt Publishing, 2016. P. 42.

tiempo restante para que la cuenta expire, la fecha de la última conexión, el tiempo total de uso de internet, la cantidad total de datos usados y su respectivo limite, que es previamente asignado por el administrador al momento de crear el usuario. Además, permite realizar diferentes filtros en el despliegue de datos, como por ejemplo, mostrar únicamente las sesiones activas.

- Configuración de elementos visuales como por ejemplo el nombre de la ciudad, de la empresa, el logo desplegado, el idioma de la interfaz y el texto del “footer” de la página de “login”.

- Asignación de interfaces de red tanto para la “LAN” como para la “WAN”, es decir, para definir cuál interfaz estará conectada a los clientes, y cual a “Internet”.

- Asignación de una máscara y una dirección IP para el servidor de la red “LAN”.

- Configuración del módulo “Coova Chilli”, en donde podemos establecer un tiempo de cierre sesión por defecto para usuarios inactivos o “Default sesión idle timeout”, con el fin de no consumir recursos innecesariamente.

- Asignación de un rango de direcciones ip para el servidor “DHCP”.

- Creación de una lista blanca de sitios web o dominios accesibles para usuarios no autenticados.

- Modificación del nombre de la red inalámbrica o “SSID”.

- Almacenamiento y despliegue de todos los cambios hechos por el administrador, como por ejemplo, creación de usuarios, o modificaciones en la interfaz gráfica del portal cautivo.

Para que los módulos mencionados anteriormente sirvan una aplicación web de portal cautivo, inicialmente deben estar prendidos los servicios de cada uno, posteriormente, al momento que algún usuario acceda a la red inalámbrica, “CoovaChilli” lo detecta, puesto que constantemente monitorea el tráfico “http” por el puerto 80 del servidor, gracias a “DnsMasq” y “Apache”, “CoovaChilli” redirige al

usuario al archivo “hotspot.php” el cual carga la página web del portal cautivo que se le despliega al usuario. En esta página el usuario debe autenticarse ingresando sus credenciales, las cuales son enviadas por “FreeRadius” mediante el protocolo “RADIUS”, se realiza una comparación de las credenciales ingresadas con las existentes en la base de datos “MySql”, dependiendo de esta comparación, se permite o no, el ingreso del usuario a la red inalámbrica.

Como se evidencia, la filosofía de los portales cautivos es proteger una red inalámbrica mediante autenticación, es por esta razón, que los módulos dependen entre sí y cumplen una funcionalidad importante en este proceso, por lo tanto, no pueden ser removidos del servidor.

10. MODIFICACIÓN DE “GRASEHOTSPOT” PARA AGREGAR FUNCIONALIDAD DE DISCRIMINACIÓN DE CONTENIDOS

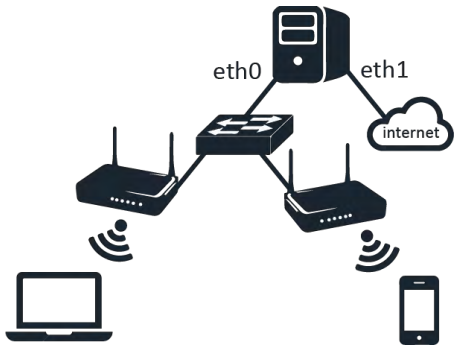
En este capítulo se describen las alternativas consideradas con sus respectivas implicaciones para la modificación del código fuente de “GraseHotSpot”, finalmente, se explica el proceso llevado a cabo para filtrar el contenido dependiendo del “Access Point”.

10.1 TOPOLOGÍA DE LA RED

Cualquier modificación que se realice sobre “GraseHotSpot” debe tener en cuenta su configuración predeterminada, no sólo en términos de software, sino de hardware y topología de red.

Por defecto, “GraseHotSpot” viene compilado para trabajar sobre un servidor con “Ubuntu” o “Debian”, que debe tener mínimo dos interfaces de red, comúnmente nombradas “eth0” y “eth1”. La interfaz “eth1” es la que el servidor utiliza para acceder a internet, mientras que la “eth0” es usada para proveer el portal cautivo y eventualmente, conexión a internet, a los clientes. Esto significa que los clientes deben conectarse a esta interfaz, pero dependiendo del número de clientes que se deseen conectar y su ubicación física, se necesitan otros dispositivos para asegurar el alcance de la red. Por esta razón, lo más común es conectar un switch a “eth0”, que permita a su vez disponer de múltiples puntos de acceso inalámbricos desde los cuales los clientes finales establecerán conexión con el servidor, como se muestra en la Figura 28.

Figura 28. Topología de la red para las pruebas de implementación del portal cautivo



Fuente: Elaboración propia.

10.2 FILTRADO DE TRÁFICO DE RED

Bajo la topología expuesta anteriormente, todas las peticiones de los clientes finales van a parar al servidor, quien es el encargado de procesarlas y desplegar el portal cautivo, sin embargo, estas peticiones primero tienen que viajar por la red y deben pasar obligatoriamente por un “AP”. Para lograr que el servidor pueda mostrar diferentes contenidos dependiendo del “AP” por el cual pasó la petición, es primordial que el servidor pueda identificar y diferenciar cada uno de los “APs” disponibles en la red, para posteriormente realizar un filtrado del tráfico, que le permita discriminar cómo entregar los contenidos correctamente, es decir que mediante el filtrado de tráfico el servidor sea capaz de determinar el “AP” al cual el usuario está conectado, y por ende le puede enviar el contenido publicitario correspondiente a ese “AP”.

A continuación, se exponen diferentes métodos que se exploraron inicialmente para conseguir tanto la identificación del “AP”, como el filtro correspondiente.

10.2.1 Filtrado por dirección Mac. Las direcciones “MAC” han sido usadas por mucho tiempo como los identificadores físicos de los dispositivos de red dentro de una LAN, lo que las convierte en un excelente criterio para utilizar como base en el desarrollo de un filtro, de tal manera que se pueda reconocer desde cuál “AP” viene el tráfico, utilizando la dirección física de su interfaz de red. El filtrado por “MAC” es especialmente conveniente en este caso particular porque no implica ninguna modificación sobre los “APs” o su comportamiento, además, las direcciones “MAC” son permanentes y estáticas, siempre serán igual para un mismo dispositivo, brindando una opción de identificación totalmente estable e inequívoca.

Para evaluar la factibilidad de este filtro en el proyecto, era necesario realizar pruebas con la topología de red en la Figura 28, en primer lugar, para verificar que la dirección “MAC” de un “AP” es efectivamente accesible desde el servidor, y en segundo lugar, para explorar formas de integrar esta dirección al funcionamiento del portal cautivo mediante el diseño de un filtro.

La prueba se realizó utilizando el servidor, un “switch” y un solo “AP”, puesto que por el momento, el objetivo principal era simplemente obtener la “MAC” del “AP”, que está conectado a la “LAN”, pero no directamente a una interfaz de red del servidor. De esta forma, se utilizó la herramienta “wireshark” desde el servidor para analizar todos los paquetes de distintos protocolos que llegaban al mismo.

Durante la ejecución de la prueba, fue posible visualizar que el servidor efectivamente recibía peticiones “HTTP” de parte del cliente, y adicionalmente, se obtienen otras de protocolos “TCP” y “DNS”, sin embargo, para todos los casos, la capa de Ethernet siempre muestra intercambio entre las mismas dos direcciones MAC, pertenecientes a la interfaz de red del servidor y al dispositivo móvil, que es el cliente. En otras palabras, todo lo que sucede con los otros dispositivos, como el “AP” y el “switch” es invisible para el servidor, pues establece un canal de comunicación donde sólo le es relevante el cliente; esto teniendo en cuenta que el “AP” utilizado tiene desactivadas todas sus características de router y funciona en modo “AP”, es decir no asigna direcciones “IP”, pues tiene el servicio de “DHCP” desactivado y tampoco realiza ninguna clase de traducción de direcciones (NAT).

De manera alternativa, si se pudiese extraer la relación entre el puerto físico y la dirección “MAC” de cada interfaz del “switch”, sería posible filtrar el tráfico de red desde el servidor, ya que con esta información se sabe en qué puerto del “switch” está conectado cada “AP”, por lo tanto, se podría identificar de donde provienen las peticiones. Es por esta razón, que se identifica el protocolo “SNMP”, el cual permite extraer esta información luego de configurar los dispositivos de red para poder monitorear el “switch” desde el servidor, no obstante, implementar este filtrado de red presenta serios inconvenientes al momento de manipular la información para ser validada desde la capa de aplicación, para más detalle sobre el filtrado a través del protocolo “SNMP”, ver el capítulo 11. Trabajos futuros.

El resultado obtenido en esta prueba, donde se evidenció la imposibilidad de extraer la “MAC” del “AP” desde el servidor, va en concordancia con la configuración y el funcionamiento de “CoovaChilli”, puesto que este módulo, principal responsable de interceptar el tráfico “HTTP” y controlar el acceso a la red, hace uso tanto de la “IP” como de la “MAC” del cliente, y de hecho, guarda y utiliza estos atributos en variables que llama “Framed-IP-Address” y “Calling-Station-ID”, respectivamente, más en ningún momento utiliza ningún atributo o variable para obtener información de otros dispositivos distintos al del cliente.

10.2.2 Filtrado por SSID. El “SSID” (Service Set Identifier) es un conjunto de caracteres que se asignan a los dispositivos de red con capacidades inalámbricas para identificar su red. Todos los “APs” deben tener un “SSID”, y es posible que dos o más “APs” tengan el mismo, pero este identificador es configurable, puede ser modificado en cualquier momento accediendo al sistema de configuración del dispositivo.

Inicialmente, se consideró al “SSID” como una alternativa para realizar un filtro, teniendo en cuenta la posibilidad de asignar distintos “SSIDs” a distintos “APs”, sin

embargo, una lectura más profunda del funcionamiento y utilidad del SSID permitió descartar rápidamente esta opción, debido a que el “SSID” es utilizado exclusivamente por el “AP” y sus clientes para establecer una conexión inalámbrica, además, este atributo tiene su existencia y alcance únicamente dentro de la red inalámbrica que provee el “AP”, lo cual quiere decir que bajo ninguna circunstancia sería posible que otro dispositivo de red, como otro “AP”, conociera un “SSID” distinto al propio, y de igual forma, un servidor que se encuentra conectado vía “Ethernet” a un “AP” tampoco podría conocer el “SSID” del mismo.

10.2.3 Filtrado por dirección IP. Consiste en obtener la dirección “IP” del cliente para identificar desde que “AP” se están realizando las peticiones, para posteriormente, desplegar el contenido del portal cautivo asignado para dicho “AP”.

Inicialmente, con los dos “AP” en modo “Router”, es decir, se encargan de realizar las asignaciones de direcciones “IP” a los clientes y proporcionar conectividad hasta el “switch”, se configuran los servidores “DHCP” de cada uno, definiendo diferentes rangos de direcciones “IP” dentro de la red, con el fin de ser recuperadas en el servidor para conocer de cual “AP” se encuentra conectado el cliente e individualizar el tráfico para desplegar el portal cautivo correspondiente.

Una vez se realizó esta configuración, se sometió a pruebas de funcionamiento, en donde cada “AP” asignaba efectivamente a los clientes las direcciones “IP” dentro del rango especificado en el “DHCP”, sin embargo, se evidencio a través de la herramienta de análisis de tráfico de red “Wireshark”, que las direcciones “IP” asignadas por los “AP” no son tenidas en cuenta por el servidor “DHCP” existente en “GraseHotSpot”, por lo tanto, el modulo “DnsMasq” reasigna nuevamente las direcciones “IP” dentro de la “LAN” del servidor. Debido a la estrecha relación existente entre los módulos que conforman “GraseHotSpot”, no es posible desactivar “DsnMasq” ya que “GraseHotSpot” debe capturar los clientes e impedirles el paso a “internet” hasta ser autenticados, es en este momento donde los clientes hacen parte de otra red y se les debe asignar una dirección “IP” dentro del servidor.

Seguidamente, se configuran los “AP” en modo “Bridge”, en este modo de funcionamiento los “AP” se encargan de hacer un puente inalámbrico entre los clientes y la “LAN “, por lo tanto, el modulo “DnsMasq” realiza las asignaciones de direcciones “IP” a los clientes de forma consecutiva empezando por la dirección 10.1.0.2 independientemente de que “AP” provenga el cliente. Por ende, no es posible identificar desde el servidor de forma dinámica de que “AP” proviene el

tráfico, debido a que se necesitan datos como la “MAC” o el “SSID”, y estos no pueden ser obtenidos ya que los “AP” pertenecen a otra subred. Para más detalles, ver las secciones 10.2.1 y 10.2.2.

Sin embargo, es posible individualizar el tráfico desde el servidor de forma estática, por lo que se plantea una prueba de funcionamiento en un ambiente controlado para 4 clientes, en el cual “DnsMasq” asigna direcciones “IP” consecutivas a los clientes desde la 10.1.0.2 hasta la 10.1.0.5, y por medio de las modificaciones del código fuente de “GraseHotSpot” y un módulo de administrador de contenidos del portal cautivo integrado al servidor, individualizar el tráfico “HTTP” y desplegar diferentes contenidos en función de la dirección “IP” del cliente.

10.2.3.1 Filtrado por dirección IP utilizando NAT. “NAT” (Network Address Translation) es un mecanismo que se utiliza para mapear un conjunto de direcciones “IP” a una sola dirección “IP”, multiplexandola a través de distintos puertos. En la actualidad, este tipo de “NAT” se utiliza para que los hogares u oficinas puedan tener acceso a internet utilizando una sola dirección “IP” pública, mientras que al interior de la red privada múltiples dispositivos puedan estar conectados y tener diferentes “IPs” privadas.

Se considera “NAT” como una alternativa que permitirá diferenciar desde que “AP” proviene el tráfico que llega al servidor, debido a que los “APs”, que en este caso deberán ser configurados como “routers”, proveerán el servicio de “DHCP” para los dispositivos de la red inalámbrica, dando a cada cliente una dirección “IP” entre un rango de direcciones que son previamente configuradas, e implementarán “NAT”, convirtiendo cada una de las direcciones asignadas a una sola disponible en el servidor central de “GraseHotSpot”, y diferenciando las conexiones gracias al puerto.

Es de esta forma, como todos los clientes que lleguen al servidor de “Grase” a través del mismo “AP”, ingresarán con la misma dirección “IP”, de manera que cada “AP” tendrá una dirección “IP” única ante el servidor. Por consiguiente, si se utiliza una combinación de filtrado por “IP”, y aplicación de “NAT” en los dispositivos de red, debe ser posible identificar el área inalámbrica desde la cual llega un cliente basándose en la dirección “IP”.

10.2.4 Otras alternativas. De manera adicional, se exploraron teóricamente otras alternativas mediante las cuales sería posible distinguir una zona inalámbrica de otra, sin embargo, nunca fueron consideradas para realizar pequeñas pruebas unitarias por diferentes razones, las cuales también son expuestas en esta sección.

10.2.4.1 RSSI. El RSSI (Received Signal Strength Indicator) es una métrica relativa que se utiliza en el estándar 802.11 para medir la fuerza (amplitud) de la señal [24].

Actualmente, con la popularidad de las tecnologías de la información y la comunicación, la demanda por información basada en localización es creciente, para muchos y variados usos, especialmente, en lugares públicos, que van desde centros comerciales, aeropuertos, hasta salas de exposición, oficinas, entre muchos otros. Existen muchas tecnologías que permiten el uso de técnicas de localización, como por ejemplo, infrarrojo, señal de radiofrecuencia, "Bluetooth", etc. Sin embargo, el auge de las redes inalámbricas y los dispositivos móviles ha llevado a un aumento dramático en la cantidad de "APs" que se pueden encontrar en sitios públicos, que además son "APs" de uso libre y gratuito, razón por la cual la tecnología que es actualmente más utilizada para localización es el "WI-FI" [25].

En el escenario planteado en este proyecto, se podría pensar en el "RSSI" como un indicador conveniente al momento de diferenciar un área inalámbrica de otra, al entender que si estamos hablando de una red pública inalámbrica que cubre un área física extensa, luego las diferencias entre las fuerzas de la señales recibidas implican necesariamente diferencias entre distancias, es decir, un "AP" que se encuentre cerca al punto de medición debe indicar un "RSSI" mucho más alto que un "AP" que se encuentre muy lejos de dicho punto, por lo tanto, es posible mapear y asociar niveles de "RSSI" a distancias cuando se conoce la topología de la red y las locaciones de los "APs".

Sin embargo, hay ciertas características inherentes al "RSSI" que deben ser tenidas en cuenta para su uso, y bajo las cuales este parámetro se convierte en uno difícil de utilizar para el propósito de esta investigación.

---

[24] COLEMAN, David. CWNA Certified Wireless Network Administrator Official Study Guide. Indianápolis: Wilkey, 2009. P. 89.

[25] YUAN FENG, Xiuyan. RSSI-based algorithm for indoor localization [en línea]. Scientific Research. Qingdao: Ocean University of China. 2013. 6 p. [Consultado el 5 de junio de 2017]. Disponible en: http://file.scirp.org/pdf/CN_2013071010352139.pdf

En primer lugar, los “RSSI” no son medidas absolutas, sino relativas. Esto quiere decir que aunque el “RSSI” que es medido de 0 a 255, puede ser expresado en medidas como el “dBm”, y esta medida no significa lo mismo para todos los fabricantes, ya que los valores del “RSSI” dependen directamente de la implementación y la escala o rango que el fabricante haya designado para el dispositivo inalámbrico. En otras palabras, un dispositivo de Cisco, por ejemplo puede lanzar un “RSSI” de -30 “dBm” y esto puede ser distinto que -30 “dBm” en un dispositivo de Aruba. En consecuencia, si se utilizará este parámetro para la identificación de áreas inalámbricas en este proyecto, habría que garantizar que las medidas son coherentes, ya sea utilizando todos los dispositivos de una misma marca y vendedor, o realizando un extenso estudio de los diferentes vendedores y su interpretación del “RSSI” para garantizar que se utilizarán valores convertidos a un mismo sistema o rango.

En segundo lugar, el RSSI es una métrica establecida dentro del estándar 802.11, lo cual quiere decir que se utiliza en redes inalámbricas, lo cual es, de cierto modo, inoportuno para este proyecto, puesto que pretende brindar una solución centralizada a partir de la cual todos los paquetes de la red y todas las funcionalidades del portal cautivo deberían poder ser administradas desde el servidor central, y si bien el servidor central posee una tarjeta de red inalámbrica, esta se utiliza únicamente para la entrada del internet, mientras que la interfaz que se usa para la transmisión de datos entre “APs” y servidor es “Ethernet”, es decir, no es una tarjeta de red inalámbrica y por lo tanto a través de esta no llegan paquetes que correspondan al estándar 802.11, en consiguiente, no sería posible obtener datos de “RSSI” correspondientes a los “APs” en el servidor central.

10.2.4.2 NAT dinámico y estático. La implementación de “NAT” que se expone en la sección 10.2.3.1 Filtrado por dirección IP utilizando NAT. “NAT” (Network Address Translation) es un mecanismo que se utiliza para mapear un conjunto de direcciones “IP” a una sola dirección “IP”, multiplexandola a través de distintos puertos. En la actualidad, este tipo de “NAT” se utiliza para que los hogares u oficinas puedan tener acceso a internet utilizando una sola dirección “IP” pública, mientras que al interior de la red privada múltiples dispositivos puedan estar conectados y tener diferentes “IPs” privadas. corresponde a la forma de “NAT” más utilizada, que se conoce como “sobrecarga” y utiliza “PAT” (Port Address Translation) para multiplexar una sola “IP” utilizando varios puertos. No obstante, existen otras implementaciones de “NAT” que serían en realidad, más factibles y viables para utilizar en este proyecto específicamente, como lo son el “NAT” dinámico y el “NAT” estático.

Estos dos tipos de “NAT” se caracterizan por el hecho de que guardan relación uno a uno y no multiplexan en ningún caso, simplemente mapean una “IP” dentro de un rango a otra “IP” dentro de un rango disponible. Se diferencian en que el “NAT” dinámico asigna de forma dinámica una dirección “IP” dentro de un rango a cualquier dirección “IP” dentro de otro rango especificado, mientras que el “NAT” estático espera una dirección “IP” específica para asignársela a otra dirección “IP” que ya ha sido especificada.
El “NAT” dinámico especialmente, sería perfecto para utilizar bajo la topología de red planteada para el portal cautivo, pues permitiría mapear las direcciones “IP” de tal manera que todos los clientes de un “AP” pudieran tener una dirección válida dentro del servidor, y a la vez, limitar dicha dirección a un rango definido por el “AP”, de tal manera que sería posible utilizar el filtrado por “IP” para distinguir inequívocamente al cliente, y al mismo tiempo, tener en cuenta el rango al que pertenece la dirección para distinguir inequívocamente el “AP”.

Para el desarrollo de este proyecto se hizo imposible implementar pruebas utilizando “NAT” dinámico puesto que no se disponía de dispositivos inalámbricos que tuvieran esta funcionalidad. Los “APs” utilizados en esto proyecto son marca D-link referencia DIR-600, y sólo poseen “NAT” por sobrecarga. Los “APs” más sencillos poseen únicamente “NAT” por sobrecarga puesto que es el más utilizado, teniendo en cuenta que el “NAT” se utiliza mayormente para la salida de distintos dispositivos a “Internet”, y que por lo general los usuarios sólo disponen de una “IP” pública. En un caso común, si un usuario implementará “NAT” dinámico, tendría que tener un número de “IPs” públicas equivalente al número de “IPs” privadas que espera tengan acceso a “Internet”, y este caso es menos visto.

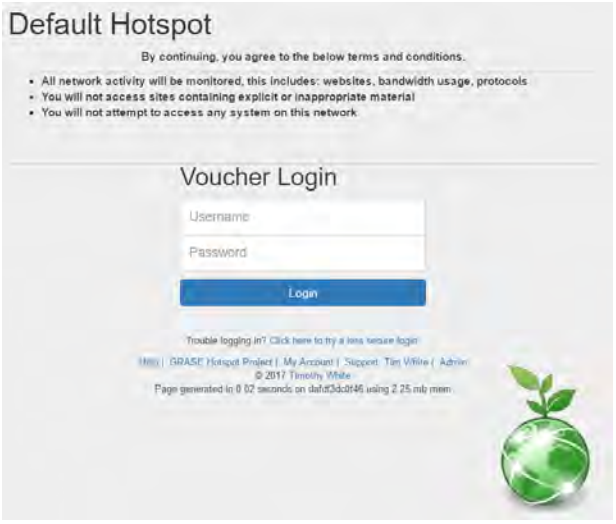
Es por todo lo anterior, que la alternativa de solución seleccionada para realizar el filtrado de tráfico de red y despliegue de contenidos publicitarios, es a través del filtrado por dirección “IP”, ya que este es un parámetro que puede ser obtenido y manipulado para realizar las validaciones respectivas mediante “PHP”. Debido a que se plantea el uso de filtro “IP” tanto para identificación de cliente como para reconocimiento de “AP” o área inalámbrica, se realizarán entonces dos pruebas para contrastar cada caso de uso. Para más detalles sobre las pruebas de funcionamientos, ver el capítulo 10. Pruebas de implementación de “GraseHotSpot” modificado sobre una red abierta con dos “APs”.

10.3 MODIFICACIÓN DEL CÓDIGO FUENTE

Al momento que “CoovaChilli” detecta que un cliente ha accedido a la red inalámbrica, despliega el portal cautivo para que se autentique y proceder a usar

el servicio de “Internet” tal como se muestra en la Figura 29. El archivo que ejecuta la lógica de esta aplicación web es “hotspot.php”.

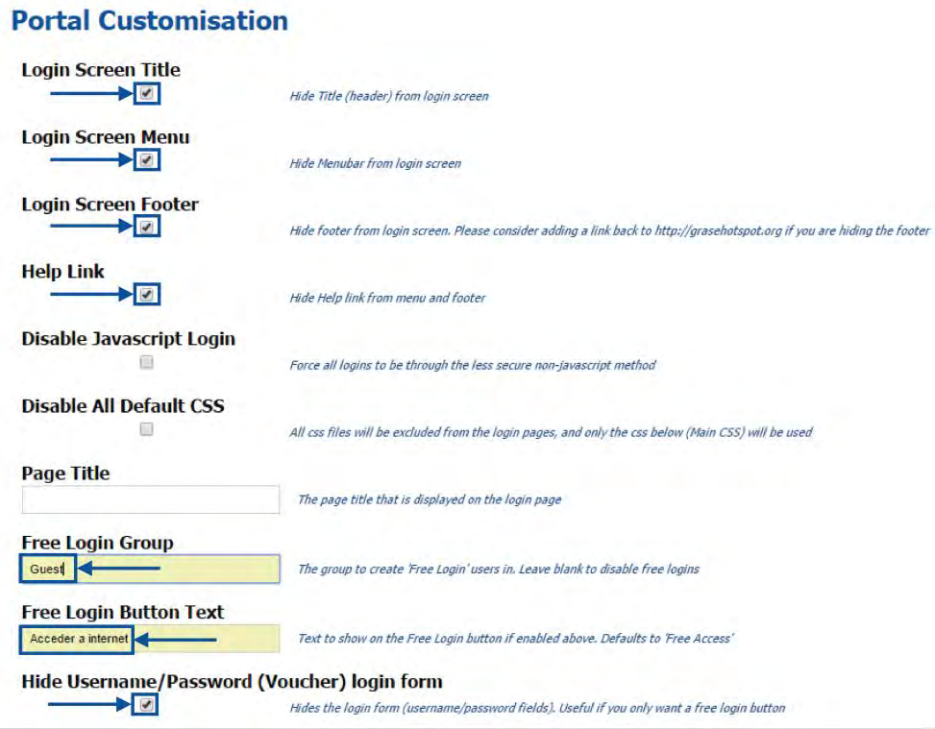
Figura 29. Portal cautivo por defecto de "GraseHotSpot"



Fuente: Elaboración propia.

Como primera medida, el portal cautivo únicamente debe desplegar un botón para acceder a “Internet” sin exigir el ingreso de credenciales. Por lo tanto, como se ilustra en la Figura 30, en el menú de administrador se configura el portal cautivo para modificar la vista y omitir el ingreso de credenciales. Se desactiva la visibilidad de estos componentes; título, logo, términos y condiciones y pie de página. Finalmente, en el portal de administrador se crea un grupo de usuarios al cual van a pertenecer los clientes que no ingresen credenciales, se elimina el formulario de “LogIn” remplazándolo por un botón con la etiqueta “Acceder a internet”.

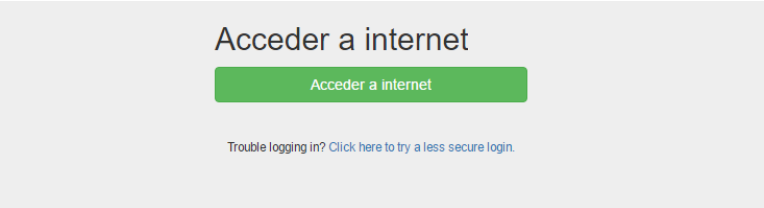
Figura 30. Configuración "free login" del portal cautivo desde el menú de administrador



Fuente: Elaboración propia.

Con estas modificaciones, “GraseHotSpot” permite desplegar un portal cautivo para ofrecer el servicio de internet con un “click” ”, tal como se muestra en la Figura 31. Sin embargo, como “CoovaChilli” debe comparar obligatoriamente credenciales con “FreeRadius” y posteriormente con “MySql”, “GraseHotSpot” realiza el proceso de “LogIn” de forma transparente para el usuario mediante un algoritmo que registra credenciales aleatorias y permite crear una sesión por cada cliente para el posterior uso de “Internet.

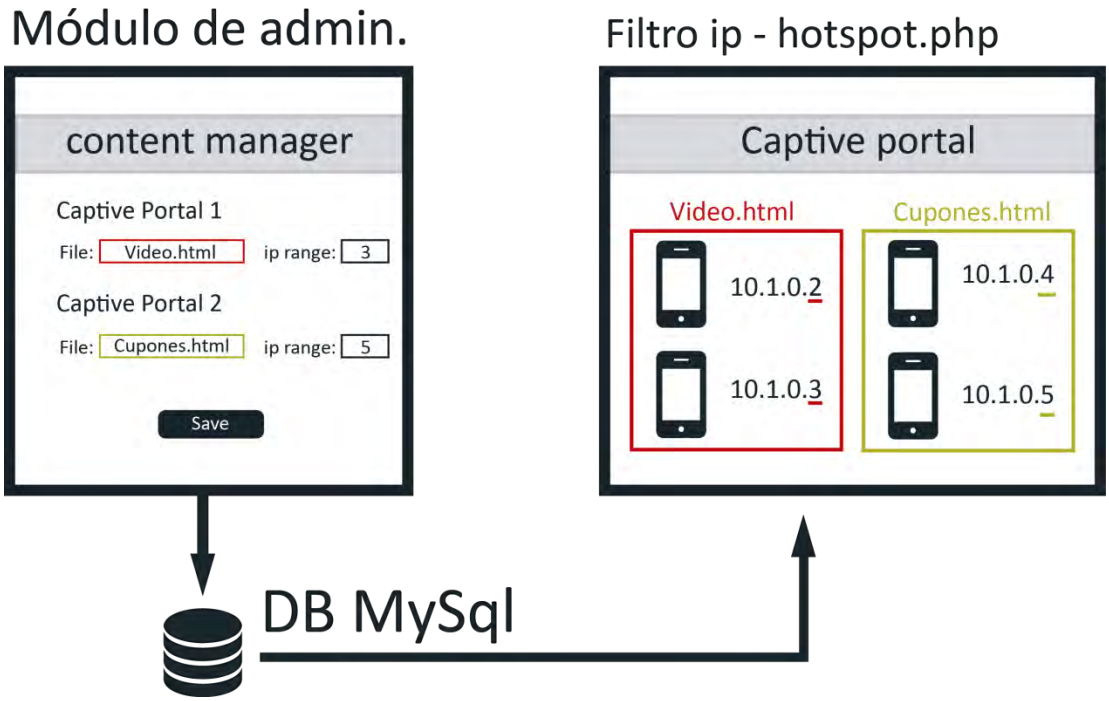
Figura 31. Portal cautivo sin ingreso de credenciales



Fuente: Elaboración propia.

A continuación, se desarrolla y se integra al servidor una aplicación web para administrar los contenidos que se van a desplegar en un determinado rango de direcciones “IP”, y se modifica el archivo “hotspot.php” para que consulte la información de la base de datos en donde se encuentra la configuración del contenido y rango de “IP” para cada “AP” establecidos previamente por el administrador, finalmente, con esta información desde el archivo “hotspot.php”, se obtiene la dirección “IP” del cliente para compararla contra los rangos de la base de datos y decidir que contenido desplegar en el portal cautivo. La interacción entre estos tres módulos se ilustra en la Figura 32. El código fuente de esta modificación se puede encontrar en el Anexo A.

Figura 32. Modificaciones realizadas a "GraseHotSpot" para agregar funcionalidad de filtrado y discriminación de contenidos



Fuente: Elaboración propia.

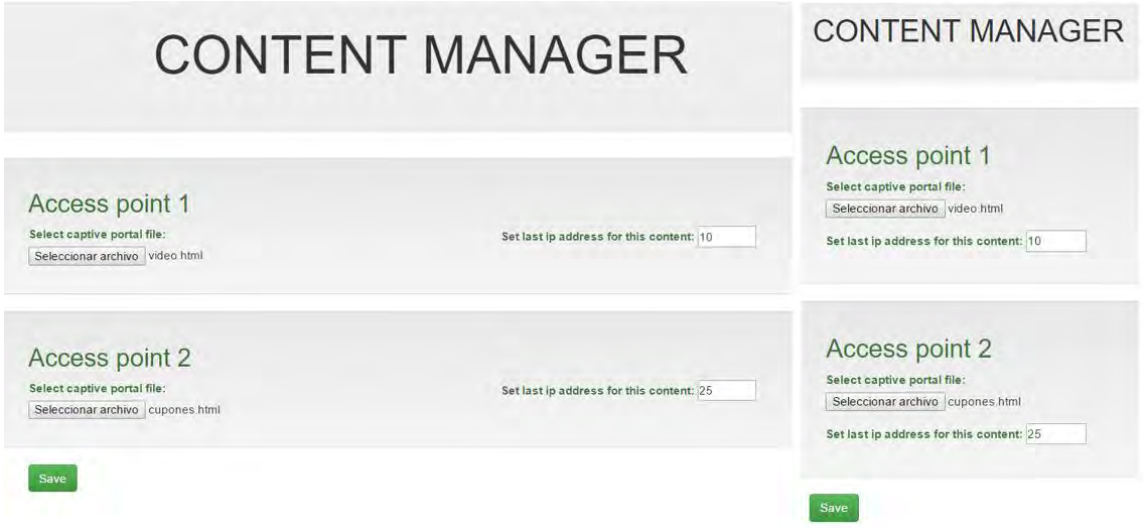
10.3.1 Módulo de administración de contenidos del portal cautivo. Este módulo permite al administrador de la red asignar los contenidos con su respectivo rango de direcciones “IP” que van a ser desplegados en cada portal cautivo.

Debido a que la información de configuración debe ser persistente, es decir, siempre debe estar disponible tanto para modulo del administrador como para el archivo “hotspot.php”, es necesario crear una base de datos en la cual se almacene dicha información.

Mediante “MySql”, se crea una base de datos llamada “content_manager” con la tabla “access_points”, la cual contiene campos para identificar el número del “AP”, es decir, “AP-1” o “AP-2”, el archivo en formato “.html” que va a ser desplegado y el rango de direcciones “IP” para dicho contenido.

Posteriormente, mediante el “framework” de desarrollo web responsivo “Bootstrap 3”, se crea la vista de la aplicación, la cual consta de un formulario en donde se asignan los valores de configuración del portal cautivo mencionados anteriormente, a la izquierda para computadores y a la derecha para dispositivos móviles tal como se muestra en la Figura 33.

Figura 33. Vista de la aplicación web del módulo de administración de contenidos



Fuente: Elaboración propia.

Al momento que el administrador de la red guarde los valores de configuración del portal cautivo, mediante el lenguaje de programación “php”, se obtienen del formulario y se insertan en la base de datos “content_manager”. El código fuente

del módulo de administrador tanto “front-end” como “back-end” pueden ser consultados en el Anexo B y Anexo C respectivamente.

Vale la pena aclarar que los rangos de direcciones “IP” para los contenidos, están definidos por el valor máximo del ultimo octeto, por ejemplo, si el administrador debe desplegar el archivo video.html en el rango de direcciones “IP” concebido entre la 10.1.0.2 hasta la 10.1.0.10, y el archivo cupones.html de la 10.1.0.11 hasta la 10.1.0.25, debe ingresar en el formulario el número 10 para el primer “AP” y el número 25 para el segundo “AP”. Estos valores serán consultados por el archivo “hotspot.php” para ser validados y desplegar el contenido correspondiente.

Finalmente, se configura el servidor web “Apache” para que aloje esta aplicación web en la dirección 10.1.0.1/grase/content-manager/index.php.

10.3.2 Modificación del archivo “hotspot.php”. Este archivo se encarga de ejecuta la lógica del portal cautivo y llamar los elementos que conforman la vista en formato “.html”.

Inmediatamente el portal cautivo es cargado, se conecta y consulta la información de configuración en la base de datos y las asigna a las variables globales “$file1”, “$range1”, “$file2”, “$range2”.

Posteriormente, haciendo uso del “framework” “jQuery”, el cual simplifica la sintaxis de “JavaScript” para interactuar con los elementos de la vista del portal cautivo, se crea una función que se ejecuta en el “head” del documento “html” incrustado en “hotspot.php” para referenciar en la vista las direcciones de los archivos “.html” del contenido en las variables “#App_1” y “#App_2”.

En el “body” del documento “html”, se incrusta código “php”, el cual ejecuta la función “is_app_1()” que a su vez ejecuta la función “get_client_ip()” que retorna la dirección “IP” del cliente. En el interior de “is_app_1()” se extrae el ultimo octeto de la dirección “IP” con la función “explode(string $delimiter , string $string)”[26] y se almacena en la variable local “$last_ip_number” con la que se procede a realizar una validación con los rangos de direcciones “IP” almacenados en las variables globales “$range1” y “$range2”, de esta forma se determina si con la dirección “IP” del cliente se debe desplegar el contenido almacenado en “$file1” o “$file2”,

[26] PHP: Explode – manual [en línea]. The PHP Group, 2001. [Consultado 25 de febrero de 2017]. Disponible en internet: http://php.net/manual/es/function.explode.php

por lo tanto, la función “is_app_1()” retorna “true” para desplegar el contenido 1 o “false” para el contenido 2.

Finalmente, se manipula el valor retornado por la función “is_app_1()”, por lo tanto, si es verdadero, se ejecuta el comando “echo ‘ ’”, y si es falso, “echo ‘ ’”, de esta forma, se agrega dinámicamente un nuevo “div” a la vista del portal cautivo que despliega el contenido referenciado en la variable “App_1” o “App_2” extraída de la configuración del administrador, por lo tanto, el portal cautivo filtra y muestra diferentes contenidos dependiendo de la dirección “IP” del cliente. Ver la Figura 32.

11. PRUEBA DE IMPLEMENTACIÓN DE "GRASEHOTSPOT" MODIFICADO SOBRE UNA RED ABIERTA CON DOS "ACCESS POINTS"

Una vez modificado el código fuente de "GraseHotSpot" para filtrar e individualizar el tráfico de la red y agregar el módulo de administración de contenidos, se plantean dos pruebas de implementación sobre una red abierta con dos "AP" de forma controlada para en entorno ficticio.

11.1 PRUEBA DE IMPLEMENTACIÓN DE "GRASEHOTSPOT" CON FILTRO "IP" PARA IDENTIFICAR CLIENTES

La primera prueba de implementación consiste en levantar todos los servicios implicados en el servidor "GraseHotSpot", seguidamente, acceder al portal de administración de contenidos para configurar el portal cautivo de la siguiente forma:

- Desplegar en el portal cautivo del "AP-1", el archivo llamado video.html para las direcciones "IP" 10.1.0.2 y 10.1.0.3.

- Desplegar en el portal cautivo del "AP-2", el archivo llamado cupones.html para las direcciones "IP" 10.1.0.4 y 10.1.0.5.

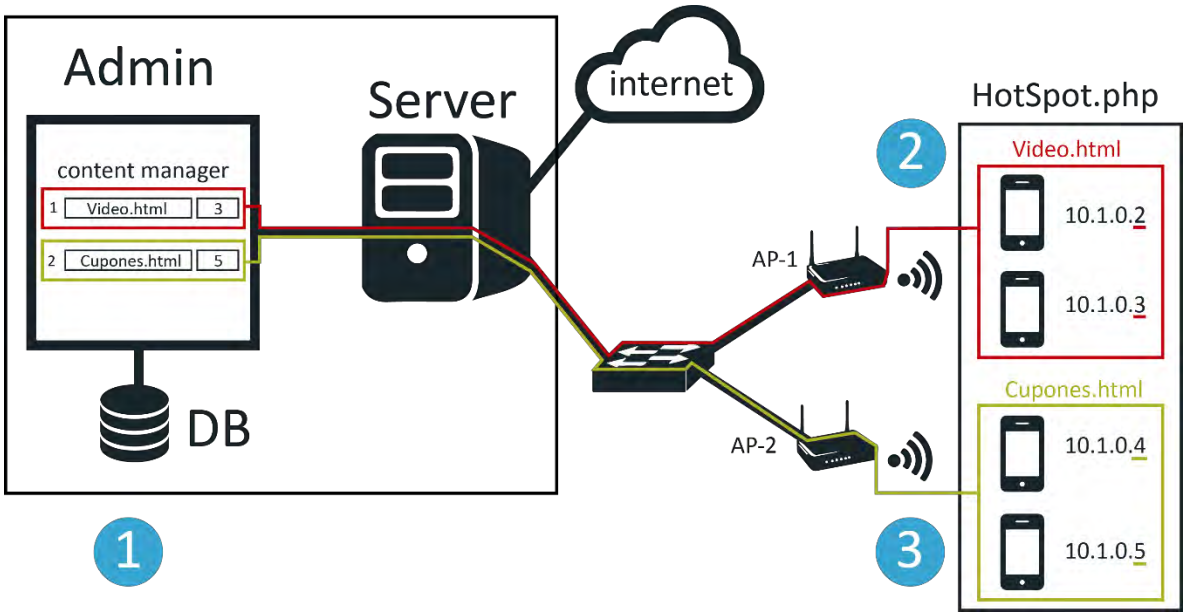
Como se evidencia, la prueba de implementación se plantea únicamente para 4 clientes, los cuales deben acceder a la red inalámbrica de forma controlada, debido a que el filtrado se realiza a través de las direcciones "IP" consecutivas que asigna el servidor de "DHCP" "DnsMasq".

Terminada esta configuración, "GraseHotSpot" se encuentra listo para recibir clientes y desplegar los diferentes portales cautivos dependiendo de la dirección "IP".

Los dos primeros clientes deben acceder a la red a través del "AP-1" a los cuales se les asignan las direcciones "IP" 10.1.0.2 y 10.1.0.3, por lo tanto, el archivo "hotspot.php" despliega el portal cautivo con el contenido configurado para ese rango de direcciones "IP", es decir, video.html. Finalmente, los 2 clientes restantes, deben acceder a la red a través del "AP-2" a los cuales se les asignan

las direcciones “IP” 10.1.0.4 y 10.1.0.5, por ende, el archivo “hotspot.php” despliega el portal cautivo con el contenido de cupones.html. Esta prueba de implementación se resume en la Figura 34.

Figura 34. Prueba de implementación de "GraseHotSpot" modificado sobre una red abierta con dos "Access Points"



Fuente: Elaboración propia.

Cabe mencionar que el administrador de la red está en la libertad de controlar los contenidos desplegados para un rango de direcciones “IP” específicos, por lo tanto, si se desea cambiar el contenido de alguno de los dos portales cautivos, simplemente se asigna el nuevo el archivo en formato “.html”.

11.2 PRUEBA DE IMPLEMENTACIÓN DE “GRASEHOTSPOT” CON FILTRO “IP” PARA ACCESS POINTS Y CONFIGURACIÓN “NAT”

La segunda prueba de implementación es similar a la primera, en el sentido en que utiliza la misma modificación basada en la implementación de un filtro “IP”, pero se distingue en la configuración adicional que hay que realizar sobre los “Access Points” para ponerlos en modo “router” y permitirles utilizar “NAT”, así como el servidor “DHCP” interno.

En primer lugar, se debe desactivar el modo “bridge” del “AP” para permitirle utilizar el “NAT”. Posteriormente hay que configurar el “DHCP” que servirá a los clientes inalámbricos, definiendo una “IP” para el dispositivo, junto con la respectiva máscara de red y el rango de direcciones que el “DHCP” tendrá permitido asignar. De esta forma se le asigna al “AP-1” la dirección 192.168.3.1 con máscara 255.255.255.0 y el rango de direcciones de .100 a .199. Para el “AP-2” se utiliza exactamente la misma máscara y el mismo rango, sin embargo, esta vez la dirección será 192.168.4.1.

Una vez lista la configuración de los “APs”, se procede a iniciar el servidor de “GraseHotSpot” y se edita la distribución de los contenidos utilizando el módulo de “Content Manager”, como se indica a continuación:

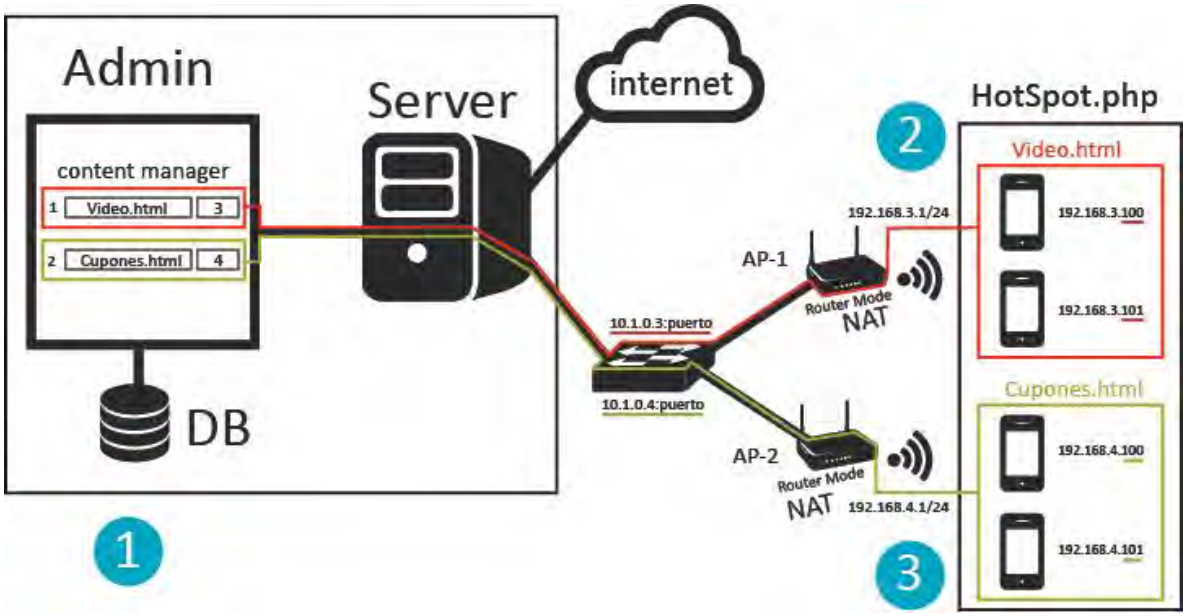
- Desplegar en el portal cautivo del “AP-1”, el archivo llamado video.html para la dirección “IP” 10.1.0.3

- Desplegar en el portal cautivo del “AP-2”, el archivo llamado cupones.html para la dirección “IP” 10.1.0.4

Ya que sólo hay dos “APs”, sólo será necesaria la configuración de sus dos direcciones, sin embargo, para observar el comportamiento de los clientes bajo este modelo, durante la prueba, 4 clientes se conectarán a los “APs”, dos en cada uno, y se espera que los clientes obtengan la “IP” dentro de la red correspondiente a su “AccesPoint”, a la vez que cuando se les exige pasar por el portal cautivo, puedan ver el contenido perteneciente a su área inalámbrica.

De esta forma, los clientes conectados al “AP-1” deberían poseer las direcciones 192.168.3.100 y 192.168.3.101, mientras que cuando acceden al navegador de Internet pueden visualizar el contenido presente en “video.html”. Paralelamente, los clientes del “AP-2” obtendrán las direcciones 192.168.4.100 y 192.168.4.101, y se les presentará el contenido de “cupones.html”. Esta prueba se ilustra en la Figura 35.

Figura 35. Prueba de implementación de "GraseHotSpot" modificado sobre una red abierta con dos "Access Points" configurados como "routers"



Fuente: Elaboración propia.

11.3 DESARROLLO DE CONTENIDOS PARA EL PORTAL CAUTIVO

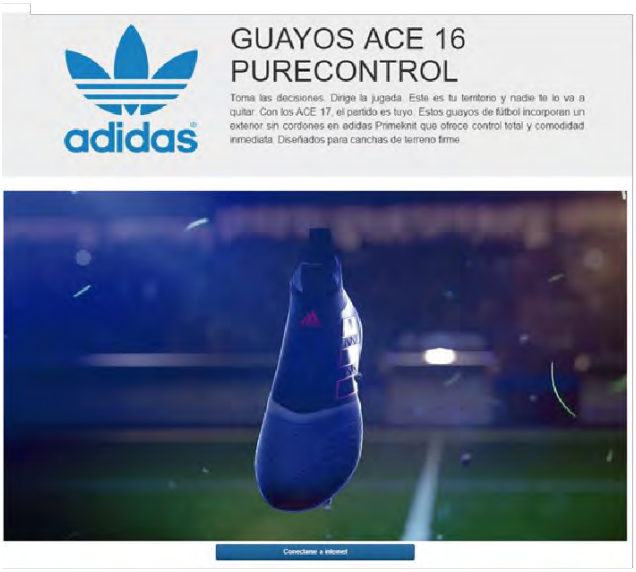
Para realizar las pruebas de funcionamiento, se desarrollaron dos aplicaciones web responsivas, con el fin de ser desplegadas correctamente en todos los periféricos que accedan a estas.

El desarrollo de estas aplicaciones web fue pensado para un entorno enteramente publicitario, en donde se despliegan diferentes contenidos como por ejemplo marcas, productos, servicios y descuentos en un centro comercial. Una vez el contenido es desplegado, el usuario simplemente debe interactuar con un botón para acceder normalmente al servicio de internet.

La capa de aplicación del contenido de estos portales cautivos, fue realizada con tecnología web. Se empleó "HTML5" y "CSS3" haciendo uso del "framework" "Bootstrap 3", adicionalmente, para las interacciones sobre los elementos desplegados, se usó el lenguaje de programación "JavaScript", finalmente, las animaciones y los efectos dinámicos se realizaron gracias a la librería de animaciones web llamada "Animate.css

11.3.1 Portal cautivo con video publicitario. La función básica de esta aplicación web de portal cautivo, es reproducir un video alusivo a una marca, producto, servicio o descuento de forma automática al momento de ser cargada, además, en su interfaz gráfica se ubica un botón en la parte inferior con la etiqueta “Conectarse a internet”, el cual debe ser presionado para acceder al servicio tal como se ilustra en la Figura 36. En el Anexo D se encuentra el código fuente de este contenido publicitario.

Figura 36. Portal cautivo con video publicitario de guayos ACE 16 de "Adidas"



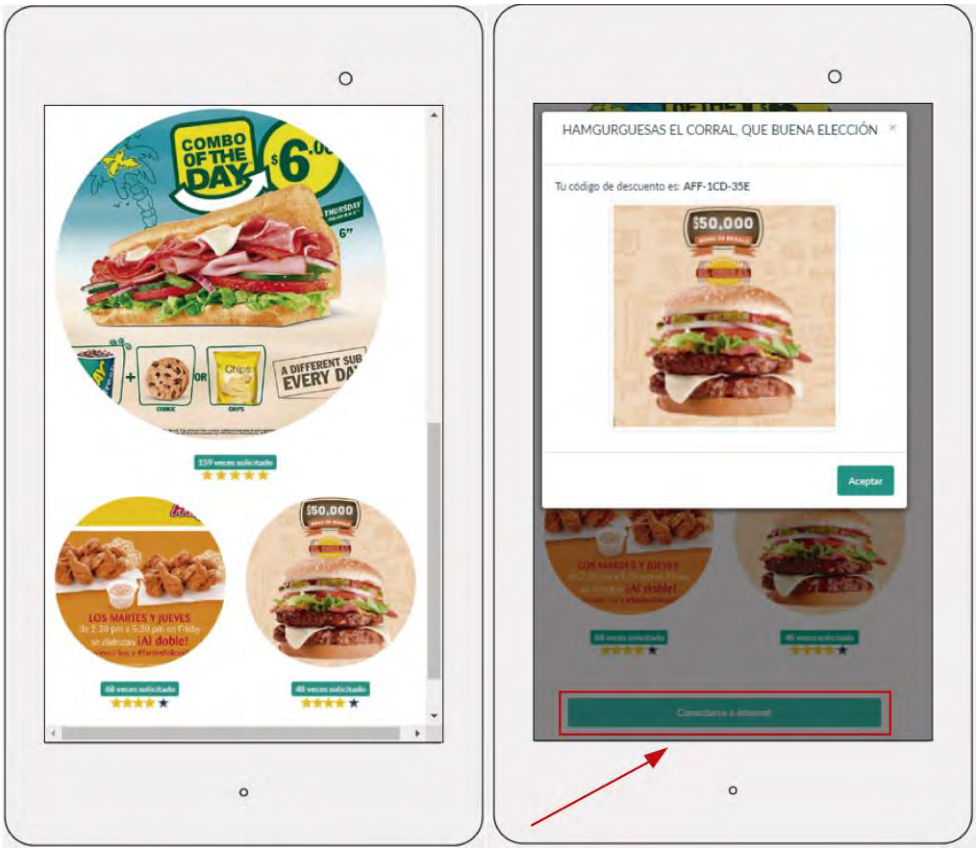
Fuente: Elaboración propia.

11.3.2 Portal cautivo de cupones para restaurantes. Este portal cautivo se diseñó para ser desplegado en zonas de comidas, en el cual, se le informa al cliente mediante una serie de imágenes alusivas a los productos de cada restaurante, las diferentes opciones y promociones que existen en ese espacio y momento del día.

Cada una de las imágenes mencionadas anteriormente, al ser presionadas abren un modal o ventana emergente con un pequeño mensaje del producto y su respectivo código de descuento. Para redimir este cupón, se plantea a futuro que el usuario deba presentar el código de descuento en la caja de dicho restaurante, con el fin de llevar un control de las ventas realizadas a través de este medio publicitario interactivo.

Al igual que el portal cautivo anterior, se sitúa un botón en la parte inferior de la interfaz gráfica, que debe ser presionado para acceder normalmente a “internet” tal como se ilustra en la Figura 37. Vale la pena aclarar, que ver el cupón no implica usarlo, esta decisión es finalmente tomada por el usuario. En el Anexo E se encuentra el código fuente de este contenido publicitario.

Figura 37. Portal cautivo de cupones para restaurantes



Fuente: Elaboración propia.

11.4 RESULTADOS

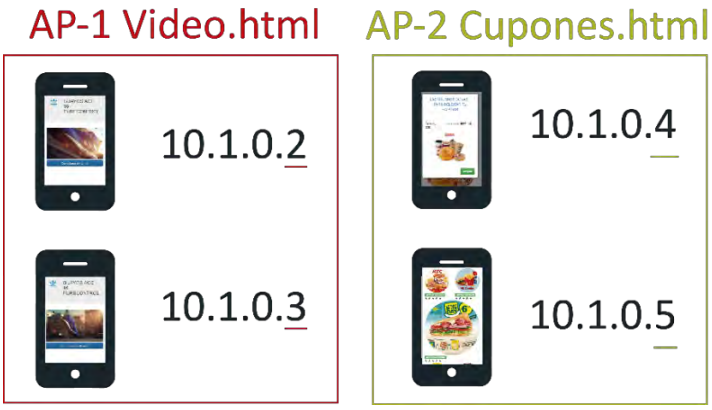
A continuación, se consignan los resultados esperados y obtenidos tras la implementación de las dos pruebas planteadas.

1. **11.4.1 Resultado del filtrado con "IP" para identificar clientes.** Como se puede apreciar en la

Figura *38*, se esperó que para los clientes conectados al "AP-1" las direcciones "IP" hubiesen sido 10.1.0.2 y 10.1.0.3 para consumir el contenido del portal cautivo video.html, y para los clientes conectados al "AP-2" las direcciones "IP" 10.1.0.4 y 10.1.0.5 para desplegar el contenido de los cupones para restaurantes. Además, vale la pena aclarar que el "SSID" de ambas redes "LAN" es irrelevante, este podría hacer referencia a la ubicación física de cobertura de cada "AP", debido a que en el proceso de filtrado de tráfico, únicamente se está usando el valor del ultimo octeto de la dirección "IP" del cliente.

Finalmente, el proceso de conexión, autenticación y despliegue de contenido es totalmente transparente e indetectable para el cliente, debido a que al tratarse "LAN" abiertas, no es necesario realizar ninguna configuración previa del dispositivo que se conecta al "AP" y una vez el cliente se encuentra conectado y cambie de ubicación hacia otra zona de cobertura, el dispositivo móvil recordará dicha "LAN" para conectarse automáticamente.

Figura 38. Resultados esperados de la prueba de implementación mediante "IP"



Fuente: Elaboración propia.

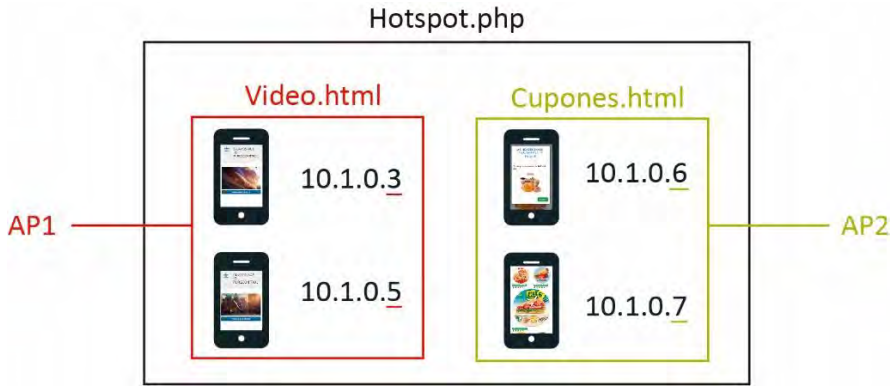
Sin embargo, "DsnMasq" asigna las direcciones "IP" de la siguiente forma; 10.1.0.2 para la interfaz del servidor, 10.1.0.3 para el primer cliente, 10.1.0.4 para la interfaz "WAN" del "router", y de la 10.1.0.5 en adelante para los siguientes clientes.

Vale la pena aclarar que el “switch” usado en la prueba no opera únicamente en la capa de enlace debido a que es un “router” configurado para realizar esta función, por lo tanto, tiene funcionalidades de capa de red del modelo “OSI” requiriendo la asignación de una dirección “IP”.

Por lo tanto, los dos primeros clientes tienen las direcciones “IP” 10.1.0.3 y 10.1.0.5 respectivamente, y los dos clientes restantes la 10.1.0.6 y 10.1.0.7. Por ende, que se modifican los rangos de direcciones “IP” a través del módulo desarrollado “Content Manager”, para desplegar el video publicitario en el portal cautivo del “AP-1” hasta la dirección “IP” 10.1.0.5, y desplegar los cupones para restaurantes en el portal cautivo del “AP-2” hasta la dirección “IP” 10.1.0.7.

Por último, el resultado de esta prueba fue el esperado, como se evidencia en la Figura 39, los primeros dos clientes que se conectaron al “AP-1” obtuvieron las direcciones “IP” 10.1.0.3 y 10.1.0.5, por lo tanto, se les desplego el contenido de portal cautivo configurado por el administrador, es decir, video.html. Finalmente, los dos clientes restantes que se conectaron al “AP-2”, se les asignó las direcciones “IP” 10.1.0.6 y 10.1.0.7, por esta razón, el portal cautivo fue desplegado con el contenido del archivo cupones.html.

Figura 39. Resultados obtenidos de la prueba de implementación mediante “IP”



Fuente: Elaboración propia.

11.4.2 Resultado del filtrado con "NAT" para identificar redes. Como se ilustra en la Figura 40, se esperó que los "APs" asignaran a sus respectivos clientes las direcciones "IP" definidas previamente en la configuración del "DHCP", posteriormente, realizar el cambio de las direcciones "IP" mediante "NAT" según el "AP" al cual el cliente esté conectado, seguidamente, se identifican las direcciones "IP" de los clientes en la sub red del servidor ya que siempre serán la 10.1.0.3 o 10.1.0.4 para el "AP-1" y el "AP-2" respectivamente, de este modo, es posible identificar desde que área inalámbrica el cliente está conectado para desplegar el contenido perteneciente a dicha área previamente asignado en el módulo de administrador de contenidos.

Figura 40. Resultados esperados de la prueba de implementación mediante "NAT"

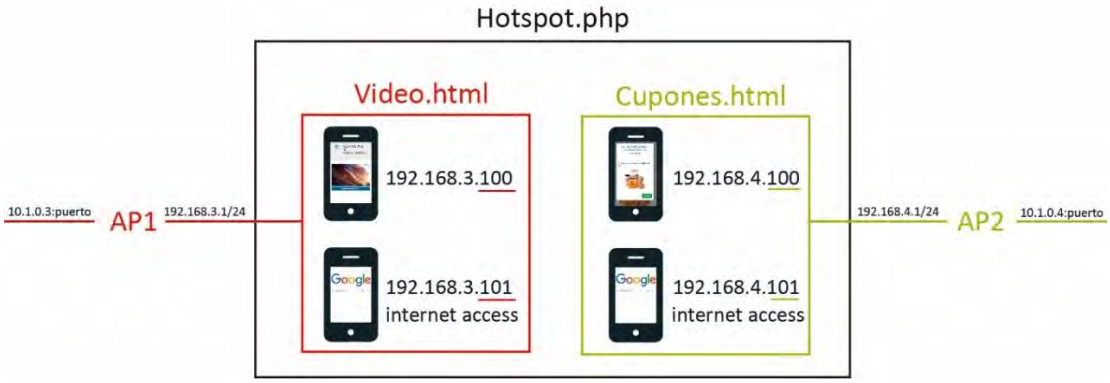
Fuente: Elaboración propia.

Luego de realizar la prueba de implementación, los resultados fueron los esperados en cuanto a la asignación y traducción de direcciones "IP" por parte de los "AP", sin embargo, como se muestra en la Figura 41, gracias a la implementación de "NAT", "GraseHotSpot" siempre verá a los clientes con las direcciones 10.1.0.3 y 10.1.0.4 según el "AP" al cual estén conectados, por lo tanto, no es posible desde el servidor identificar individualmente cada cliente a través de la dirección "IP" a pesar de conocer de cual sub red proviene la petición, causando que el primer cliente conectado a cada "AP" deba autenticarse, es decir

ver contenido publicitario, y los siguientes tengan acceso a internet omitiendo este paso.

Por ejemplo, si el primer cliente se conecta a la red inalámbrica del “AP-1”, como consecuencia del “DHCP”, se le asigna la dirección “IP” 192.168.3.100, posteriormente, a causa del mecanismo “NAT” al pasar por el “AP-1” “DnsMasq” asigna la dirección “IP” 10.1.0.3 al “AP-1” y la dirección de dicho cliente para “GraseHotSpot” se convierte en la asignada por “DnsMasq”, es decir, 10.1.0.3, por lo tanto, al cliente se le despliega el contenido publicitario del área inalámbrica del “AP-1” hasta que se autentique, seguidamente, si un nuevo cliente se conecta, se le asigna la dirección “IP” 192.168.3.101 al interior del área inalámbrica, pero al pasar por el “AP-1” hacia el servidor, “NAT” cambia dicha dirección por la 10.1.0.3 la cual ya está autenticada por el cliente anterior permitiéndole acceso a internet sin restricción alguna.

Figura 41. Resultados obtenidos de la prueba de implementación mediante “NAT”



Fuente: Elaboración propia.

Una vez realizadas las pruebas, se procede a elaborar una comparación de las características a favor y en contra de cada método de filtrado de tráfico de red.

En cuanto al filtrado por “IP”, es posible identificar cada cliente de forma individualizada, por consiguiente, para poder consumir normalmente el servicio de internet, deben ver contenidos publicitarios, además, las funcionalidades de gestión de clientes propias de “GraseHotSpot” como la asignación de cuotas, monitoreo de tiempo y administración de recursos de red, pueden ser implementadas de forma particular para cada usuario. No obstante, se imposibilita identificar el área inalámbrica desde la cual el cliente esté conectado.

En cuanto al filtrado por “IP” implementando “NAT” en cada “AP”, es posible identificar inequívocamente el área inalámbrica desde la cual el cliente está conectado, sin embargo, “GraseHotSpot” únicamente conoce una dirección “IP” por cada red inalámbrica, dicho de otro modo, el comportamiento de cada “AP” se asemeja a un servidor “proxy” el cual hace las peticiones de los clientes a nombre suyo, de tal forma que, los contenidos publicitarios solo serán desplegados al primer cliente de cada red inalámbrica, puesto que la dirección “IP” de los siguientes clientes será la misma, la cual ha sido previamente autenticada.

Finalmente, se reconoce la importancia de refinar el proceso de filtrado por “IP” implementando “NAT” en cada “AP”, ya que el uso de este método da como resultado un “software” cercano a un servicio capaz de ser implementado en un entorno real, es por esto que se consigna siguiente apartado la implementación de un filtro que conste de dirección “IP” y puerto, con el fin de reconocer infaliblemente desde el servidor, cada uno de los clientes con la respectiva red inalámbrica desde la cual se realicen las peticiones.

12. TRABAJOS FUTUROS

El resultado de este proyecto resuelve el problema planteado inicialmente, sin embargo, dicha solución no es aplicable en un contexto real debido a los múltiples inconvenientes hallados durante el desarrollo, impidiendo que el sistema se adapte a un flujo dinámico de clientes que se conectan y desconectan de la red inalámbrica.

Es por esta razón, que se plantean dos posibles alternativas de solución al problema, que podrían ser implementadas en un entorno real para ser llevadas a fase de producción, es decir, que sean soluciones estables y consistentes que identifiquen y filtren el tráfico “http”, y que además, se adapten al flujo dinámico de clientes existentes en una red inalámbrica pública abierta.

12.1 FILTRADO DE TRÁFICO DE RED A TRAVÉS DEL PROTOCOLO “SNMP”

Esta alternativa consiste en hacer uso de la información almacenada en el “MIB” alojado en el “switch”, del cual se pueden extraer los datos del tráfico de red de cada puerto físico del “switch”.

El “MIB” o “Management Information Base”, es una base de datos que contiene información sobre los dispositivos “hardware” que están conectados sobre cada una de las interfaces de un dispositivo de red, además, es actualizada en tiempo real [27].

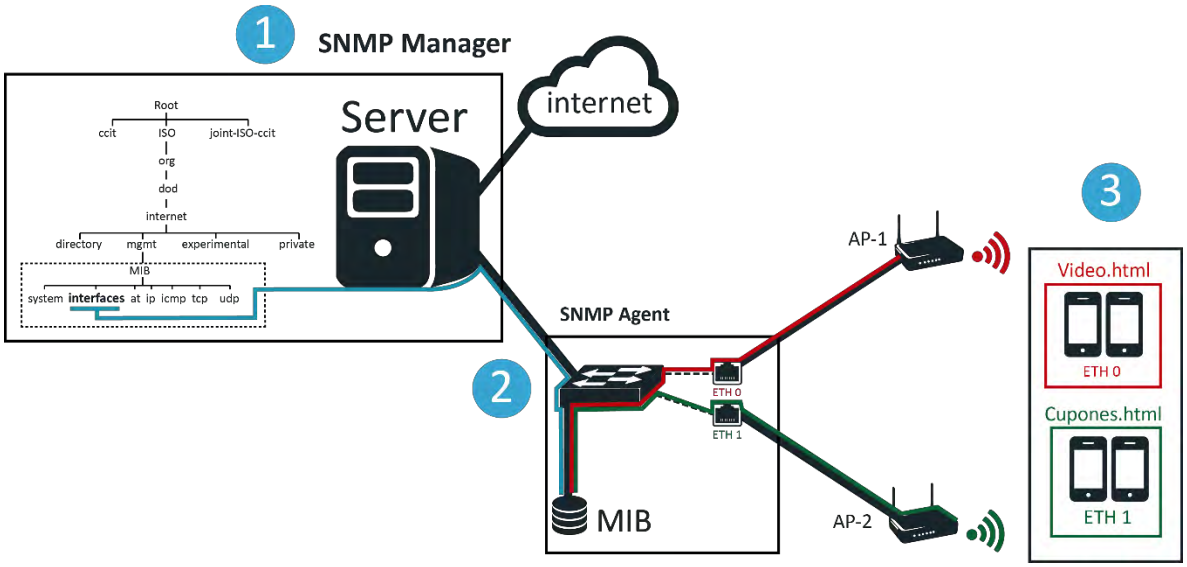
Esta base de datos está organizada de forma jerárquica o en forma de árbol, en donde existen nodos estructurales o “ramas”, los cuales únicamente almacenan su posición en el árbol, sin embargo, dicha información es usada para llegar a otros nodos de capas inferiores. Finalmente, se encuentran las hojas o nodos de información, los cuales almacenan los datos de las interfaces, estas hojas se encuentran referenciadas por un “OID” o “Object Identifier”, el cual es una cadena de índices concatenados que indican la ubicación del nodo desde “root” [28].

[27] HUCABY, David. CCNP Routing and Switching SWITCH 300-115. Indianapolis: Cisco Press, 2014. Pag. 324.
[28] Ibid. P. 324.

Debido a que la configuración y administración del portal cautivo se realizará de forma centralizada, es decir desde el servidor, se debe usar el protocolo “SNMP” o “Simple Network Management Protocol”, el cual permite consultar el “MIB” del “switch” de forma remota. Este proceso consta de dos participantes, el “SNMP manager”, el cual consume la información situada en el “MIB” de uno o más dispositivos de red, en este caso es el servidor que aloja la instancia “GraseHotSpot”, y el “SNMP agent”, que es el dispositivo de red que está siendo monitoreado, en este caso, el “switch” [29].

Mediante la información de la rama “interfaces” al interior del “MIB” y el protocolo “SNMP”, es posible identificar desde el “SNMP manager” a que puerto del “switch” o “SNMP agent” está conectado cada “Access Point”, y de esta forma filtrar el tráfico de red para desplegar diferentes contenidos dependiendo del “AP” de donde provengan las peticiones “http”, tal como se ilustra en la Figura 42.

Figura 42. Filtrado de red a través del protocolo SNMP



Fuente: Elaboración propia.

Sin embargo, el principal problema de esta solución, es la manipulación de los datos, si bien es posible obtener el puerto físico del “Switch” del cual proviene el tráfico, es sumamente complejo extraerlos para posteriormente ser usados en la capa de aplicación con el lenguaje programación “PHP” usado en el servidor, para finalmente validar y desplegar el contenido para cada “AP”. Adicionalmente, no se

[29] Ibid. P. 325

cuenta con la disponibilidad de tiempo y recursos suficiente para la investigación y desarrollo de esta posible solución.

Vale la pena mencionar que para implementar esta solución, el "switch" obligatoriamente debe ser compatible con el protocolo SNMP, además, dicha versión del protocolo debe ser la misma tanto en el "SNMP agent" como en el "SNMP manager" y en el módulo de "software desarrollado", de lo contrario, no es posible monitorear y extraer remotamente la información necesaria para realizar el filtrado de trafico de red.

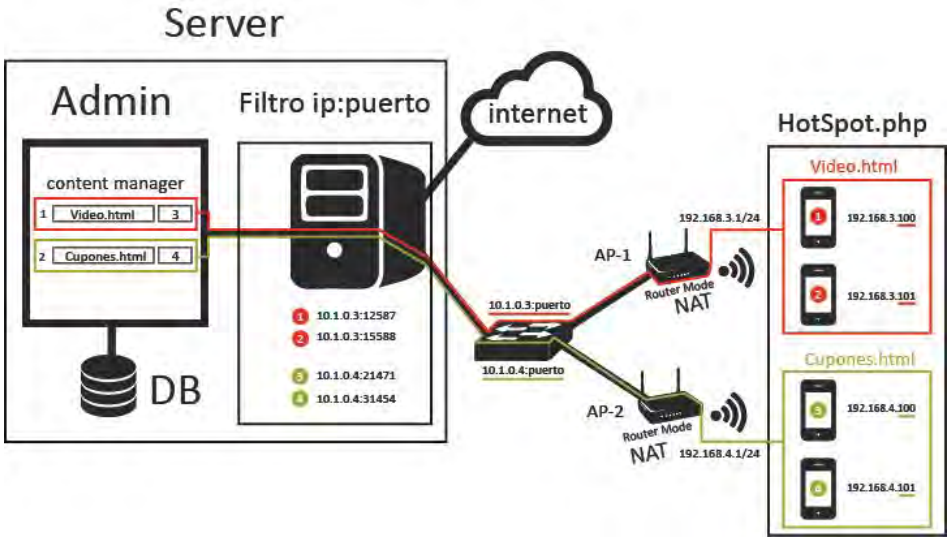
12.2 FILTRADO DE TRÁFICO DE RED POR "IP" Y PUERTO UTILIZANDO "NAT"

Esta alternativa consiste en realizar una mejora a la solución planteada en este proyecto. Al igual que la anterior, cada "AP" se configura para implementar "NAT" y proveedor direcciones "IP" a los clientes mediante "DHCP" en un segmento de red específico. Como se ilustra en la Figura 43, gracias al uso del mecanismo "NAT", se realiza el cambio de las direcciones "IP" de los clientes según el "AP" al cual se conecten, seguidamente, se identifican las direcciones "IP" de los clientes en la sub red del servidor, 10.1.0.3 o 10.1.0.4 para el "AP-1" y el "AP-2" respectivamente, y a diferencia de la solución desarrollada en este proyecto, se extrae adicionalmente el puerto de cada dirección "IP", por lo tanto, se tiene el par "IP":"Port" con el cual es posible identificar y autenticar en el servidor a cada uno de los clientes de manera inequívoca y obtener la red inalámbrica desde la cual están conectados, permitiendo desplegar correctamente el contenido publicitario perteneciente a dicha área para cada uno de los usuarios.

Implementar "NAT" en los "APs", trae como consecuencia que cada dirección "IP" traducida, este acompañada con su respectivo puerto con el fin de evitar ambigüedades [30]. Actualmente el módulo de portal cautivo "CoovaChilli" implementado al interior de "GraseHotSpot", usa únicamente la dirección "IP" de los clientes para autenticar y permitir el paso a "Internet", es por esto que se debe modificar el código fuente de dicho módulo para que adicionalmente valide la autenticación con el puerto, y de este modo evitar cualquier tipo de ambigüedad.

[30] HUCABY, David. Cisco Field Manual. Indianapolis: Cisco Press, 2002. p. 238

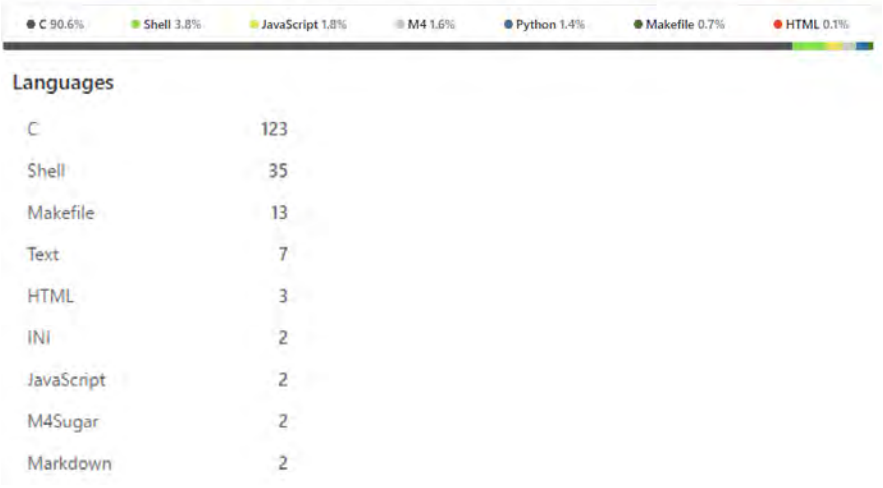
Figura 43. Filtrado de red por “IP” y puerto utilizando “NAT”



Fuente: Elaboración propia.

Finalmente, es importante mencionar que el cambio sustancial para el desarrollo de esta alternativa requiere un alto grado de conocimiento especialmente en los lenguajes de programación “C” y “Shell”, ya que como se muestra en la Figura 44, el 90.6% de los archivos de “CoovaChilli” están escritos en “C”, es decir 123 archivos, y un 3.8% está escrito en “Shell”, es decir 35 archivos.

Figura 44. Lenguajes de programación usados en "CoovaChilli"



Fuente: GitHub. [Consultado 22 de mayo de 2017]. Disponible en Internet: https://github.com/coova/coova-chilli

13. CONCLUSIONES

En cuanto al área de programación, es necesario discriminar la capa de la lógica con la de presentación, puesto que cada una de estas representó un reto diferente en cuanto al desarrollo y modificación de “GraseHotSpot”.

Para la capa de presentación o “front-end”, fue necesario explorar diferentes herramientas que facilitaran el diseño y desarrollo de aplicaciones web. Esto permitió hallar “frameworks” como “Bootstrap” y “jQuery”, los cuales han sido de suma importancia, ya que permiten implementar componentes web estéticos y funcionales de forma ágil, además, gracias a su filosofía de desarrollo “web responsive”, fue posible tener en cuenta la gran variedad de dispositivos que pueden acceder a dichas aplicaciones web para desplegar adaptativamente el contenido de acuerdo al tamaño del navegador.

Adicionalmente, para el diseño del formulario desplegado en la aplicación “web” “Content Manager”, se planteó cargar los valores de configuración actuales almacenados en la base de datos, es decir, el archivo “HTML” y los rangos para cada uno de los dos portales cautivos, con el fin de retroalimentar al administrador los valores existentes, y no simplemente abrir el formulario en blanco cada vez que se deban realizar modificaciones, sin embargo, esto únicamente fue posible para las entradas de los rangos, debido a que por aspectos de seguridad ningún navegador “web” permite asignar valores por defecto a entradas de tipo “file”.

En la capa de lógica o “back-end”, se encontraron diversos retos al momento de modificar el código fuente e integrar el modulo desarrollado, debido a que “GraseHotSpot” cuenta con una enorme cantidad de “scripts” y archivos de configuración relacionados entre sí, y que además carecen de documentación (aun siendo esta la alternativa con las características más favorables para el desarrollo del proyecto), por estas razones, fue complejo encontrar los archivos y parámetros que debían ser modificados. Además, la solución desarrollada requirió la implementación de una base de datos, la cual provee de información a los dos módulos involucrados en la modificación de “GraseHotSpot”, “Content Manager” y el filtro de contenidos por direcciones “IP”, por ende, fue un reto integrar “MySql” a estos módulos puesto que uno se encuentra en el servidor y el otro se ejecuta en el lado del cliente, trayendo como consecuencia, que se consuma dicha información desde diferentes subredes de la topología planteada, finalmente, esto genera una alta susceptibilidad de presentar errores al momento de realizar modificaciones.

Durante la etapa de modificación, se reconoce la diferencia de complejidad entre un proyecto funcional que se encuentra en etapa de producción y proyectos de carácter académicos, en donde muchas veces no existe un alto grado de prioridad en el uso de patrones de arquitectura de “software” como por ejemplo el modelo vista controlador o “MVC”, mientras que en este caso particular, “GraseHotSpot” implementa dichos patrones con el fin de aplicar buenas prácticas de programación, es por esta razón, que se encontró una herramienta de desarrollo útil pero hasta el momento desconocida llamada “Smarty”, la cual permite trabajar ordenadamente para separar la vista del controlador, siendo esto un factor de suma importancia para el desarrollo de proyectos grandes.

En donde más inconvenientes se presentaron, fue en la red de dispositivos, debido a que no fue posible individualizar el tráfico de la forma en la que se planteó inicialmente, puesto que los datos requeridos como la dirección “MAC” o el “SSID” de los “AP”, no pueden ser extraídos por fuera de la red a la que pertenecen, siendo este problema causado por la ubicación de un “switch” que genera dos subredes adicionales, una para cada “AP”. Seguidamente, se exploraron posibles alternativas haciendo uso de las direcciones “IP” de los clientes, las cuales pueden ser obtenidas y manipuladas en la capa de aplicación, por lo tanto, se planteó asignar diferentes rangos de direcciones “IP” para cada zona a través del “DHCP” de los “APs”, sin embargo, se evidencio que el modulo “DnsMasq” encargado de asignar las direcciones “IP” a los clientes, las reasigna nuevamente con el fin de identificarlos al interior del servidor, por ende se imposibilita la individualización del tráfico de red de forma precisa para cada “AP” y se propone una solución alternativa planteada en la prueba de implementación descrita en el capítulo 10.

A causa de los múltiples problemas hallados, se propone en el capítulo 11, dos posibles alternativas de solución bastante prometedoras para individualizar el tráfico de red, las cuales continúan acotadas en ser soluciones administrables y completamente viables.

En el planteamiento de la alternativa de solución de filtrado mediante “IP”:”Port”, la dirección “IP” permite identificar la red inalámbrica, mientras que el puerto permite identificar los clientes. Es de suma importancia reconocer el papel que juegan estos parámetros, ya que el filtrado de trafico de red sin la implementación de alguno de estos dos, no permite implementar correctamente las funcionalidades de este proyecto.

El uso del lenguaje de programación del lado del servidor “PHP”, permitió extraer de forma precisa la dirección “IP” de los clientes conectados en la subred del servidor, esto gracias al uso del arreglo de ámbito “superglobal” de “PHP”

“$_SERVER[ ]”, el cual almacena diversas variables con información de la red proporcionada por el servidor “GraseHotSpot”, como por ejemplo “$_SERVER['HTTP_CLIENT_IP']”, la cual contiene la dirección “IP” del cliente que este ejecutando el “script”, para posteriormente ser usada en el filtrado del de trafico de red y desplegar el portal cautivo correspondiente a la zona de cobertura de cada “AP”. Gracias a la implementación de “PHP”, se logró el resultado esperado en el filtrado de contenidos planteado en la prueba de implementación del capítulo 10.

La implementación de la matriz de selección realizada a las alternativas de portales cautivos previamente caracterizados a través de diferentes criterios considerados pertinentes, tuvo un rol extremadamente importante, debido a que a dicha metodología de selección permitió escoger el portal cautivo más adecuado facilitando considerablemente el desarrollo de este proyecto. Si este proceso se hubiese omitido, no habría forma de tener una visión general de las consideraciones que implica la implementación y modificación de un proyecto robusto como lo es un portal cautivo, como por ejemplo la cantidad de módulos, el lenguaje de código fuente y demás aspectos relevantes que influyeron considerablemente en el desarrollo de este proyecto.

BIBLIOGRAFÍA

AALTO, Lauri; GOTHLIN; Nicklas; KORHONEN, Jani y OJALA, Timo. Bluetooth and WAP Push Based Location-Aware Mobile Advertising System [en línea].[Consultado 08 de Marzo de 2016]. Disponible en Internet: http://www.mediateam.oulu.fi/publications/pdf/496.pdf

About GRASE Hotspot [en línea]. Grase Hotspot, 2011. [Consultado 5 de noviembre de 2016]. Disponible en Internet: https://grasehotspot.org/about/

About the Apache HTTP server Project [en línea]. The Apache Software Foundation, 1997. [Consultado 30 de enero de 2017]. Disponible en Internet: https://httpd.apache.org/ABOUT_APACHE.html

Acceso a Internet [en línea]. Metas. Ministerio de tecnologías de la información y las comunicaciones. [Consultado 03 de Marzo de 2016] Disponible en Internet: http://micrositios.mintic.gov.co/vivedigital/2014-2018/proposito.php?lg=18

ANDRE, Fernando; PELLEJERO, Izaskun. WLAN: Fundamentos y aplicaciones de seguridad. Barcelona: Marcombo S.A., 2006. 160 p.

ANDREU, Joaquin. Servicios en red. Madrid: Editex S.A., 2011. 300 p.

Browser Statistics [en línea]. W3Schools. [Consultado 11 de marzo de 2016]. Disponible en Internet: http://www.w3schools.com/browsers/browsers_stats.asp

COLEMAN, David. CWNA Certified Wireless Network Administrator Official Study Guide. Indianápolis: Wilkey, 2009. 768 p.

CoovaChilli - man pages [en línea]. Coova.github.io. [Consultado 14 de agosto de 2016]. Disponible en Internet: http://coova.github.io/CoovaChilli/man-pages.html

CoovaChilli. [Consultado 14 de agosto de 2016]. Disponible en Internet: https://coova.github.io/CoovaChilli/

Doc – WiFiDog [en línea]. WiFiDog, 2010. [Consultado 10 de Agosto de 2016]. Disponible en Internet: http://dev.wifidog.org/wiki/doc

Documentación y procedimientos sobre Zeroshell [en línea]. Fulvio Ricciardi, 2005. [Consultado 7 de agosto de 2016]. Disponible en Internet: http://www.zeroshell.net/es/documentation/

EasyHotspot - Documentation [en línea]. Easyhotspot.inov.asia, 2010. [Consultado 14 de agosto de 2016]. Disponible en Internet: http://coova.github.io/CoovaChilli/man-pages.html

EasyHotSpot. [Consultado 08 de agosto de 2016]. Disponible en Internet: http://easyhotspot.inov.asia/index.php/documentation#introduction

El sistema operativo GNU [en línea]. Free Software Foundation, Inc, 2016. [Consultado 30 de enero de 2017]. Disponible en Internet: https://www.gnu.org/licenses/licenses.es.html#GPL

GitHub. [Consultado 22 de mayo de 2017]. Disponible en Internet: https://github.com/coova/coova-chilli

Global Public Wi-Fi Hotspots Will Reach 7.8 Million in 2015 and Continue To Grow at a CAGR of 11.2% through 2020 [en línea]. Singapore: ABI Research, 2015. [consultado 29 de Marzo de 2016]. Disponible en Internet : https://www.abiresearch.com/press/global-public-wi-fi-hotspots-will-reach78-million/

HERNANDEZ LÓPEZ, Jorge Angel. Implementación de un portal cautivo para la autenticación de usuarios en redes usando herramientas de software libre [en línea]. Tesis ingeniero en computación. Ciudad de México: Universidad Nacional Autónoma de México. Facultad de Ingeniería. 2012. 142 p. [Consultado el 18 de Marzo de 2016]. Disponible en: http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/2644/tesis.pdf?sequence=1

HUCABY, David. CCNP Routing and Switching SWITCH 300-115. Indianapolis: Cisco Press, 2014. 324 p.

HUCABY, David. Cisco Field Manual. Indianapolis: Cisco Press, 2002. 657 p.

Internet Sharing Devices. Newcastle: TextBlue. [Consultado 01 de Marzo de 2016]. Disponible en Internet: http://www.textblue.net/?page=textblue_internet_sharing_devices

MALDONADO, Angel. Implementación de un portal cautivo que permita el control de acceso al servicio de Internet a los estudiantes del colegio San Luis Gonzaga a través de una autenticación de los usuarios mediante un servicio AAA implementado en un servidor que trabaje con protocolo RADIUS. Tesis ingeniero de sistemas. Quito: Universidad Politécnica Salesiana. Facultad de Ingeniería, 2012. 130 p.

MANCERA, Jenny. La era del marketing digital y las estrategias publicitarias en Colombia [en línea]. Bogotá: Universidad Nacional de Colombia, 2013. [Consultado el 19 de Marzo de 2016]. Disponible en Internet: http://www.fce.unal.edu.co/uifce/proyectos-de-estudio/pdf/La%20era%20del%20Marketing%20Digital

MARQUÉS, Guillermo. AAA y FreeRadius: Lulu, 2016. 38 p.

MILLER, Michael. The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities Are Changing the World. Indianapolis: Que, 2015. 336 p.

MUELA, Clara. La publicidad en Internet: situación actual y tendencias en la comunicación con el consumidor [en línea]. En: ZER. 2008, Vol. 13. No. 24, p.132-201. [Consultado el 19 de Marzo de 2016]. Disponible en Internet: http://www.ehu.eus/zer/hemeroteca/pdfs/zer24-08-muela.pdf

MySql Customers [en línea]. Oracle Corporation, 2017. [Consultado 30 de enero de 2017]. Disponible en Internet: https://www.mysql.com/customers/

OpenWISP: public wifi [en línea]. openWISP.org, 2016. [Consultado 6 de noviembre de 2016]. Disponible en Internet: http://openwisp.org/whatis.html

Packet Fence: About [en línea]. Inverse, 2016. [Consultado 6 de noviembre de 2016]. Disponible en Internet: https://packetfence.org/about.html

PepperSpot – The next generation captive portal [en línea]. Thibault Vançon, 2012. [Consultado 7 de noviembre de 2016]. Disponible en Internet: http://pepperspot.sourceforge.net/index.php?n=Doc.UserDocumentation

PfSense: Main page [en línea]. Rubicon Communications, 2015. [Consultado 8 noviembre de 2016]. Disponible en Internet: https://doc.pfsense.org/index.php/Main_Page

PHP: Explode – manual [en línea]. The PHP Group, 2001. [Consultado 25 de febrero de 2017]. Disponible en internet: http://php.net/manual/es/function.explode.php

Premios [en línea]. Galardones otorgados al plan Vive Digital. Ministerio de tecnologías de la información y las comunicaciones. [Consultado 03 de Marzo de 2016] Disponible en Internet: http://micrositios.mintic.gov.co/vivedigital/logros-plan/logro.php?lg=27

Redes inalámbricas en los países en desarrollo. 4 ed. Hacker Friendly LLC., 2013. 530 p. [consultado 04 de marzo de 2016]. Disponible en Internet: http://wndw.net/pdf/wndw3-es/wndw3-es-ebook.pdf

Reporte de industria del sector TIC [en línea]. Comisión de Regulación de Comunicaciones, 2015. [Consultado 03 de marzo de 2016]. Disponible en Internet: http://colombiatic.mintic.gov.co/602/articles-13464_archivo_pdf.pdf

SAWANT, Uday. Ubuntu Server Cookbook. Birmingham: Packt Publishing, 2016. 456 p.

SCARFONE, Karen; DICOI, Derrick; SEXTON, Matthew, TIBBS, Cyrus. Guide to Security Legacy IEEE 802.11 Wireless Networks [En Línea]: Recommendations of the National Institute of Standars and Technology. Gaithersburg: National Institute of Standars and Technology, 2008. [consultado el 17 de Marzo de 2016]. Disponible en Internet: http://csrc.nist.gov/publications/nistpubs/800-48-rev1/SP800-48r1.pdf

SOENGAS, Xosé; VIVAR, Hipólito; ABUIN, Natalia. Del consumidor analógico al digital [En Línea]: Nuevas estrategias de publicidad y marketing para una sociedad hiperconectada. En: TELOS: cuadernos de comunicación e innovación. Junio-Septiembre, 2015, 12 p.

SOLANO, Johanna; OÑA, Mercedes. Estudio de portales cautivos de gestión de acceso inalámbrico a internet de la ESPOCH [en línea]. Tesis de grado ingeniero en sistemas informáticos. Riobamba: Escuela Superior Politécnica de Chimborazo. Facultad de informática y electrónica. 2009. 164 p. [Consultado el 18 de Marzo de 2016]. Disponible en: http://dspace.espoch.edu.ec/bitstream/123456789/103/1/18T00381.pdf

TANENBAUM, Andrew. Redes de computadoras. 5 ed. Naucalpan de Juárez: Pearson, 2012. 816 p.

Usage of server-side programming languages for websites [en línea]. W3Techs. Web Technology Surveys. [Consultado 11 de marzo de 2016]. Disponible en Internet: http://w3techs.com/technologies/overview/programming_language/all

WBA Industry Report 2011 [En Línea] : Global Developments In Public Wi-Fi. London: Wireless Broadband Alliance, 2011. [consultado 29 de Marzo de 2016]. Disponible en Internet: http://www.wballiance.com/wba/wp-content/uploads/downloads/2012/07/16_WBA-Industry-Report2011-_Global-Developments-in-Public-Wi-Fi-1.00.pdf 5

WifiDog. [Consultado 10 de Agosto de 2016]. Disponible en Internet: http://dev.wifidog.org/wiki/Screenshots

YUAN FENG, Xiuyan. RSSI-based algorithm for indoor localization [en línea]. Scientific Research. Qingdao: Ocean University of China. 2013. 6 p. [Consultado el 5 de junio de 2017]. Disponible en: http://file.scirp.org/pdf/CN_2013071010352139.pdf

ZeroShell. [Consultado 07 de agosto de 2016]. Disponible en Internet: http://www.zeroshell.org/hotspot-router/

ANEXOS

Los anexos que se mencionan a lo largo de este documento se entregan en el CD en formato digital.

ANEXO A: CÓDIGO FUENTE DE LA MODIFICACIÓN DEL ARCHIVO "HOTSPOT.PHP".

```
<?php

$conn;
   $file1;
   $range1;
   $file2;
   $range2;
   connect();
   if(connect()){
      getCaptivePortalsData();
      mysqli_close($GLOBALS['conn']);
   }
   function connect(){
      $servername = "localhost";
      $username = "root";
      $password = "";
      $dbname = "content_manager";
      $GLOBALS['conn'] = mysqli_connect($servername, $username, $password,
$dbname);
      if (!$GLOBALS['conn']) {
         return false;
      }else{
         return true;
      }
   }
   function getCaptivePortalsData(){
      $sql = sprintf("SELECT id,dir,max_ip FROM `accesspoints`");
      $res = $GLOBALS['conn']->query($sql);
      if ($res->num_rows > 0) {
         while($row = $res->fetch_assoc()) {
            if($row["id"]==1){
               $GLOBALS['file1'] = $row["dir"];
               $GLOBALS['range1'] = $row["max_ip"];
```

```
            }
            if($row["id"]==2){
                $GLOBALS['file2'] = $row["dir"];
                $GLOBALS['range2'] = $row["max_ip"];
            }
        }
    }
  }
 function get_client_ip() {
    $ipaddress = '';
    if (isset($_SERVER['HTTP_CLIENT_IP']))
        $ipaddress = $_SERVER['HTTP_CLIENT_IP'];
    else if(isset($_SERVER['HTTP_X_FORWARDED_FOR']))
        $ipaddress = $_SERVER['HTTP_X_FORWARDED_FOR'];
    else if(isset($_SERVER['HTTP_X_FORWARDED']))
        $ipaddress = $_SERVER['HTTP_X_FORWARDED'];
    else if(isset($_SERVER['HTTP_FORWARDED_FOR']))
        $ipaddress = $_SERVER['HTTP_FORWARDED_FOR'];
    else if(isset($_SERVER['HTTP_FORWARDED']))
        $ipaddress = $_SERVER['HTTP_FORWARDED'];
    else if(isset($_SERVER['REMOTE_ADDR']))
        $ipaddress = $_SERVER['REMOTE_ADDR'];
    else
        $ipaddress = 'UNKNOWN';
    return $ipaddress;
 }
 function is_app_1(){
   $client_ip = get_client_ip();
   $last_ip1_number = explode(".", $client_ip);
   if ($last_ip1_number[3] <= $GLOBALS['range1']) {
     return true;
   }elseif($last_ip1_number[3] >= $GLOBALS['range2']) {
     return false;
   }
 }
?>

<!doctype html>
<!--[if gt IE 8]><!--> <html class="no-js" lang=""> <!--<![endif]-->
   <head>
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
      <title></title>
      <meta name="description" content="">
      <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
      <link rel="stylesheet" href="css/bootstrap.min.css">
      <link rel="stylesheet" href="css/bootstrap-theme.min.css">
      <link rel="stylesheet" href="css/main.css">
      <script src="js/vendor/modernizr-2.8.3-respond-1.4.2.min.js"></script>
      <script src="js/vendor/jquery-1.11.2.min.js"></script>
      <script>
        $(function(){
          $("#App_1").load("<?php echo htmlspecialchars($file1);?>");
          $("#App_2").load("<?php echo htmlspecialchars($file2);?>");
        });
      </script>
    </head>
    <body>
      <?php
        if(is_app_1()){
          echo "<div id='App_1'></div>";
        }else{
          echo "<div id='App_2'></div>";
        }
      ?>
      <script
src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
      <script>window.jQuery || document.write('<script src="js/vendor/jquery-
1.11.2.min.js"><\/script>')</script>
      <script src="js/vendor/bootstrap.min.js"></script>
      <script src="js/main.js"></script>
    </body>
</html>
require_once('includes/site.inc.php');

load_templates(array('loginhelptext', 'belowloginhtml', 'termsandconditions',
'aboveloginhtml'));

if(isset($_GET['disablejs'])){
    // Set cookie
    setcookie('grasenojs','javascriptdisabled', time()+60*60*24*30);
    // Redirect via header to reload page?
    header("Location: http://$lanIP:3990/prelogin");
}

if(isset($_GET['enablejs'])){
    // Set cookie
    setcookie('grasenojs','', time()-60*60*24*30);
    // Redirect via header to reload page?
    header("Location: http://$lanIP:3990/prelogin");
```

```
}

$res = @$_GET['res'];
$userurl = @$_GET['userurl'];
$challenge = @$_GET['challenge'];

if($userurl == 'http://logout/') $userurl = '';
if($userurl == 'http://1.0.0.0/') $userurl = '';

if($Settings->getSetting('disablejavascript') == 'TRUE'){
   $nojs = true;
   $smarty->assign("nojs" , true);
   $smarty->assign("js" , false);
   $smarty->assign("jsdisabled" , true);
}elseif( isset($_COOKIE['grasenojs']) && $_COOKIE['grasenojs'] ==
'javascriptdisabled'){
   $nojs = true;
   $smarty->assign("nojs" , true);
   $smarty->assign("js" , false);
}else{
   $nojs = false;
   $smarty->assign("nojs" , false);
   $smarty->assign("js" , true);
}

$smarty->assign("user_url", $userurl);
$smarty->assign("challenge", $challenge);
$smarty->assign("RealHostname", trim(file_get_contents('/etc/hostname')));

if($Settings->getSetting('autocreategroup')){
   $smarty->assign('automac', true);
}

if(!isset($_GET['res'])){
   // Redirect to prelogin
      header("Location: http://$lanIP:3990/prelogin");
}

require_once '../radmin/automacusers.php';
if(@$_GET['automac']){
   automacuser();
   exit;
}

switch($res){
```

```
    case 'already':
        if($nojs){
            $smarty->display('loggedin.tpl');
            exit;
        }
        break;
    case 'failed':
        $reply = array("Login Failed");
        if($_GET['reply'] != '') $reply = array($_GET['reply']);
        $smarty->assign("error", $reply);
    case 'notyet':
    case 'logoff':
        setup_login_form();
        break;
    case 'success':
        if($_GET['uid'] == mactoautousername($_GET['mac']))       {
            break;
        }
        load_templates(array('loggedinnojshtml'));
        $smarty->display('loggedin.tpl');
        exit;
        break;
}

function setup_login_form(){
    global $smarty;
    $smarty->display('portal.tpl');
    exit;
}
$smarty->display('portal.tpl');
```

ANEXO B. CÓDIGO FUENTE DEL “FRONT-END” DEL PORTAL DE ADMINISTRADOR DE CONTENIDOS.

```
 <!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>App_admin</title>
   <link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
   <link rel="stylesheet" href="assets/css/styles.css">
</head>
<body>
   <div class="jumbotron">
      <h1 class="text-center">CONTENT MANAGER</h1>
   </div>
   <form name="access_points" method="post" action="database_connector.php">
      <!--Form 1-->
      <div class="well">
         <div class="container">
            <div class="row">
               <div class="col-md-12">
                  <h2 class="text-success">Access point 1</h2>
               </div>
            </div>
            <div class="row">
               <div class="col-md-7">
                  <div class="form-group has-success">
                     <label class="control-label" for="file-input">Select captive portal
file:</label>
                     <input value="HOLA" type="file" name="file_input_1">
                     <!--required="true"-->
                  </div>
               </div>
               <div class="col-md-5">
                  <div class="form-group has-success">
                     <label class="control-label" for="text-input">Set last ip address
for this content:</label>
                     <input value="<?php echo htmlspecialchars($range1); ?>"
type="number" name="range1" min="1" max="254" step="1">
                  </div>
```

```
            </div>
          </div>
        </div>
      </div>

      <!--Form 2-->
      <div class="well">
        <div class="container">
          <div class="row">
            <div class="col-md-12">
              <h2 class="text-success">Access point 2</h2>
            </div>
          </div>
          <div class="row">
            <div class="col-md-7">
              <div class="form-group has-success">
                <label class="control-label" for="file-input">Select captive portal
file:</label>
                <input type="file" name="file_input_2">
              </div>
            </div>
            <div class="col-md-5">
              <div class="form-group has-success">
                <label class="control-label" for="text-input">Set last ip address
for this content:</label>
                <input value="<?php echo htmlspecialchars($range2); ?>"
type="number" name="range2" min="1" max="254" step="1">
              </div>
            </div>
          </div>
        </div>
       <div class="container">
          <button class="btn btn-success" type="submit">Save</button>
        </div>
    </form>
    <script src="assets/js/jquery.min.js"></script>
    <script src="assets/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

ANEXO C. CÓDIGO FUENTE DEL “BACK-END” DEL PORTAL DE ADMINISTRADOR DE CONTENIDOS.

```
<?php
   $conn;
   $dir1;
   $range1;
   $dir2;
   $range2;
   connect();
   if(connect()){
      getDataToForm();
      mysqli_close($GLOBALS['conn']);
   }
   function connect(){
      $servername = "localhost";
      $username = "root";
      $password = "";
      $dbname = "content_manager";
      $GLOBALS['conn'] = mysqli_connect($servername, $username, $password,
$dbname);
      if (!$GLOBALS['conn']) {
         return false;
      }else{
         return true;
      }
   }
   function getDataToForm(){

      $sql = sprintf("SELECT id,dir,max_ip FROM `accesspoints`");
      $res = $GLOBALS['conn']->query($sql);
      if ($res->num_rows > 0) {
         while($row = $res->fetch_assoc()) {
            if($row["id"]==1){
               $GLOBALS['dir1'] = $row["dir"];
               $GLOBALS['range1'] = $row["max_ip"];
               echo "dir: " .$GLOBALS['dir1']. " - max_ip: " . $GLOBALS['range1'].
"<br>";
            }
            if($row["id"]==2){
               $GLOBALS['dir2'] = $row["dir"];
```

```
                $GLOBALS['range2'] = $row["max_ip"];
                echo "dir: " .$GLOBALS['dir2']. " - max_ip: " . $GLOBALS['range2'].
"<br>";
            }
         }
      }
   }
        function upDateData(){
                $dir1=$_POST["file_input_1"];
                $range1=$_POST["range1"];
                $dir2=$_POST["file_input_2"];
                $range2=$_POST["range2"];
                $sql1 = "UPDATE `accesspoints` SET `dir` = '$dir1',`max_ip` =
'$range1' WHERE `accesspoints`.`id` = 1";
                $sql2 = "UPDATE `accesspoints` SET `dir` = '$dir2',`max_ip` =
'$range2' WHERE `accesspoints`.`id` = 2";
                                if (mysqli_query($GLOBALS['conn'], $sql1)) {
                   echo "<strong>Access point 1</strong> up dated successfully
<br>";
                } else {
                   echo "Error: " . $sql1 . "<br>" . mysqli_error($GLOBALS['conn']);
                }
                if (mysqli_query($GLOBALS['conn'], $sql2)) {
                   echo "<strong>Access point 2</strong> up dated successfully";
                } else {
                   echo "Error: " . $sql2 . "<br>" . mysqli_error($GLOBALS['conn']);
                }
        }
        echo "<br><a href='index.php'>Return to content manager</a>";
?>
```

ANEXO D. CÓDIGO FUENTE DEL “FRONT-END” PARA DESPLEGAR VIDEOS PUBLICITARIOS.

```
<!doctype html>
<!--[if gt IE 8]><!--> <html class="no-js" lang=""> <!--<![endif]-->
   <head>
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
      <title></title>
      <meta name="description" content="">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="apple-touch-icon" href="apple-touch-icon.png">
      <link rel="stylesheet" href="css/bootstrap.min.css">
      <style>
         body {
            padding-top: 50px;
            padding-bottom: 20px;
         }
      </style>
      <link rel="stylesheet" href="css/bootstrap-theme.min.css">
      <link rel="stylesheet" href="css/main.css">
      <script src="js/vendor/modernizr-2.8.3-respond-1.4.2.min.js"></script>
      <script src="js/main.js"></script>
   </head>
   <body>
            <!-- Main jumbotron for a primary marketing message or call to action -->
           <div class="jumbotron" style="padding-top: 0px;">
                    <div class="container">
                       <div class="row">
                                <div class="col-lg-4 col-md-4 col-sm-4 col-xs-3">
                                        <br>
                                        <img     src="img/logo-adidas.png"     class="img-
responsive" alt="logo">
                                </div>
                                <div class="col-lg-8 col-md-8 col-sm-8 col-xs-9">
                                        <h1     class="hidden-xs">GUAYOS     ACE     16
PURECONTROL</h1>
                                        <h3     class="hidden-lg     hidden-md     hidden-
sm">GUAYOS ACE 16 PURECONTROL</h3>
                                <p class="hidden-xs text-justify">Toma las decisiones.
Dirige la jugada. Este es tu territorio y nadie te lo va a quitar. Con los ACE 17, el
```

```
partido es tuyo. Estos guayos de fútbol incorporan un exterior sin cordones en adidas Primeknit que ofrece control total y comodidad inmediata. Diseñados para canchas de terreno firme.</p>
                        </div>
                        </div>
                    </div>
            </div>
            <div class="container-fluid">
                    <div class="row" >
                            <div id="video_container" class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
                                    <video id="myVideo" tabindex="0" autoplay muted>
                                      <source src="video/ace17.mp4">
                                    </video>
                            </div>
                    </div>
                        <div class="row">
                            <div class="col-lg-4 col-md-4 hidden-xs hidden-sm"></div>
                            <div class="col-lg-4 col-md-4 col-sm-12 col-xs-12">
                                    <button id="connectionButton" type="button" class="btn btn-primary btn-md btn-block disabled">
                                            Conectarse a internet
                                    </button>
                            </div>
                        </div>
            </div> <!-- /container -->
            <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
            <script>window.jQuery || document.write('<script src="js/vendor/jquery-1.11.2.min.js"><\/script>')</script>
            <script src="js/vendor/bootstrap.min.js"></script>
    </body>
    <style>
                video {
                        width: 100%;
                        height: auto;
                }
        </style>
</html>
```

ANEXO E. CÓDIGO FUENTE DEL “FRONT-END” PARA DESPLEGAR CUPONES DE RESTAURANTES INTERACTIVOS.

```
<!DOCTYPE html>
<html>

<head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>App_P_C_Cupones</title>
   <link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
   <link rel="stylesheet" href="assets/css/styles.css">
   <link rel="stylesheet" href="assets/css/custom.css">
   <link rel="stylesheet" href="assets/css/animate.css">
   <link rel="stylesheet" href="assets/css/animations.css">
</head>

<body>

      <div class="page-header animated zoomInDown">
         <h2 class="text-uppercase text-center text-primary hidden-xs hidden-sm" id="titulo">Obtén descuentos en tus restaurantes favoritos</h2>
         <h4 class="text-uppercase text-center text-primary hidden-md hidden-lg" id="titulo">Obtén descuentos en tus restaurantes favoritos</h4>
         <p class="text-center text-muted hidden-md hidden-lg" id="Subtitulo">Selecciona el cupón y muestra el código en la caja.</p>
         <p class="text-center text-muted hidden-xs hidden-sm" id="SubtituloGrande">Selecciona el cupón y muestra el código en la caja.</p>
      </div>


   <div class="container">

      <div class="row">
         <div class="col-md-4 col-sm-4 col-xs-12">
            <img type="button" data-toggle="modal" data-target="#modal" class="img-circle img-responsive animated pulse" src="assets/img/1 - Donas.jpg">
               <div class="row animated infinite pulse">
                  <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12 Calificacion">
                     <span class="label label-success">105 veces solicitado</span>
                  </div>
```

```
                <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12 Cailficacion">
                  <i class="glyphicon glyphicon-star" id="estrella_buena"></i>
                  <i class="glyphicon glyphicon-star"></i>
                </div>
              </div>
          </div>
            <div class="pt-page-rotateInNewspaper modal fade" role="dialog" tabindex="-1" id="modal">
              <div class="modal-dialog" role="document">
                <div class="modal-content">
                  <div class="modal-header">
                    <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">×</span></button>
                    <h4 class="text-uppercase text-center text-primary modal-title">Las mejores donas para deleitar tu paladar</h4><br></div>
                  <div class="modal-body">
                    <p>Tu código de descuento es: <strong>EBF-14D-33E</strong></p>
                    <div class="row">
                      <div class="col-lg-2 col-md-2 col-sm-2 col-xs-2"></div>
                      <div class="col-lg-8 col-md-8 col-sm-8 col-xs-8"> <img type="button" class="img-rounded img- img-responsive animated infinite pulse" src="assets/img/1 - Donas.jpg"></div>
                      <div class="col-lg-2 col-md-2 col-sm-2 col-xs-2"></div>
                    </div>

                  </div>
                  <div class="modal-footer">
                    <button onclick="enableConnection()" class="btn btn-success" type="button" data-dismiss="modal">Aceptar</button>
                  </div>
                </div>
              </div>
            </div>
<div class="row">
    <div class="col-lg-4 col-md-3 col-sm-1 col-xs-1"></div>
    <div class="col-lg-4 col-md-6 col-sm-10 col-xs-10">
      <button id="connectionButton" class="btn btn-success btn-block disabled" type="button">
      Conectarse a internet</button>
    </div>
    <div class="col-lg-4 col-md-3 col-sm-1 col-xs-1"></div>
  </div>
  <script src="assets/js/jquery.min.js"></script>
  <script src="assets/bootstrap/js/bootstrap.min.js"></script>
```

```
    <script src="assets/js/custom.js"></script>
</body>
</html>
```