ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROTOTIPO DE UN SISTEMA DE PROTECCIÓN PERIMETRAL Y NAVEGACIÓN SEGURA PARA ENTORNOS DOMÉSTICOS

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

CARLOS ANDRÉS CUEVA ANCHAPAXI
andres-cueva@outlook.com

DIRECTOR: MSc. JOSÉ ANTONIO ESTRADA JIMÉNEZ
jose.estrada@epn.edu.ec

Quito, julio 2016

DECLARACIÓN

Yo, Carlos Andrés Cueva Anchapaxi, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

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

_____

Carlos Andrés Cueva Anchapaxi

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Carlos Andrés Cueva Anchapaxi, bajo mi supervisión.

_____

MSc. José Antonio Estrada

DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Agradezco a todos quienes hicieron posible este trabajo, especialmente a mis padres y al MSc. José Antonio Estrada.

DEDICATORIA

A mis amados padres Isabel y Manuel

A mi amado sobrino Jeremy

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE CÓDIGO

RESUMEN

El presente proyecto describe el desarrollo de un prototipo de un sistema de protección perimetral y navegación segura para entornos domésticos mediante el uso de herramientas *open source*. El prototipo permite aplicar un control general sobre el tráfico de salida en capa de red, aplicar un control personalizado en el acceso web y recolectar y visualizar información sobre la navegación web realizada con los *host* de la red de un hogar.

El prototipo está conformado por un servidor proxy HTTP, un firewall, un sistema recolector de información de navegación web y una interfaz de administración. Se utiliza como plataforma de cómputo un computador Raspberry Pi y una distribución Linux.

El firewall se encarga de filtrar tráfico en capa de red. El servidor proxy HTTP controla el acceso web y registra información sobre las peticiones HTTP que procesa. El sistema recolector de información de navegación web procesa los registros de las peticiones HTTP procesadas por el servidor proxy HTTP para extraer la información de navegación web requerida y la almacena en una base de datos. La interfaz de administración permite utilizar las funcionalidades del prototipo; puede ser accedida localmente a través de un navegador web.

La documentación sobre el desarrollo de este prototipo se encuentra organizada en cuatro capítulos.

En el capítulo 1 se describen los conceptos fundamentales sobre los servicios de red utilizados para el control del tráfico saliente, en capa de red y capa aplicación, de una red privada. Se realiza una revisión de varias herramientas *open source* que pueden ser utilizadas en una solución de protección perimetral y navegación segura. Se describen las características del computador Raspberry Pi. Se describen las características de la metodología de desarrollo de software Extreme Programming. Finalmente, se presenta el concepto de usabilidad y un método de evaluación.

En el capítulo 2 se determinan los requerimientos para el desarrollo del prototipo, y en base a estos se documentan las historias de usuario. Se planifican las

iteraciones y entregas. Finalmente, se describen los componentes del prototipo, a través de los cuales se implementan los requerimientos.

En el capítulo 3 se describe el desarrollo de los componentes del prototipo. Se documentan las pruebas realizadas para verificar la implementación de los requerimientos y el rendimiento del prototipo, entre las que se incluyen: pruebas unitarias, pruebas de aceptación, evaluación de la usabilidad de la interfaz de administración y prueba de carga en procesador y memoria. Finalmente, se presenta el costo referencial de desarrollo del prototipo.

En el capítulo 4 se presentan conclusiones y recomendaciones obtenidas en el trabajo realizado.

PRESENTACIÓN

Debido al incremento masivo del acceso a Internet desde los hogares, en cada domicilio se tiene pequeñas redes privadas conectadas a la red pública. Se entiende aquí, por tanto, como entornos domésticos de red aquellos conformados por una red en un hogar. Al igual que ocurre en entornos de red corporativos, en los ámbitos domésticos también es una preocupación el tipo de información a la que pueden tener acceso miembros vulnerables de una familia (como los niños).

Sin embargo, las herramientas y sistemas que permiten controlar el tráfico hacia Internet normalmente son costosos ya que se orientan a empresas, y su administración y uso requiere de conocimientos técnicos.

Teniendo en cuenta la necesidad de una solución de protección perimetral y navegación segura para una red en un hogar, en el presente proyecto se desarrolla un prototipo de solución telemática que permita al jefe del hogar, de manera relativamente sencilla, aplicar un control de acceso a servicios de Internet básico, aplicar un control de acceso web personalizado, así como visualizar información sobre la navegación web realizada.

CAPÍTULO 1

1. FUNDAMENTO TEÓRICO

1.1 RED EN UN HOGAR

La masificación del acceso a Internet en los hogares debido a la reducción de los costos de acceso [1] [2] ha provocado que, en estos hogares, se formen pequeñas redes privadas a las que se conectan prácticamente todos los miembros de la familia para acceder a Internet. Estas redes privadas, en el nivel más sencillo, están conformadas por uno o dos computadores. Sin embargo, en la actualidad, debido al bajo costo de dispositivos electrónicos, la red en un hogar puede estar conformada por varios de estos dispositivos, como: celulares, *tablets,* computadores portátiles, etc., generalmente conectados a Internet de forma inalámbrica a través de un de punto de acceso, *router* multifuncional (ver Figura 1.1), para acceder a varios servicios: servicio web, DNS (*Domain Name System*), correo electrónico, descarga y transferencia de archivos, entre otros.

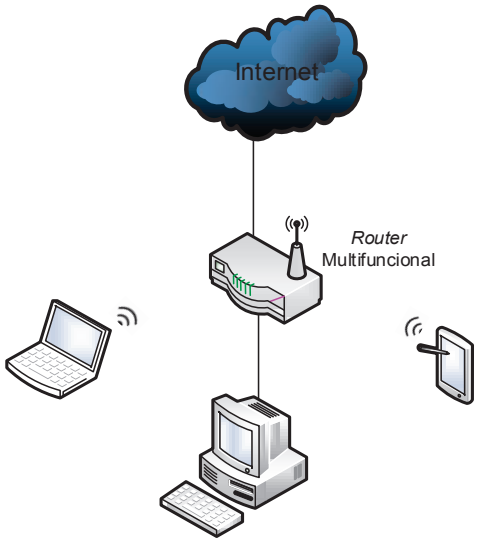

Figura 1.1 Esquema de un entorno doméstico de red

1.2 PROTECCIÓN PERIMETRAL Y NAVEGACIÓN SEGURA

Un sistema de protección perimetral y navegación segura se ubica en el borde de una red de comunicaciones, y puede estar compuesto por uno o más servicios, e. g., firewall, proxy, filtro web, IDS (*Intrusion Detection System*), que permiten

gestionar el tráfico que entra y sale de la red, con el propósito de proteger tanto los recursos internos de la red como a los usuarios en el acceso a la red pública.

1.2.1 SERVICIOS

El presente proyecto se concentrará en el control del tráfico de salida[1]. Por ello, en esta sección se describe características importantes de los servicios de red que permitan cumplir esta función.

1.2.1.1 Firewall

El firewall es el encargado de permitir o denegar tráfico entre redes. En una red de comunicaciones, usualmente es el primer componente de un sistema de seguridad perimetral. Para que la utilización de un firewall sea efectiva, éste debe ser instalado en el punto donde la red interna se conecta con el Internet, tal como se observa en la Figura 1.2. Considerando que al pasar todo el tráfico por el firewall, y con la configuración correspondiente, es posible aceptar o rechazar determinado tipo de tráfico.

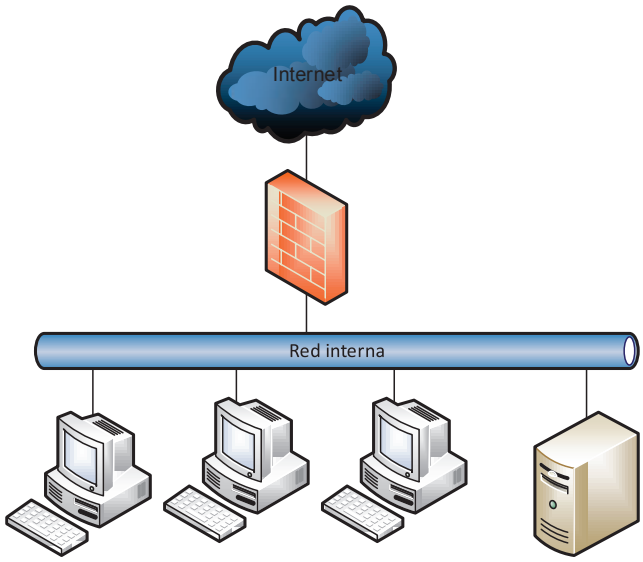

Figura 1.2 Firewall entre la red interna y el Internet

1.2.1.1.1 Criterios de diseño

Un firewall puede ser diseñado considerando dos enfoques [3]:

[1] Tráfico de salida: es aquel tráfico generado por los *hosts* de una red interna con destino fuera de la red en la que se encuentran, generalmente en Internet.

- *Lo que no está explícitamente permitido está prohibido:* El firewall, por defecto, debería bloquear todo el tráfico entrante y saliente, y que cada servicio o aplicación requerida debería ser solicitada al administrador para que la acepte explícitamente en el firewall. Esto incrementa el nivel de seguridad porque solo ciertos servicios son permitidos.
- *Lo que no está explícitamente prohibido está permitido*: El firewall, por defecto, debería permitir todo el tráfico, y que cada servicio potencialmente dañino debería ser bloqueado cuando se evidencie el riesgo. Esto crea un ambiente flexible con más servicios disponibles para los usuarios, pero disminuye el nivel de seguridad.

1.2.1.1.2 Filtrado de paquetes [3]

Un firewall acepta o deniega (filtra) un paquete, en función de ciertos parámetros de la cabecera de un paquete IP o de un segmento TCP (*Transmission Control Protocol*) o UDP (*User Datagram Protocol*). Esta información puede ser: la dirección IP (*Internet Address*) origen y destino, el protocolo encapsulado (e. g., TCP, UDP, ICMP), el puerto TCP/UDP de origen y destino, tipo de mensaje ICMP (*Internet Control Message Procotol*), interfaz de red de entrada y de salida del paquete.

Beneficios del filtrado de paquetes:

- La mayoría de dispositivos que trabajan en la capa de red (*routers*) poseen sistemas de firewall que se utilizan para realizar filtrado de paquetes; en este caso existe poco o ningún costo para implementar el filtrado de paquetes ya que en el software de un *route*r está incluida esta característica.
- El filtrado de paquetes generalmente es transparente para los usuarios y las aplicaciones, debido a que no es necesaria una configuración en cada *host*.

Limitaciones del filtrado de paquetes:

- La definición de reglas de filtrado es una tarea que necesita de conocimientos sobre los servicios de Internet, formato de los paquetes y los valores que se esperaría encontrar en cada campo.

- Si un filtro complejo es requerido, la regla de filtrado puede llegar a ser tan larga y complicada haciendo difícil su comprensión y administración.
- Hay pocas facilidades para verificar, de forma inmediata, si una regla ha sido configurada correctamente por lo que puede dar paso a una vulnerabilidad.
- Un firewall de filtrado de paquetes normalmente no trabaja en capas superiores a la capa de red o de transporte, por lo que no permite tomar decisiones con base en la información en la capa de aplicación. Esto es un problema, ya que se pueden realizar ataques enviando comandos o funciones en la capa de aplicación de los paquetes permitidos por el firewall.

1.2.1.2 Proxy [3] [4]

Un proxy actúa como un intermediario entre dos redes, por lo general entre una red local y el Internet. Es considerado un firewall de capa aplicación o un componente de un sistema de firewall que mejora la capacidad de filtrado de paquetes. Un proxy realiza una intermediación de peticiones y respuestas de protocolos de servicios de Internet, generalmente Web y FTP (*File Transfer Protocol*), lo que le permite inspeccionar el contenido y aplicar un filtrado de tráfico en capa aplicación.

Por cada servicio de Internet se necesita un servidor proxy, por lo que, en ocasiones, una configuración en el lado del cliente debe es necesaria. Por ejemplo, para una conexión a través de un proxy HTTP (*Hypertext Transfer Protocol*), en ocasiones, es necesaria la configuración del navegador web del cliente para establecer la dirección IP y puerto del servidor proxy a través del cual se realizará la conexión.

1.2.2 SOLUCIONES PARA ENTORNOS DOMÉSTICOS

Para un entorno doméstico de red las soluciones de protección perimetral y navegación segura son escasas, ya que por lo general, estas se orientan a entornos más complejos. Sin embargo, existen herramientas o sistemas que pueden ser utilizadas como parte de una solución de este tipo en un entorno doméstico.

1.2.2.1 *Router* de Acceso

En un hogar con acceso a Internet, el *router* (*router* doméstico) provisto por el proveedor de servicios de Internet (ISP) para acceder a Internet surge como la solución más básica para un entorno doméstico. Sin embargo, sacarle provecho a esta solución requiere conocimientos técnicos básicos de redes TCP/IP. Como es el punto por donde circula el tráfico, hacia y desde Internet, este dispositivo, en mayor o menor medida, puede controlar un tipo de tráfico determinado. Entre las funcionalidades que puede tener un *router* doméstico se tiene:

- Filtrado de paquetes por número de puerto de origen o destino, dirección IP origen o destino, MAC origen.
- Control de acceso por horario de acceso.
- Filtrado web por nombre de host de destino.

1.2.2.2 Iptables

Iptables es un programa que se incluye como parte del *kernel* Linux desde su versión 2.4. Esta herramienta es utilizada para administrar el conjunto de reglas de las tablas de filtrado de paquetes del *kernel* Linux. Iptables permite construir un firewall, manipular campos de la cabecera de un paquete IP, configurar NAT (*Network Address Translation*) para compartir una conexión o permitir acceso externo a servicios de una red privada [5].

1.2.2.3 Squid [6]

Squid es una aplicación *open source* que implementa un servidor proxy-cache web que soporta los protocolos HTTP, HTTPS (*Hypertext Transfer Protocol Secure*) y FTP. Squid puede ser utilizada para almacenamiento en *cache*, controlar el acceso web de los *hosts* de una red privada mediante el filtrado de peticiones HTTP, registrar información de las transacciones HTTP que luego puede ser procesada para extraer información sobre la navegación web de los *hosts* de la red privada.

1.2.2.4 Dansguardian [7]

Dansguardian es un programa de línea de comandos para filtrar contenido web. Entre los mecanismos de filtrado se tiene: comparación de palabras, filtrado URL

(*Uniform Resource Locator*) y filtrado mediante PICS[2]. Un aspecto importante sobre Dansguardian es que tiene que trabajar en conjunto con Squid para intermediar las peticiones HTTP y aplicar el filtrado web.

1.2.2.5 ClearOS [8] [9]

Es una distribución Linux con funciones de un servidor y plataforma de red. Está diseñado para hogares, pequeños negocios y ambientes empresariales distribuidos. La distribución es flexible, puede ser descargada de forma gratuita, e incluye una gran lista de características y servicios integrados, los cuales pueden ser configurados a través de una interfaz web. Entre las herramientas que forman parte de ClearOS se tiene: sistema de detección de intrusos, filtrado web, firewall, administrador de ancho de banda, servidor de correo electrónico.

1.3 RASPBERRY PI

Es un computador de placa reducida, desarrollado por la Fundación Raspberry Pi con el propósito de proveer a personas de todas las edades, en especial a niños y adolescentes, un ambiente económico para explorar la informática [10]. El modelo de Raspberry Pi utilizado en este proyecto es el modelo B+ (ver Figura 1.3.)

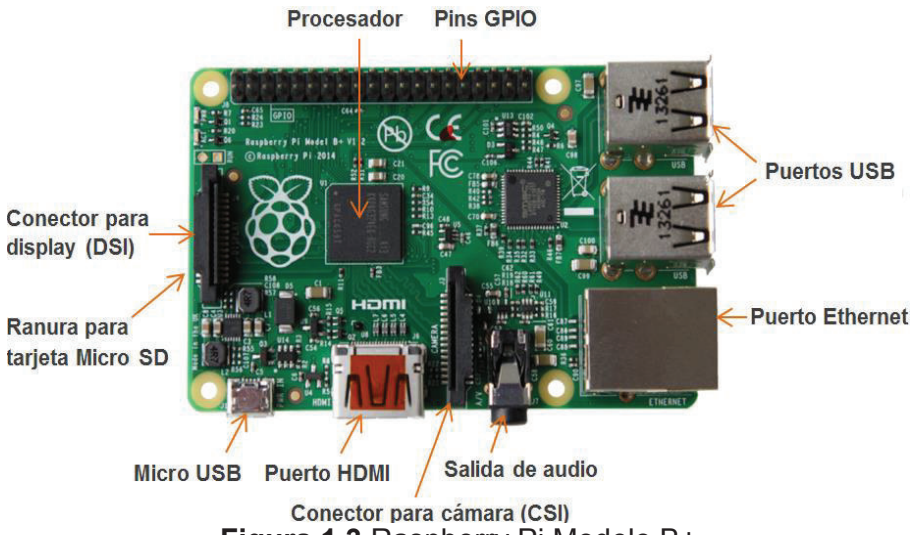

Figura 1.3 Raspberry Pi Modelo B+

Las principales características del hardware de un Raspberry Modelo B+ se describen en la Tabla 1.1.

[2] PICS (*Platform for Internet Content Selection):* Especificación que permite asociar etiquetas (*metadata*) con contenido de Internet [50].

Elemento	Descripción
Procesador	Procesador ARM Broadcom BCM2835 de 700Mhz, 512MB de RAM.
Puertos USB	4 puertos USB 2.0, cada uno tiene una capacidad de 100[mA] hasta 500[mA]. Para dispositivos que superen el consumo límite de corriente, estos deben tener su propia fuente de energía. Para el caso de un mouse y teclado para controlar el Raspberry Pi la corriente suministrada es más que suficiente.
Puerto micro USB	A través del puerto micro USB se conecta una fuente de 5 [V], el modelo B típicamente consume de 700 [mA] a 1000 [mA]
Ranura para tarjeta *Secure Digital* (SD)	Para la memoria micro SD que almacena toda información para que el Raspberry Pi funcione.
Puerto HDMI (*High-Definition Multimedia Interface*)	Salida de video digital.
Salida de audio	*Jack* estándar de audio analógico.
Puerto Ethernet	Puerto Ethernet que soporta una velocidad de hasta 100Mbps.
Pins de entrada y salida de propósito general (GPIO)	Mediante la programación de estos *pins* un Raspberry puede interactuar con otros dispositivos.
Interfaz serial para cámara (CSI)	Permite conectar directamente al Raspberry un módulo de cámara.
Interfaz serial para display (DSI)	Para comunicarse con un LCD o pantalla de *display* OLED (*Organic Light-Emitting Diode*).

Tabla 1.1 Descripción del hardware de un Raspberry Pi modelo B+

Las características de procesamiento, interfaces para interactuar con diferente hardware y software libre compatible con un Raspberry Pi permiten desarrollar una gran variedad de aplicaciones telemáticas. A continuación, se describen tres aplicaciones telemáticas que brindan una idea del potencial de un computador Raspberry Pi.

Firewall y sistema de detección de Intrusos: solución que implementa un sistema de seguridad que realiza un control y monitoreo del tráfico que entra y sale de una pequeña red de comunicaciones [11].

Centro multimedia: un Raspberry Pi es una opción con gran potencial para construir un centro multimedia con OSMC [12], en el que se puede ver videos HD, series de TV, escuchar música, entre otros servicios.

Mayordomo de puerta de garaje: sistema que realiza un control de acceso en la puerta del garaje con notificaciones de correo electrónico, mensajes SMS de

quien entra o sale, captura de video en la entrada, y administración del sistema de forma remota a través de un sitio web [13].

Lo más destacable de un computador Raspberry Pi es su bajo costo en relación a la capacidad de procesamiento y memoria que ofrece y la gran comunidad que se encuentra detrás apoyando su desarrollo y promoviendo su utilización.

1.4 METODOLOGÍA DE DESARROLLO DE SOFTWARE

El desarrollo de software no es una tarea sencilla, la creación de un producto de software requiere de un proceso en el cual se debe realizar varias actividades o tareas (obtención y análisis de requerimientos, planificación del proceso, diseño, etc.) que pueden resultar complejas o difíciles de administrar. Por tal razón, una metodología de desarrollo es utilizada, con el fin de ayudar a estructurar, planificar y controlar las actividades involucradas en el proceso de desarrollo de software [14].

Existe una gran variedad de metodologías, las cuales pueden ser clasificadas de forma general en metodologías tradicionales y metodologías ágiles.

1.4.1 METODOLOGÍAS TRADICIONALES [14] [15]

Las metodologías tradicionales proponen un proceso disciplinado para el desarrollo de software con el objetivo de hacerlo predecible y eficiente. Esto es realizado mediante un proceso detallado con un fuerte énfasis en la planificación. El desarrollo es secuencial y sigue un plan de proyecto bien definido desde el inicio del proceso. Otra característica es la falta de flexibilidad, lo cual resulta en un alto costo cuando es necesario implementar un cambio.

En este tipo de metodologías se prioriza la documentación, planificación y procesos (plantillas, técnicas de administración, revisiones, etc.).

1.4.2 METODOLOGÍAS ÁGILES

Las metodologías ágiles surgen como alternativa al enfoque tradicional. Este enfoque promueve un punto intermedio entre un desarrollo de software sin procesos y un desarrollo con demasiados procesos [15]. Son flexibles, preparadas para responder a cambios, particularmente en cuanto a requerimientos, y

considerando el desarrollo de software como una actividad colectiva con la participación del cliente.

El desarrollo de software es iterativo e incremental. En lugar de entregar un producto de software completo al final del ciclo de desarrollo, se planifican entregas de versiones funcionales en períodos cortos, cada versión con mayor funcionalidad que la anterior. Las versiones son evaluadas para garantizar la calidad del producto de software.

Varias metodologías ágiles han sido desarrolladas y se puede mencionar las siguientes:

- Scrum
- Extreme Programming
- Crystal Clear
- Agile Unified Process

Una metodología ágil es apropiada para proyectos pequeños (e. g., desarrollo de prototipos), cuando no se conoce todos los requerimientos al iniciar el proyecto y con poco tiempo para el desarrollo.

A continuación se presenta las características de la metodología ágil de desarrollo de software Extreme Programming que es tomada como referencia para el desarrollo del prototipo del presente proyecto.

1.4.3 EXTREME PROGRAMMING [16]

Extreme Programming (o XP) constituye un conjunto de valores, principios y prácticas para el desarrollo de software de alta calidad de la forma más rápida posible [17]. Las ideas propuestas en XP no son nuevas. La innovación que propuso XP es la de reunir buenas prácticas de desarrollo existentes bajo un enfoque común, que se pueden apoyar entre sí lo mejor posible y que puedan ser aplicadas en el proceso de desarrollo.

XP se basa en 4 valores con el fin de establecer un ambiente productivo para desarrollar software. Estos valores son:

- *Comunicación:* El equipo de desarrollo y el cliente trabajan en conjunto. A través de la comunicación, el equipo completo puede encontrar la mejor

solución que cumpla con los requisitos y resuelva las dudas del cliente. Además, a través de este valor se comparten experiencias de problemas en el desarrollo, con el objetivo de solucionarlos y prevenirlos en el futuro.

- *Simplicidad:* Este valor fomenta la búsqueda de una solución lo más simple posible que funcione correctamente.
- *Retroalimentación:* Con cada entrega de software funcional se recibe una retroalimentación por parte del equipo y los clientes. Esta información es valiosa para detectar y prevenir problemas.
- *Coraje:* La metodología XP estimula la toma de decisiones, para asumir retos y afrontar problemas de manera directa.

1.4.3.1 Prácticas [18]

Extreme Programming se caracterizada por las prácticas que promueve. Las prácticas más importantes son:

- *Planificación:* Es realizada en conjunto el cliente y el equipo de desarrollo. Esta práctica está dada por las historias de usuario. Una historia de usuario describe, de forma simple y rápida, un problema a ser resuelto (si una historia tiene muchos detalles, es mejor expresar esos detalles como historias de usuario adicionales). El equipo de desarrollo determina cuánto esfuerzo necesita cada historia de usuario y cuántas de ellas puede producir en un período determinado, conocido como iteración.
- *Pequeñas entregas:* En lugar de una entrega completa al final de un largo ciclo de desarrollo, se realizan entregas de versiones del sistema en cortos períodos. La versión siguiente con mayor funcionalidad que la anterior. Los beneficios de aplicar esta práctica son: entregar valor real en cortos períodos (el cliente siente confianza y satisfacción), obtener rápidamente una retroalimentación de parte del cliente (permite una fácil adaptación a posibles cambios en los requerimientos), los desarrolladores tienen una sensación de logro y reducción del riesgo porque se corrigen rápidamente errores encontrados.
- *Pruebas:* Se realizan dos tipos de pruebas: unitarias y aceptación. Las pruebas unitarias son escritas por los desarrolladores para verificar la funcionalidad del código; este tipo de pruebas, generalmente es aplicado a

porciones de código: clases, métodos o a un pequeño grupo de clases. Por otro lado, las pruebas de aceptación son desarrolladas a partir de las historias de usuario y especificadas por el cliente, para validar el funcionamiento del sistema; en una prueba de aceptación se valida la funcionalidad en todos los escenarios posibles.

- *Diseño simple:* Crear un diseño simple, formado por solo lo necesario para satisfacer los requisitos del cliente
- *Refactorización de código:* Como la evolución del software es continua, el código puede no ser óptimo en un inicio. Esta práctica sugiere restructurar el código para hacerlo más simple, eliminar duplicidad y para que a su vez sea fácil de mantener. La refactorización es realizada bajo demanda, cuando sea necesario, e. g., antes o después de agregar nuevo código por nuevos requerimientos o cambios en los mismos.
- *Programación en parejas:* 2 programadores producen código en la misma máquina. Los programadores alternan roles (programar, supervisar), mientras el uno codifica, el otro revisa las acciones tomadas o proporciona ideas para mejorar la toma de decisiones.
- *Integración continua:* Integración y prueba de código continua, con el fin de no acumular problemas y descubrirlos tarde cuando sea complicado corregirlos.
- *El cliente es parte del equipo de desarrollo:* El cliente participa en la toma de decisiones acerca de los requerimientos y en la definición de pruebas de aceptación.
- *Estándares de codificación:* El código es escrito de acuerdo a lineamientos establecidos para facilitar la comunicación de los programadores mediante el código.

1.4.3.2 Ciclo de vida [19] [20]

El ciclo de vida de XP incluye las fases de Planificación, Desarrollo de las iteraciones, Mantenimiento y Muerte.

Planificación

En esta fase el cliente define lo que necesita mediante historias de usuarios. El equipo de desarrollo entiende las historias para determinar qué tareas deben ser

realizadas para cumplir con estas necesidades. A continuación se estima el tiempo requerido para completar el desarrollo de las historias de usuario, y de acuerdo al tiempo y prioridad de las historias de usuario en el desarrollo del sistema, se determina el orden en que deben implementarse.

Desarrollo de las iteraciones

En esta fase las historias de usuario son implementadas, en cada iteración planificada se realiza un análisis, diseño, codificación y pruebas. Se debe mencionar que el diseño debe ser simple que funcione, que pueda ser evaluado, que pueda ser implementado de forma clara, sin duplicidad, y que tenga sólo los elementos suficientes.

Mantenimiento

La naturaleza de XP permite integrar nueva funcionalidad, por lo tanto en esta fase se implementan las nuevas historias de usuario, solicitadas por el cliente y se libera una nueva versión del sistema.

Muerte

Una vez que no existen historias de usuario por ser implementadas, la fase de Muerte ha sido alcanzada. En esta fase se realiza una documentación precisa y útil acerca del sistema desarrollado, para que cuando se requiera realizar algún tipo de mantenimiento del sistema en el futuro, esta tarea pueda ser realizada sin inconvenientes.

1.5 USABILIDAD [21]

La usabilidad es un atributo de calidad que valora cuán fácil de utilizar es una interfaz de usuario. Es una medida subjetiva que depende del sistema, la audiencia objetivo y el contexto de utilización.

1.5.1 ATRIBUTOS DE LA USABILIDAD

La usabilidad está definida o compuesta por los siguientes atributos [21]:

Facilidad de aprendizaje

Es un atributo importante en la usabilidad. Este atributo es considerado por varios investigadores como el atributo más fundamental de la usabilidad [22]. A pesar

que existe un acuerdo sobre su importancia, no existe una única definición de facilidad de aprendizaje. De acuerdo a [23], la mayoría de definiciones han abordado la experiencia inicial de uso e incluyen criterios como: capacidad de alcanzar cierto nivel de experticia y eficiencia, que pueden ser utilizados para medir la facilidad de aprendizaje.

Eficiencia

Una vez que los usuarios han aprendido el diseño de la interfaz de usuario, que tan rápido pueden utilizar correctamente las funcionalidades del sistema.

Recuerdo en el tiempo

Qué tan fácil es utilizar correctamente el sistema después de no utilizarlo por un cierto período.

Errores

Cuantos errores cometen los usuarios al utilizar el sistema. Es un atributo negativo, por la tanto, mientras menos errores se tengan, el sistema es considerado más usable.

Satisfacción

Es un atributo subjetivo que indica cuán agradable fue la experiencia del usuario al utilizar el sistema.

1.5.2 EVALUACIÓN DE USABILIDAD

Evaluar la usabilidad de una interfaz de usuario durante el proceso de diseño permite descubrir errores y mejorar el diseño final, mientras que, evaluar la usabilidad de una interfaz de usuario de un sistema en producción tiene el objetivo de comparar el sistema con otro de similares características o determinar si los requerimientos se han cumplido [24].

Los métodos para evaluar la usabilidad pueden ser agrupados en las siguientes categorías [25]: comprobación de directrices y normas seguidas para el diseño, evaluación realizada por expertos, evaluación utilizando modelos y simulaciones, evaluación con usuarios potenciales. De estos métodos, el más fundamental y útil, es la evaluación con usuarios potenciales o prueba de usuario [21].

La forma más efectiva para conocer lo que funciona o no en una interfaz de usuario es observar a los potenciales usuarios mientras interactúan con la interfaz; esto se logra a través de una prueba de usuario [21]. A continuación se describen aspectos fundamentales que deben ser considerados para realizar una prueba de usuario.

Audiencia objetivo [26]

Antes de planear y realizar una prueba de usuario, se debe identificar el grupo de usuarios que representen la audiencia objetivo del sistema. La información que se obtiene depende de los participantes que se recluten, si los participantes son los potenciales usuarios, la información obtenida en la prueba será de utilidad.

Objetivos a cumplir en la interfaz [27]

Son actividades que un usuario puede realizar utilizando la interfaz del sistema; estas actividades suelen ser las funcionalidades del sistema.

Escenarios [27]

Un escenario es utilizado para describir una actividad en un contexto familiar para el usuario, con el propósito de comprometerlo a realizar la actividad.

Observación [23]

Mientras se observa a los usuarios interactuar en la interfaz, se debe recolectar la mayor cantidad de información posible. La recolección de información puede ser realizada mediante herramientas de software, dispositivos de audio y video, o se puede tomar apuntes de forma manual. La información recolectada es analizada para corregir problemas en la interfaz de usuario o para validar los requerimientos de usabilidad.

CAPÍTULO 2

2. DISEÑO DEL PROTOTIPO

En este capítulo se presentan los requerimientos del prototipo, en base a estos se documentan las historias de usuario. Luego, se planifican las iteraciones y entregas. Finalmente, se describe la arquitectura del prototipo.

2.1 REQUERIMIENTOS DEL PROTOTIPO

El prototipo a desarrollar está enfocado a solucionar necesidades básicas de los jefes del hogar respecto al control del uso de Internet (control sobre el tráfico de salida), especialmente el uso del servicio web, en un entorno doméstico.

Para obtener información acerca de esas necesidades, se realizó una encuesta. Además, esta encuesta fue utilizada para obtener detalles sobre requerimientos iniciales del prototipo que fueron definidos en el plan de proyecto de titulación:

- Filtrado personalizado de tráfico web: por destino y por horario
- Bloqueo total de tráfico web en base a computador origen
- Lista de los sitios web más visitados
- Tiempo de navegación web
- Historial de navegación web

2.1.1 ENCUESTA

La encuesta fue constituida por preguntas para conocer la situación en los entornos domésticos en torno al uso del servicio web, conocer de forma general el uso de servicios de Internet y obtener información sobre los requerimientos iniciales del prototipo.

2.1.1.1 Población objetivo

La encuesta fue dirigida a los jefes, o padres de familia, de hogares con acceso a Internet. Para el cálculo de la muestra se tomó como referencia la cantidad de hogares con acceso a Internet de la provincia de Pichincha. De acuerdo al “Censo de población y vivienda en el Ecuador” del año 2010, en la provincia de Pichincha existen 190.920 hogares con acceso a Internet [28].

El cálculo del tamaño de la muestra se realiza con la ecuación (2.1).

$$n = \frac{Z^2 p(1-0,5)}{e^2} \quad (2.1)$$

La ecuación (2.1) es aplicada cuando el tamaño de la población es muy grande (a partir de 100.000), en dónde [29]:

n = El tamaño de la muestra que se calcula.

e = Es el nivel de precisión o margen de error de muestreo. Este margen se expresa en puntos porcentuales, por lo general está 1% y 10%. Para este estudio se consideró un margen de error de 8%.

$\mathbf{Z^2}$ = Es la desviación del valor medio que se acepta para lograr el nivel de confianza deseado. En función del nivel de confianza que se busque, se utiliza un valor determinado que viene dado por la forma que tiene la distribución de Gauss. El valor de Z puede ser consultado en tablas estadísticas, el valor correspondiente a un nivel de confianza de 95% es 1,96.

La muestra calculada, en base a una población de 190.920, es:

$$n = \frac{1,96^2 * 0,5 * (1 - 0,5)}{0,08^2} = 150,06$$

2.1.1.2 Información obtenida

La encuesta fue distribuida mediante redes sociales y correo electrónico (el cuestionario realizado se encuentra en el Anexo A). Se obtuvieron 155 respuestas de jefes del hogar con edades entre los 24 y 71 años. A continuación se presenta la información obtenida.

Conocimiento de los riesgos y nivel de preocupación por la información que puede ser encontrada al navegar en Internet

- La mayoría de los encuestados, 83%, conoce sobre riesgos a los cuales puede estar expuesta su familia cuando navega en Internet. Entre los riesgos conocidos se encuentran: acoso, comunicación con desconocidos, contenido inadecuado, robo de identidad, fraude, adicción a juegos online, etc.
- El 59% de los encuestados se sienten muy preocupados por el contenido inapropiado que podría encontrar su familia cuando accede a la Web.

Mecanismos para controlar el uso de Internet en el hogar

- Los encuestados aplican algunos mecanismos para tratar de controlar el uso de Internet: se establece un lugar específico donde utilizarlo (24% de las respuestas), un padre de familia supervisa personalmente el uso (34% de las respuestas) o se establece un horario de utilización (11% de las respuestas). En el caso de los encuestados que no aplican ningún tipo de regla, aquellos representan casi el 50%.

Nivel de preocupación por algunos sitios web de acuerdo al tipo de contenido

- Se solicitó a los encuestados que califiquen el nivel de preocupación que causan algunas categorías de sitios web en las que los miembros del hogar podrían navegar al acceder a la Web. Los resultados obtenidos pueden ser observados en el Figura 2.1.

Información de interés sobre la actividad web de los miembros del hogar

- Se solicitó a los encuestados que indiquen la información que les gustaría conocer acerca de la actividad web de sus hijos. La información de más interés es: alertas por intentos de acceso a sitios web bloqueados (72% de las respuestas), otra información de interés es: sitios web visitados (59% de las respuestas), historial web (57% de las respuestas) y la cantidad de tiempo de navegación web (36% de las respuestas).

Nivel de aceptación por algunas funcionalidades del prototipo

- La mayoría de los encuestados, 86%, considera de utilidad bloquear el acceso a sitios web relacionados a las categorías presentes en el Figura 2.1, en particular los sitios web de las categorías que causan mayor preocupación
- La mayoría de los encuestados, 85%, considera de utilidad limitar el tiempo de navegación web mediante un horario de acceso.
- Se solicitó sugerencias para horarios de acceso web en función de la edad de los miembros del hogar y la información obtenida es la siguiente: para niños, los horarios sugeridos varían en el transcurso de la tarde; para adolescentes, los horarios sugeridos varían durante la tarde y noche; y para mayores de 18 años, fue sugerido que el acceso web debería ser sin restricción de horario.

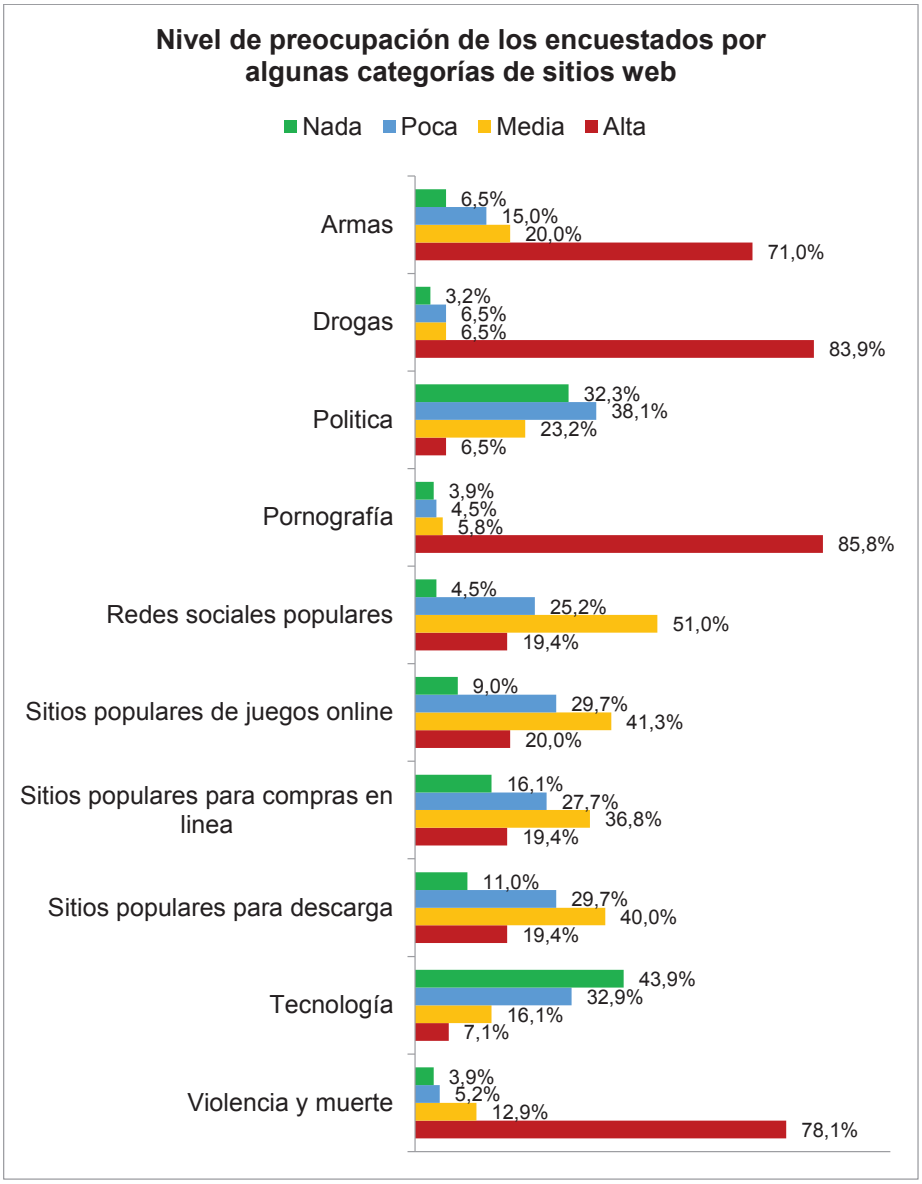

Figura 2.1 Nivel de preocupación por algunas categorías de sitios web

Se evidencia en los resultados de la encuesta que existe preocupación por la información que puede ser accedida por los miembros del hogar cuando navegan en Internet, esto seguramente es porque la mayoría de los encuestados conoce los riesgos a los que se expone una persona al acceder a la red pública.

De igual manera, los resultados permitieron conocer que los jefes del hogar utilizan mecanismos (e. g., se establece un lugar específico donde utilizarlo, un padre de familia supervisa personalmente el uso o se establece un horario de utilización) para tratar de controlar el uso de Internet en el hogar. Sin embargo,

estos métodos pueden resultar poco efectivos cuando el jefe del hogar no se encuentra en el hogar para aplicarlos.

2.1.2 USUARIOS Y ROLES

- **ADMINISTRADOR:** Jefe del hogar, encargado de la configuración de las funcionalidades del prototipo.

2.1.3 HISTORIAS DE USUARIO

En esta sección se documentan las historias de usuario, desarrolladas a partir de la información recopilada en la encuesta y los requerimientos iniciales del prototipo. Las historias de usuario son organizadas en los siguientes módulos:

- Administración de dispositivos y perfiles
- Control de acceso web
- Reportes de información de navegación web
- Control de acceso a servicios de Internet
- Configuración general

2.1.3.1 Administración de dispositivos y perfiles

En este módulo se agrupan las historias de usuario: Registrar dispositivo (ver Tabla 2.1), Eliminar dispositivo (ver Tabla 2.2), Perfil para agrupar dispositivos (ver Tabla 2.3), Modificar perfil (ver Tabla 2.4) y Eliminar perfil (ver Tabla 2.5).

Historia de usuario	**Número:** 1
Nombre de historia:	Registrar dispositivo
Prioridad de desarrollo:	Alta
Descripción: El administrador debe registrar con un nombre los dispositivos utilizados para navegar en la Web, con el propósito de identificar la fuente del tráfico web que luego podrá ser filtrado. Dispositivos no registrados no tienen acceso.	

Tabla 2.1 Historia de usuario Registrar dispositivo

Historia de usuario	**Número:** 2
Nombre de historia:	Eliminar dispositivo
Prioridad de desarrollo:	Media
Descripción: El administrador puede eliminar el registro de un dispositivo, como consecuencia de esta operación la información sobre la navegación web realizada con el dispositivo debe ser borrada.	

Tabla 2.2 Historia de usuario Eliminar dispositivo

Historia de usuario	**Número:** 3
Nombre de historia:	Perfil para agrupar dispositivos
Prioridad de desarrollo:	Alta
Descripción: El administrador debe crear un perfil y seleccionar los dispositivos que serán agrupados en el perfil.	

Tabla 2.3 Historia de usuario Perfil para agrupar dispositivos

Historia de usuario	**Número:** 4
Nombre de historia:	Modificar perfil
Prioridad de desarrollo:	Media
Descripción: El administrador puede cambiar el nombre del perfil, quitar o asignar nuevos dispositivos. Los dispositivos pueden estar libres o pertenecer a otros perfiles.	

Tabla 2.4 Historia de usuario Modificar perfil

Historia de usuario	**Número:** 5
Nombre de historia:	Eliminar perfil
Prioridad de desarrollo:	Media
Descripción: El administrador puede eliminar perfiles. Cuando un perfil sea eliminado, sus dispositivos asignados deben ser liberados y la información de navegación web de cada uno de los dispositivos debe ser borrada.	

Tabla 2.5 Historia de usuario Eliminar perfil

2.1.3.2 Control de acceso web

En el módulo Control de acceso web se agrupan las historias de usuario: Bloquear el acceso a sitios web de categorías (ver Tabla 2.6), Bloquear el acceso a sitios web específicos (ver Tabla 2.7), Permitir el acceso a sitios web específicos (ver Tabla 2.8), Eliminar registros de sitios web específicos (ver Tabla 2.9), Establecer horario de acceso web (ver Tabla 2.10), Bloquear completamente el acceso web (ver Tabla 2.11) y Permitir acceso web sin restricción de horario (ver Tabla 2.12).

Historia de usuario	**Número:** 6
Nombre de historia:	Bloquear el acceso a sitios web de categorías
Prioridad de desarrollo:	Alta
Descripción: El administrador puede bloquear el acceso a varios sitios web agrupados en ciertas categorías, a cada perfil agregado; estas categorías podrán ser seleccionadas de un grupo predefinido.	

Tabla 2.6 Historia de usuario Bloquear el acceso a sitios web de categorías

Historia de usuario	**Número:** 7
Nombre de historia:	Bloquear el acceso a sitios web específicos
Prioridad de desarrollo:	Alta
Descripción: El administrador puede ingresar el nombre de *host* de un sitio web cuyo acceso requiera ser bloqueado a un perfil.	

Tabla 2.7 Historia de usuario Bloquear el acceso a sitios web específicos

Historia de usuario	**Número:** 8
Nombre de historia:	Permitir el acceso a sitios web específicos
Prioridad de desarrollo:	Alta
Descripción: El administrador puede ingresar el nombre de *host* de un sitio web cuyo acceso requiera ser permitido a un perfil; opción utilizada cuando se requiera permitir el acceso a un sitio web perteneciente a una categoría bloqueada.	

Tabla 2.8 Historia de usuario Permitir el acceso a sitios web específicos

Historia de usuario	**Número:** 9
Nombre de historia:	Eliminar registros de sitios web específicos
Prioridad de desarrollo:	Alta
Descripción: El administrador puede escoger los sitios web, de la lista correspondiente (bloqueados o permitidos) de cada perfil, y eliminarlos.	

Tabla 2.9 Historia de usuario Eliminar registros de sitios web específicos

Historia de usuario	**Número:** 10
Nombre de historia:	Establecer horario de acceso web
Prioridad de desarrollo:	Alta
Descripción: El administrador puede establecer un horario de acceso web en cada día de la semana para cada dispositivo asignado a un perfil. Se debe mantener aplicado el filtrado web de las historias de usuario No.6, No.7 y No.8.	

Tabla 2.10 Historia de usuario Establecer horario de acceso web

Historia de usuario	**Número: 11**
Nombre de historia:	Bloquear completamente el acceso web
Prioridad de desarrollo:	Alta
Descripción: El administrador puede bloquear completamente el acceso web de forma individual a cada dispositivo asignado a un perfil.	

Tabla 2.11 Historia de usuario Bloquear completamente el acceso web

Historia de usuario	**Número:** 12
Nombre de historia:	Permitir acceso web sin restricción de horario
Prioridad de desarrollo:	Alta
Descripción: El administrador puede permitir el acceso web sin restricción de horario a cada dispositivo asignado a un perfil. Se debe mantener el filtrado web de las historias de usuario No.6, No.7 y No. 8.	

Tabla 2.12 Historia de usuario Permitir acceso web sin restricción de horario

2.1.3.3 Reportes de información de navegación web

En el módulo Reportes de información de navegación web se agrupan las historias de usuario: Intentos de acceso a sitios web bloqueados (ver Tabla 2.13), Historial web (ver Tabla 2.14), Buscar en historial web (ver Tabla 2.15), Sitios web más visitados (ver Tabla 2.16) y Tiempo de navegación web (ver Tabla 2.17).

Historia de usuario	**Número:** 13
Nombre de historia:	Intentos de acceso web bloqueados
Prioridad de desarrollo:	Alta
Descripción: El administrador puede visualizar y eliminar registros de intentos de acceso que son bloqueados (filtros: hoy, últimos 7 días, últimos 30 días, todo); cada registro está formado por: URL solicitada, fecha y hora del intento y nombre del dispositivo utilizado (asignado a un perfil). El administrador también puede eliminar registros específicos.	

Tabla 2.13 Historia de usuario Intentos de acceso a sitios web bloqueados

Historia de usuario	**Número:** 14
Nombre de historia:	Historial web
Prioridad de desarrollo:	Alta
Descripción: El administrador puede visualizar y eliminar registros de páginas web visitadas (filtros: hoy, últimos 7 días, últimos 30 días, todo); cada registro está formado por: URL, fecha y hora de la visita, y nombre del dispositivo utilizado (asignado a un perfil). El administrador también puede eliminar registros específicos.	

Tabla 2.14 Historia de usuario Historial web

Historia de usuario	**Número:** 15
Nombre de historia:	Buscar en historial web
Prioridad de desarrollo:	Alta
Descripción: El administrador puede ingresar una palabra para buscar en los registros del historial web; los resultados de la búsqueda serán los registros que contengan esa palabra.	

Tabla 2.15 Historia de usuario Buscar en historial web

Historia de usuario	**Número:** 16
Nombre de historia:	Sitios web más visitados
Prioridad de desarrollo:	Alta
Descripción: El administrador puede visualizar y eliminar registros de los sitios web más visitados (filtros: hoy, últimos 7 días, últimos 30 días, todo); cada registro está formado por: nombre de *host*, fecha y número de visitas y nombre del dispositivo utilizado (asignado a un perfil). El administrador también puede eliminar registros específicos.	

Tabla 2.16 Historia de usuario Sitios web más visitados

Historia de usuario	**Número:** 17
Nombre de historia:	Tiempo de navegación web
Prioridad de desarrollo:	Alta
Descripción: El administrador puede visualizar y eliminar registros tiempo de navegación web (últimos 7 días, últimos 30 días, todo); cada registro está formado por: fecha y cantidad de tiempo de navegación web y nombre del dispositivo utilizado (asignado a un perfil). El administrador también puede eliminar registros específicos.	

Tabla 2.17 Historia de usuario Tiempo de navegación web

2.1.3.4 Control de acceso a servicios de Internet

En el módulo Control de acceso a servicios de Internet se agrupan las historias de usuario: Registro servicios de Internet predefinidos (ver Tabla 2.18), Registro personalizado de servicios de Internet (ver Tabla 2.19) y Administración registros de servicios de Internet (ver Tabla 2.20).

Historia de usuario	**Número:** 18
Nombre de usuario:	Registro de servicios de Internet predefinidos
Prioridad de desarrollo:	Media
Descripción: Por defecto, todo el tráfico que no pase por el servidor proxy del prototipo debe ser bloqueado en capa de red. Para permitir el tráfico de un servicio, El administrador debe escoger el servicio de una lista predefinida y registrarlo. Cada *ítem* de la lista tendrá la siguiente información: nombre (no editable), número de puerto (identificador de servicio, no editable) y estado (habilitado/deshabilitado).	

Tabla 2.18 Historia de usuario Registro servicios de Internet predefinidos

Historia de usuario	**Número:** 19
Nombre de usuario:	Registro personalizado de servicios de Internet
Prioridad de desarrollo:	Media
Descripción: En el caso que el administrador requiera permitir el tráfico de un servicio que no se encuentre predefinido, podrá registrarlo (servicio personalizado) con la siguiente información: nombre, número de puerto y estado.	

Tabla 2.19 Historia de usuario Registro personalizado de servicios de Internet

Historia de usuario	**Número:** 20
Nombre de usuario:	Administración de registros de servicios de Internet
Prioridad de desarrollo:	Media
Descripción: El administrador puede editar o eliminar el registro de un servicio. En el caso de eliminar el registro o cambiar el valor del campo estado (a “deshabilitado”) el tráfico de ese servicio ya no está permitido.	

Tabla 2.20 Historia de usuario Administración registros de servicios de Internet

2.1.3.5 Configuración general

En el módulo configuración general se agrupan las historias de usuario: Restablecer configuración inicial (ver Tabla 2.21), Cuenta de administrador (ver Tabla 2.22), Autenticación (ver Tabla 2.23), Cambiar contraseña (ver Tabla 2.24), Centro de ayuda (ver Tabla 2.25).

Historia de usuario	**Número:** 21
Nombre de historia:	Restablecer configuración inicial
Prioridad de desarrollo:	Baja
Descripción: El administrador puede restablecer la configuración inicial del prototipo. Al utilizar esta función se borrará la configuración realizada por el administrador (registros de dispositivos, perfiles, reglas aplicadas, información de navegación web y servicios de Internet).	

Tabla 2.21 Historia de usuario Restablecer configuración inicial

Historia de usuario	**Número:** 22
Nombre de historia:	Cuenta de administrador
Prioridad de desarrollo:	Baja
Descripción: Por defecto, en el sistema estará creada una cuenta para el administrador, a la cual se puede modificar el nombre de usuario, contraseña, escoger una pregunta de seguridad de una lista predefinida (para cambiar contraseña sin ingresar al sistema) y cambiar la respuesta de esta pregunta.	

Tabla 2.22 Historia de usuario Cuenta de administrador

Historia de usuario	**Número:** 23
Nombre de historia:	Autenticación
Prioridad de desarrollo:	Baja
Descripción: El administrador debe ingresar al sistema con el nombre de usuario y contraseña de la cuenta de administrador.	

Tabla 2.23 Historia de usuario Autenticación

Historia de usuario	**Número:** 24
Nombre de historia:	Cambiar contraseña
Prioridad de desarrollo:	Baja
Descripción: El administrador debe ingresar el nombre de usuario y respuesta de seguridad de la cuenta de administrador si requiere cambiar la contraseña sin ingresar al sistema.	

Tabla 2.24 Historia de usuario Cambiar contraseña

Historia de usuario		**Número:** 25
Nombre de historia:	Centro de ayuda	
Prioridad de desarrollo:	Baja	
Descripción: El administrador puede acceder a un manual de usuario en la interfaz de administración para informarse cómo utilizar las diferentes funcionalidades		

Tabla 2.25 Historia de usuario Centro de ayuda

2.1.4 REQUERIMIENTOS FUNCIONALES

En la Tabla 2.26 se presentan los requerimientos funcionales que deberá cumplir el prototipo.

Código	Definición
RF01	El prototipo deberá permitir al administrador registrar dispositivos utilizados por los usuarios de la red interna para navegar en la Web, así como eliminar el registro.
RF02	El prototipo deberá permitir al administrador crear, modificar y eliminar perfiles para agrupar dispositivos.
RF03	El prototipo deberá permitir al administrador aplicar un control de acceso web a los dispositivos de un perfil en función del sitio web de destino: específicos y clasificados en categorías.
RF04	El prototipo deberá permitir al administrador aplicar un control de acceso web personalizado a los dispositivos de un perfil en función del horario de acceso.
RF05	El prototipo deberá permitir al administrador configurar el bloqueo de acceso web de forma individual a los dispositivos de un perfil.
RF06	El prototipo deberá permitir al administrador visualizar información sobre la navegación web realizada con los dispositivos de un perfil: historial web, tiempo de navegación web, sitios web más visitados e intentos de acceso web bloqueados.
RF07	El prototipo deberá permitir al administrador aplicar un control de acceso a servicios de Internet en función del número de puerto utilizado por cada servicio.
RF08	El prototipo deberá proveer al administrador la opción de restablecer la configuración inicial del sistema.
RF09	El prototipo deberá controlar el acceso a la interfaz de administración mediante un nombre de usuario y contraseña.
RF10	El prototipo deberá proveer al administrador la opción de cambiar la contraseña sin tener que ingresar a la interfaz de administración.
RF11	El prototipo deberá proveer en la interfaz de administración una sección con información sobre cómo utilizar las funcionalidades.

Tabla 2.26 Requerimientos funcionales

2.1.5 REQUERIMIENTOS NO FUNCIONALES

En la Tabla 2.27 se presentan los requerimientos no funcionales que deben ser considerados en el desarrollo del prototipo, los cuales fueron determinados en base al plan de proyecto de titulación.

Código	Definición
RNF01	La plataforma del prototipo estará compuesta por computador Raspberry Pi y una distribución Linux.
RNF02	El administrador deberá encontrar la interfaz de administración del prototipo fácil de utilizar; se evaluarán los atributos: facilidad de aprendizaje, eficiencia, satisfacción.
RNF03	El prototipo deberá desarrollarse utilizando herramientas *open source*.

Tabla 2.27 Requerimientos no funcionales

2.1.6 PLANIFICACIÓN

2.1.6.1 Plan de Iteraciones

La planificación de las iteraciones es realizada de acuerdo a la relación que existe entre las historias de usuario y prioridad que se tiene para su desarrollo. En la Tabla 2.29 y Tabla 2.29 se presenta el resultado de esta planificación y se indica la estimación del tiempo para implementar cada una de las historias. El tiempo es estimado tomando como referencia un día de 8 horas laborables.

No.	Historia de Usuario	Prioridad	Estimación (días)	Iteración asignada
1	Registrar dispositivo	Alta	3	Primera
3	Perfil para agrupar dispositivos	Alta	3	
6	Bloquear el acceso a sitios web de categorías	Alta	4	
7	Bloquear el acceso a sitios web específicos	Alta	1,5	
8	Permitir el acceso a sitios web específicos	Alta	1,5	
9	Eliminar registros de sitios web específicos	Alta	2	
10	Establecer horario de acceso web	Alta	5	Segunda
11	Bloquear completamente el acceso web	Alta	3	
12	Permitir acceso web sin restricción de horario	Alta	2	
13	Intentos de acceso web bloqueados	Alta	6	Tercera
14	Historial web	Alta	5	
15	Buscar en historial web	Alta	2	

Tabla 2.28 Plan de iteraciones

No.	Historia de Usuario	Prioridad	Estimación (días)	Iteración asignada
16	Sitios web más visitados	Alta	5	Cuarta
17	Tiempo de navegación web	Alta	6	
2	Eliminar dispositivo	Media	1	Quinta
4	Modificar perfil	Media	1	
5	Eliminar perfil	Media	1	
18	Registro de servicios de Internet predefinidos	Media	1	
19	Registro personalizado de servicios de Internet	Media	2	
20	Administración registros de servicios de Internet	Media	1	
21	Restablecer configuración inicial	Media	1,5	
22	Cuenta de administrador	Baja	1,5	
23	Autenticación	Baja	1	
24	Cambiar contraseña	Baja	1	
25	Centro de ayuda	Baja	2	

Tabla 2.29 Plan de iteraciones (continuación)

2.1.6.2 Plan de entregas

En la Tabla 2.30 se muestra el plan de entregas que se especifica el tiempo calendario para el desarrollo del prototipo. En un día se trabajan 8 horas, en una semana se trabajan 5 días y en un mes se tiene 20 días de trabajo.

Iteración	Estimación (días)	Estimación (semanas)	Estimación (meses)	Fecha de inicio	Fecha de Entrega
Primera	15	3,0	0,8	2015-08-01	2015-08-22
Segunda	10	2,0	0,5	2015-09-24	2015-09-08
Tercera	13	2,6	0,7	2015-09-10	2015-10-30
Cuarta	11	2,2	0,6	2015-10-03	2015-10-17
Quinta	14	2,8	0,7	2015-11-19	2015-12-08
Total	49	9,8	2,5		

Tabla 2.30 Plan de entregas

2.2 DISEÑO DE ARQUITECTURA DEL PROTOTIPO

El prototipo actúa como *gateway* de una red interna (generada con un Access Point - Switch) y la red doméstica en donde será instalado, tal como se observa

en la Figura 2.2. El tráfico entre los *hosts* de la red interna y el Internet circula por el prototipo, lo que le permite, con las aplicaciones adecuadas, filtrar tráfico de salida y recolectar información de navegación web.

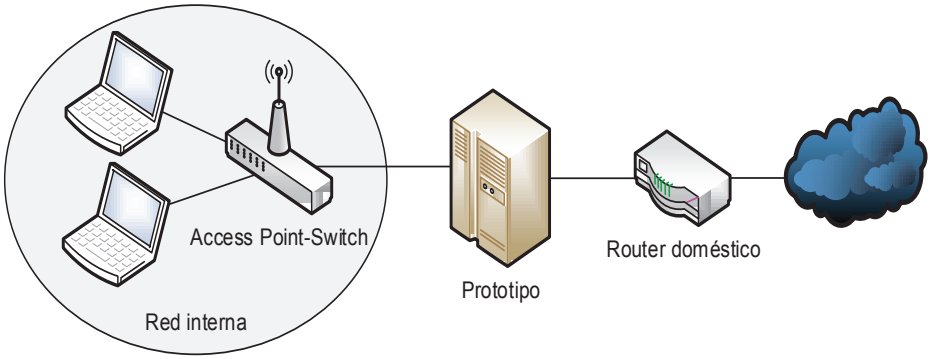

Figura 2.2 Esquema de red de un hogar con el prototipo instalado

En la Figura 2.3 se muestra la arquitectura del prototipo, la cual está conformada por tres componentes principales: filtrado de tráfico de salida, recolector de información de navegación web e interfaz de administración.

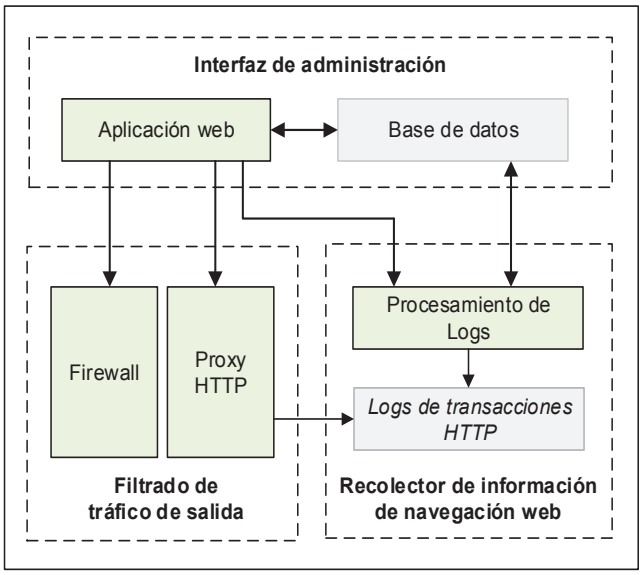

Figura 2.3 Arquitectura del prototipo

2.2.1 FILTRADO DE TRÁFICO

2.2.1.1 Firewall

La función del firewall es filtrar, en función del número de puerto (identificador de servicio), el tráfico de salida de la red interna. El criterio de diseño del firewall es:

lo que no está explícitamente permitido está prohibido; este enfoque, como fue mencionado en la sección 1.2.1.1.1, incrementa el nivel de seguridad en una red de comunicaciones.

2.2.1.2 Proxy HTTP

Un servidor proxy HTTP es utilizado para controlar el acceso web de los clientes de la red interna. Filtra peticiones HTTP en base a parámetros identificados en los requerimientos, los cuales son: origen o destino de la petición y horario de acceso. En la Figura 2.5 se ilustra el filtrado de peticiones HTTP que realizará el servidor proxy HTTP para cumplir con los requerimientos funcionales de control de acceso web. Otra función del servidor proxy, es la de registrar información (*logs*) de las transacciones HTTP procesadas.

2.2.2 RECOLECTOR DE INFORMACIÓN DE NAVEGACIÓN WEB

Este componente se encarga de procesar los *logs* de transacciones HTTP del servidor proxy HTTP, para extraer información de navegación web y almacenar esta información en una base de datos.

2.2.3 INTERFAZ DE ADMINISTRACIÓN

2.2.3.1 Aplicación web

A través de la aplicación web, el administrador utiliza las funcionalidades del prototipo. En general, esta aplicación permite configurar los servicios para filtrar tráfico de salida, ejecutar el programa de procesamiento de *logs* y manipular la base de datos.

2.2.3.1.1 Navegación entre las interfaces de usuario

La aplicación web está compuesta de varias interfaces gráficas (páginas web). En la Figura 2.4 se muestra un diagrama de navegación entre estas interfaces.

2.2.3.1.2 Diseño de Interfaces de usuario

Las interfaces de usuario fueron diseñadas con el propósito de que sean usables. Para conseguir este objetivo, el diseño fue realizado tomando como referencia algunas de las guías presentes en [30] [31] :

- Proporcionar mensajes de aviso antes de realizar acciones que cambien la configuración del prototipo.
- Proporcionar mensajes de resultado a las acciones realizadas por el administrador.
- Etiquetas con nombres claros y títulos que describan el contenido de la página.
- Utilizar lenguaje de usuario: palabras, frases y conceptos familiares al administrador.
- Distribución gráfica adecuada, espacio suficiente entre los elementos de la interfaz.
- Proporcionar ayuda.

Durante la fase de iteraciones, además de seguir las referencias descritas, se probaron implementaciones de los diseños de las interfaces de usuario con varios usuarios de un entorno doméstico, para identificar errores que impidan la facilidad de uso y corregirlos a tiempo. Como resultado de este proceso continuo de pruebas se obtuvo un diseño final que fue sujeto a evaluación, mediante una prueba de usuario. Detalles de esta prueba serán presentados en el capítulo 3, sección 3.3.3.

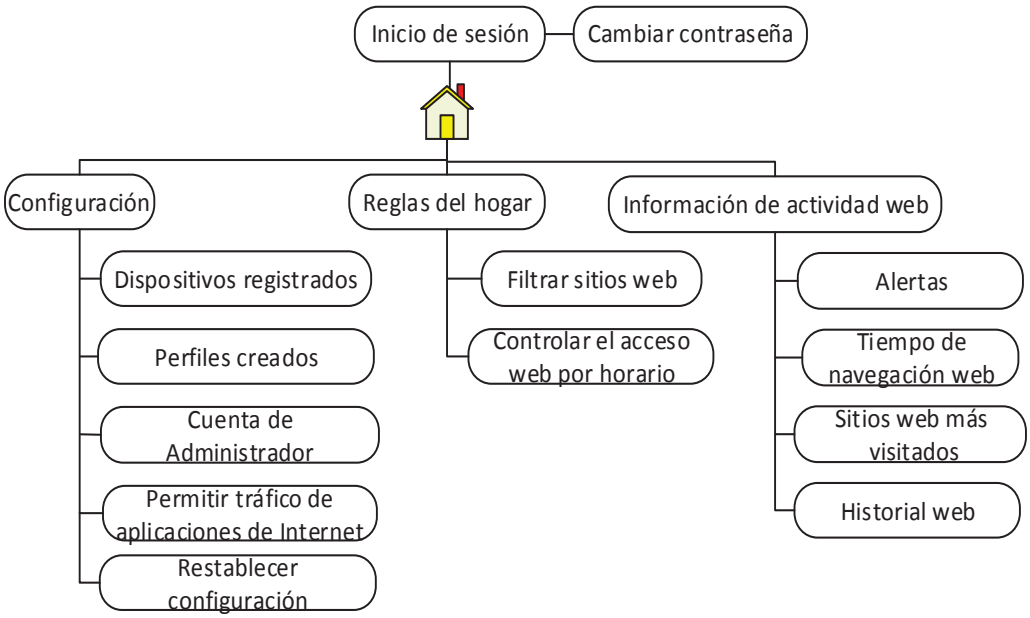

Figura 2.4 Diagrama de navegación entre las interfaces de usuario

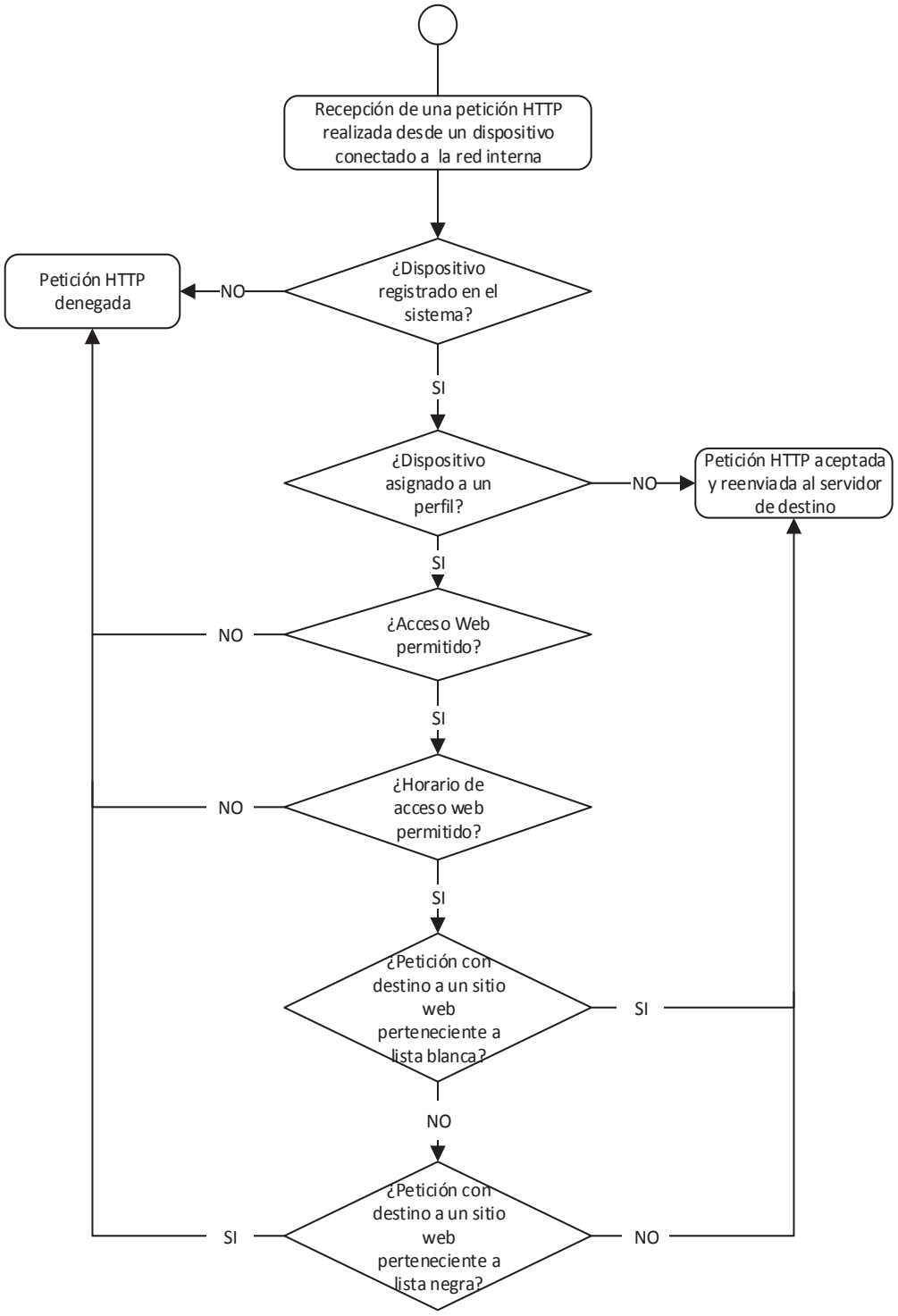

Figura 2.5 Diagrama de flujo del filtrado web realizado por el servidor proxy HTTP

2.2.3.1.3 Prototipado de interfaces de usuario

En esta sección se presentan los diseños de las principales interfaces de usuario: Configuración inicial (ver Figura 2.6), Reglas del hogar opción Filtrar sitios web (ver Figura 2.7), Reglas del hogar opción Control de acceso web por horario (ver Figura 2.8), Información de navegación web opción alertas (ver Figura 2.9), Permitir tráfico de servicios de Internet (ver Figura 2.10).

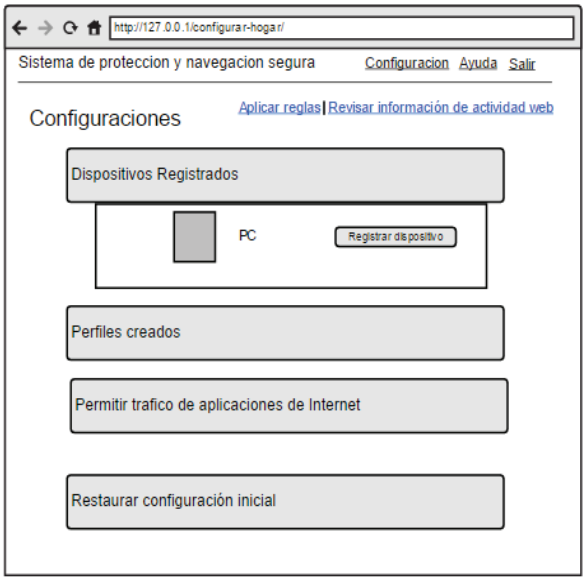

Figura 2.6 Diseño de interfaz Configuración inicial

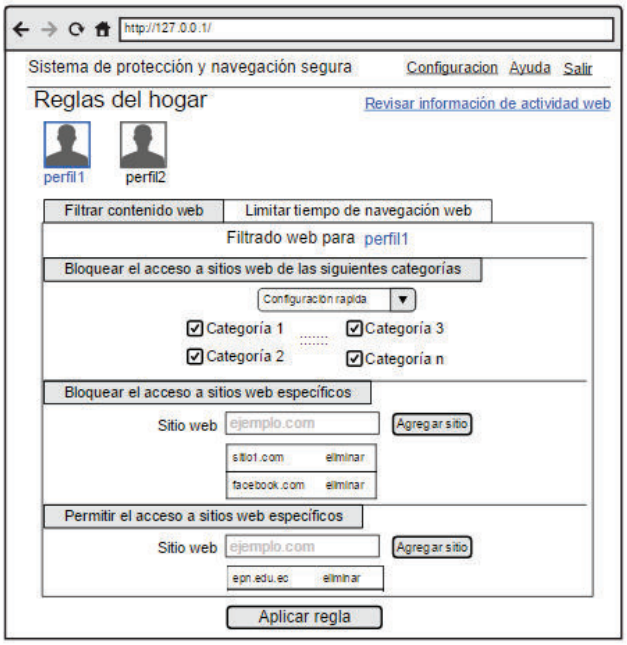

Figura 2.7 Diseño de interfaz Reglas del hogar opción Filtrar sitios web

Figura 2.8 Diseño de interfaz Reglas del hogar opción Controlar el acceso web por horario

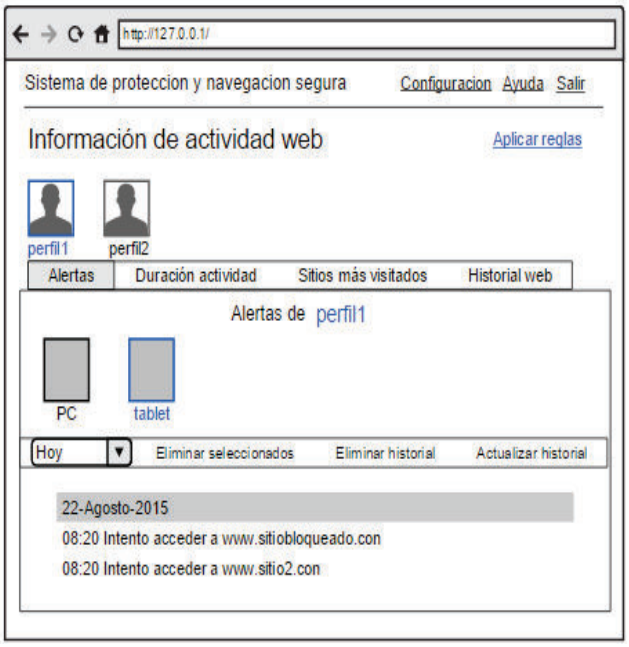

Figura 2.9 Diseño de interfaz Información de navegación web opción Alertas

Las interfaces para visualizar los registros de los sitios web más visitados, historial web y tiempo de navegación web son similares al diseño que se muestra en la Figura 2.9.

Figura 2.10 Diseño de interfaz Permitir tráfico de servicios de Internet

2.2.3.1.4 Diagrama de clases

En la Figura 2.11 se muestra el diagrama de clases de la aplicación web.

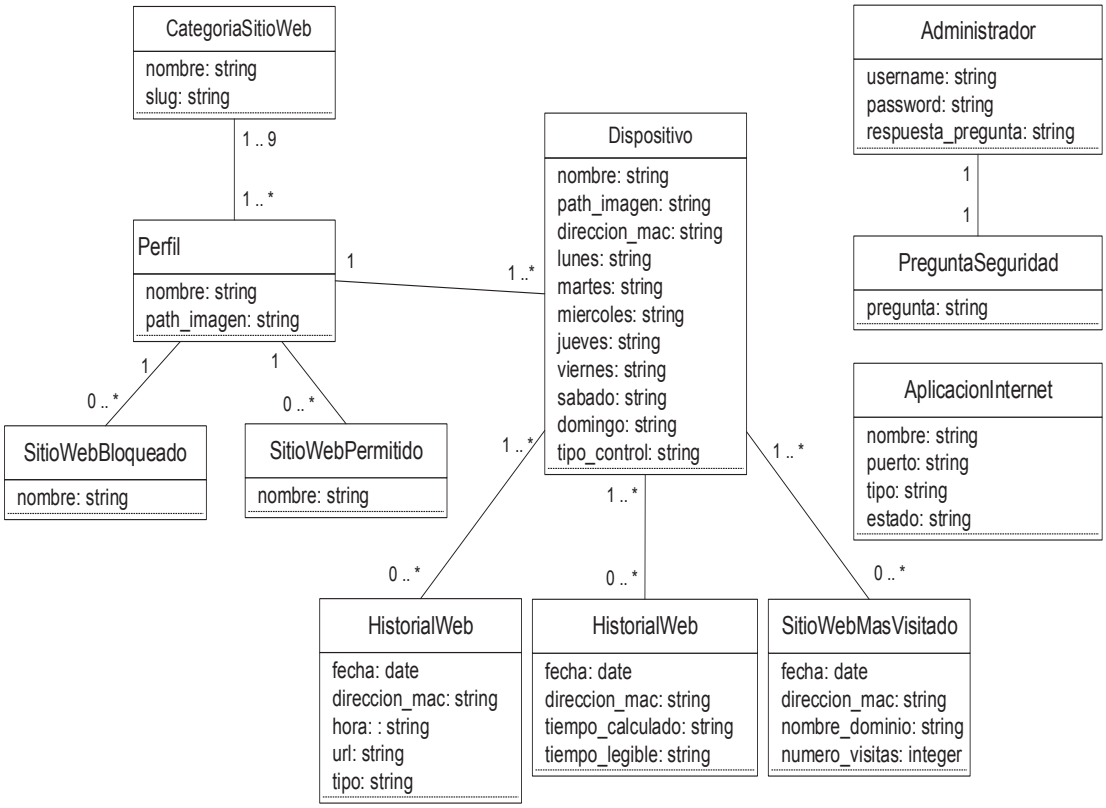

Figura 2.11 Diagrama de clases de la aplicación web

2.2.3.2 Modelo de base de datos

En la Figura 2.12 se muestra el modelo de base de datos que será implementado para almacenar la información sobre la navegación web de los usuarios de la red interna, información para el funcionamiento de la aplicación web, e información base para la configuración de los servicios de filtrado de tráfico.

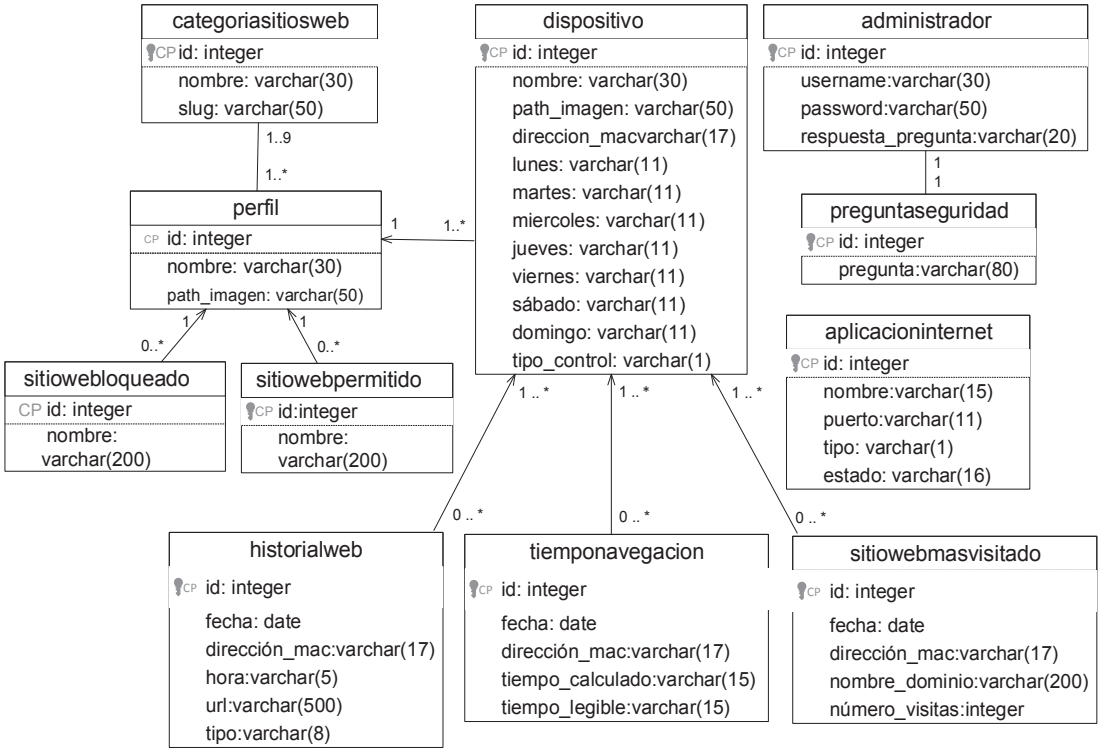

Figura 2.12 Diagrama entidad-relación del modelo de base de datos

A continuación, se describe brevemente las tablas del modelo de datos para identificar su función:

- `categoriasitioweb`: Almacena el nombre que será presentado en la interfaz web y el nombre sin caracteres especiales (slug), utilizado para procesamiento interno, de las categorías de sitios web.
- `perfil`: Almacena el nombre y el *path* de una imagen avatar (niño, niña) de los perfiles creados.
- `dispositivo`: Almacena el nombre, *path* de una imagen avatar (laptop, tablet-celular), dirección MAC y los horarios de acceso permitido para cada día de la semana, y el tipo de control (permitido, bloqueado, horario) para cada dispositivo registrado.

- `sitiowebbloqueado`: Almacena el nombre de dominio o nombre de host de un sitio web que ha sido bloqueado el acceso.
- `sitiowebpermitido`: Almacena el nombre de dominio o nombre de host de un sitio web con acceso permitido.
- `historialweb`: Almacena la fecha y dirección MAC del origen de la información de navegación web recolectada, la hora y URL accedida o bloqueada, cada una es identificada por el atributo tipo.
- `tiemponavegacion`: Almacena la fecha y dirección MAC del origen de la información de navegación web recolectada, el tiempo calculado de navegación web (utilizado para procesamiento de *logs*) y el tiempo legible que será presentado en la interfaz gráfica.
- `sitiowebmasvisitado`: Almacena la fecha y dirección MAC del origen de la información de navegación web recolectada, el nombre de *host* de los sitios web visitados y el número de visitas realizadas.
- `aplicacioninternet`: Almacena el nombre, puerto, tipo (conocida, personalizada) y el estado (habilitada/deshabilitada) de las aplicaciones o servicios de Internet registradas en el prototipo.
- `administrador`: Almacena el nombre de usuario, contraseña y respuesta a la pregunta de seguridad del administrador del sistema.
- `preguntaseguridad`: Almacena preguntas de seguridad para recuperar contraseña.

CAPÍTULO 3

3. IMPLEMENTACIÓN, PRUEBAS Y COSTO DEL PROTOTIPO

En este capítulo se describe la implementación de los requerimientos del prototipo, se documentan las pruebas realizadas para verificar y validar la implementación y se detalla el costo referencial de desarrollo del prototipo.

3.1 PLATAFORMA DEL PROTOTIPO

3.1.1 HARDWARE

El prototipo utiliza como plataforma de hardware un computador Raspberry Pi modelo B+[3] con sistema operativo Raspbian. La utilización de esta plataforma de cómputo fue establecida en el plan de proyecto de titulación. En la Tabla 3.1 se detallan las características de la plataforma.

Características de la plataforma del prototipo (Raspberry Pi, Modelo B+)	
Procesador	700 MHz, 1 núcleo
RAM	483 MB
Memoria SD	16 GB
Sistema Operativo	Raspbian Wheezy
Tarjeta de red Ethernet	Fast Ethernet
Adaptador USB a Ethernet	Fast Ethernet

Tabla 3.1 Características de la plataforma del prototipo

Como se puede observar en la Figura 2.2, el prototipo necesita dos interfaces de red: una interfaz de red para conectarse al *router* doméstico de la red de un hogar y una interfaz de red para conectar la red interna.

El Raspberry Pi modelo B+ cuenta solamente con una interfaz de red, por lo que es necesario utilizar un adaptador USB a Ethernet para conseguir otra interfaz de red.

[3] Raspberry Pi modelo B+: modelo de mayor capacidad que se encontraba disponible cuando inició el desarrollo del presente proyecto de titulación.

3.1.2 SISTEMA OPERATIVO

Existen varias distribuciones Linux para arquitecturas ARM, sin embargo, la distribución oficial de la Fundación Raspberry Pi es Raspbian: sistema operativo basado en Debian y optimizado para el hardware de un Raspberry Pi [32]. Por ello, se utiliza esta distribución. El proceso de instalación y configuración de Raspbian se detallan en el Anexo B.

3.2 IMPLEMENTACIÓN DE COMPONENTES DEL PROTOTIPO

En esta sección se explica la implementación de los componentes que conforman el prototipo. El desarrollo de cada historia de usuario requirió implementar varios componentes del prototipo a la par, e. g., implementar la Historia de usuario No. 6 (Bloquear el acceso a sitios web de categorías) requiere desarrollar la aplicación web con la interfaz Reglas del hogar, crear varias tablas en la base de datos y configurar el servidor proxy HTTP, encargado de aplicar el filtrado web. Sin embargo, para mantener ordenada la documentación se describe la implementación de cada componente por separado.

A continuación se describe la implementación de los siguientes componentes:

- Servidor DHCP
- Filtrado de tráfico: firewall y proxy HTTP
- Recolector de información de navegación web
- Interfaz de administración: aplicación web y base de datos

En la Figura 3.1 se muestra un diagrama de secuencia que refleja, de forma general, la interacción de los componentes involucrados en la configuración del control de acceso web. La interacción de los componentes para configurar el control de acceso a servicios de Internet es similar a la interacción que se ilustra en la Figura 3.1.

3.2.1 SERVIDOR DHCP

Un servidor DHCP (*Dynamic Host Configuration Protocol*) es utilizado en esta solución para asignar automáticamente parámetros de configuración TCP/IP a los *host*s que se conectan a la red interna. Otra función de este componente, es la de mantener registros de arrendamiento de direcciones IP a clientes DHCP. Estos

registros son importantes, porque son utilizados en la implementación de la historia de usuario No. 1 (Registrar dispositivo).

Para implementar el servidor DHCP se utiliza el programa dnsmasq, debido a que es ligero y fácil de configurar, adecuado para redes pequeñas [33].

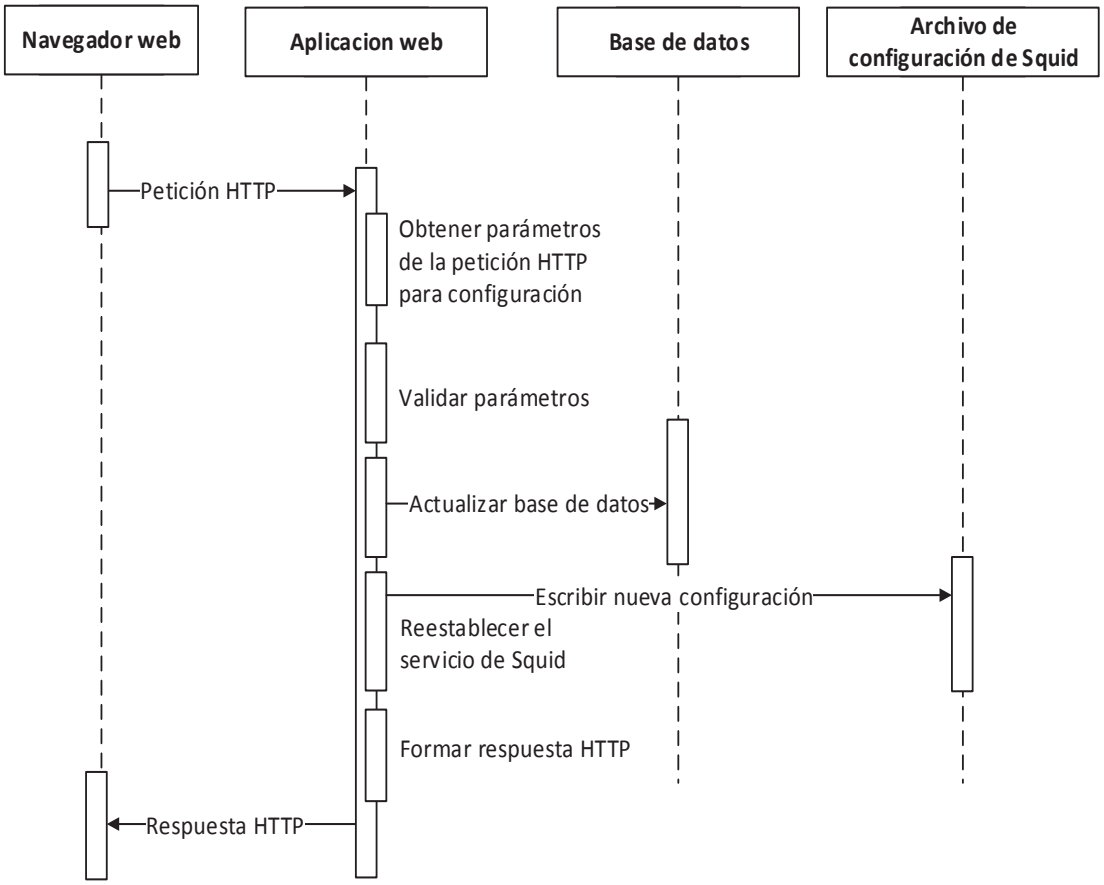

Figura 3.1 Interacción entre componentes del prototipo cuando se configura el filtrado web

3.2.1.1 Configuración del servicio DHCP

Dnsmasq es configurado en el archivo /etc/dnsmasq.conf. En este archivo se añaden las líneas del Código 3.1. La primera línea especifica la interfaz de red (eth0) por la cual dnsmasq escuchará las peticiones DHCP, y la segunda línea establece el rango de direcciones IP a distribuir, así como el período de arrendamiento (*lease time*). La tercera y última línea específica la dirección IP de los servidores DNS que serán enviadas a los clientes. En este caso, se añaden las direcciones IP de los servidores DNS públicos de Google.

```
interface=eth0
dhcp-range=192.168.8.2,192.168.8.254,255.255.255.0,12h
dhcp-option=6,8.8.8.8,8,8,4,4
```

Código 3.1 Configuración del servidor DHCP

3.2.1.2 Base de datos de arrendamiento de clientes DHCP

Dnsmasq almacena en el archivo /var/lib/misc/dnsmasq.leases los registros de las direcciones IP arrendadas a los clientes DHCP, así como sus correspondientes direcciones MAC. Ejemplos de estos registros se presentan en el Código 3.2.

```
1417132679 78:c5:e5:b4:4d:07 192.168.8.153 android-fbe390b0bb404da *
1417134678 20:54:76:e4:d8:a5 192.168.8.125 WINDOWS-PC1 *
```

Código 3.2 Registros de arrendamiento de clientes DHCP

La dirección IP y dirección MAC (ver Código 3.2, color azul) de cada registro de arrendamiento son utilizadas en el desarrollo de la historia de usuario No. 1. Detalles sobre el proceso de registro de un dispositivo son presentados en la implementación de la interfaz de administración.

3.2.2 SERVIDOR PROXY HTTP

El servidor proxy HTTP es implementado con la aplicación Squid. Las funciones de esta aplicación en el prototipo son: implementar el filtrado web (o control de acceso web), registrar información de las transacciones HTTP que procesa, y almacenar en *cache* recursos accedidos con frecuencia. Antes de explicar la configuración de las funciones de Squid, se describe aspectos fundamentales sobre el control de acceso y los *logs* de acceso en Squid.

3.2.2.1 Descripción del control de acceso web en Squid [34]

El control de acceso en Squid es codificado con elementos ACL y listas de acceso. Un elemento ACL es utilizado para definir uno o más valores de un campo de una petición HTTP (e. g., dirección IP origen o destino, nombre de dominio, *path* de la URL); estos valores permitirán identificar la petición HTTP a filtrar. Por otro lado, una lista de acceso está formada por uno o más elementos ACL, y es utilizada para establecer la acción (permitir o denegar) a tomar cuando una petición HTTP es identificada por los elementos ACL.

En Squid, las listas de acceso son verificadas en el orden en el que fueron escritas (en el archivo de configuración), y cuando ocurre una coincidencia se

aplica la acción (permitir o denegar) establecida. Si ninguna lista coincide, la acción por defecto es la opuesta a la de la última lista de acceso verificada. Así que, se recomienda configurar explícitamente una lista de acceso con la acción por defecto.

3.2.2.1.1 Definición de un elemento ACL

La sintaxis para definir un elemento ACL se encuentra en el Código 3.3. Se utiliza la directiva `acl`, un nombre para identificar el elemento ACL, el tipo (e. g., `dst`) de elemento ACL, y el valor del campo de la petición HTTP. También, en lugar definir el valor, se puede especificar el *path* del archivo que contiene el valor o valores del campo de la petición HTTP.

```
acl [nombre_ACL] dst [valor_campo]
```

Código 3.3 Sintaxis de un elemento ACL

Los tipos de elementos ACL que se utilizan para el control de acceso web en esta solución son los siguientes:

- `arp`: dirección MAC del cliente.
- `dst`: dirección IP del destino de una petición web.
- `scr`: dirección IP origen de una petición web.
- `dstdomain`: nombre de dominio del servidor de destino.
- `dstdomain_regex`: expresión regular a ser comparada en el nombre de dominio del servidor de destino.
- `time`: utilizada para especificar el horario de un día de la semana.

3.2.2.1.2 Definición de una lista de acceso

La sintaxis para definir una lista de acceso se muestra en el Código 3.4. Se utiliza la directiva `http_access`, una palabra clave (*allow* o *deny*) y el nombre o nombres de los elementos ACL. Para verificar una lista de acceso http_access con múltiples elementos ACL, Squid aplica una operación lógica AND. En otras palabras, una petición HTTP debe ser identificada por todos los elementos ACL de la lista de acceso.

```
http_access <allow|deny> [nombre_ACL1] [nombre_ACL2] … [nombre_ACLn]
```

Código 3.4 Sintaxis de una lista de acceso

3.2.2.1.3 Páginas web de error personalizadas (deny_info)

Por defecto Squid mantiene un directorio con páginas web de error, utilizadas para informar cuando se produce una denegación de acceso o errores en la transacción HTTP. De estas páginas de error, solamente una es utilizada para informar la denegación de acceso, lo cual es un inconveniente si se requiere informar la razón específica por la que se produjo.

Para superar este inconveniente, Squid provee la directiva `deny_info`, con la cual se puede especificar el *path* de la página web de error personalizada que será servida al usuario cuando se deniega una petición HTTP por cumplir con un elemento ACL en particular. Esto significa que se puede informar si la denegación de acceso se debe a que la petición HTTP estaba dirigida a un sitio web bloqueado, porque el horario de acceso no es el permitido, etc.

La sintaxis para utilizar la directiva deny_info se muestra en el Código 3.5. Se utiliza la directiva deny_info, el *path* de la página web almacenada localmente o URL, y el nombre del elemento ACL que deniega la petición HTTP. Squid recuerda el último elemento acl evaluado en la lista de acceso, y verifica si existe una línea con la directiva deny_info relacionada con ese elemento ACL para retornar la página web de error correspondiente.

```
deny_info <path/to/webpage | URL> [nombre_ACL]
```

Código 3.5 Sintaxis de deny_info

3.2.2.2 Descripción de *logs* de acceso de Squid

Squid puede mantener varios archivos de *log* para registrar información sobre su operación (cache.log), objetos del *cache* (store.log), y transacciones HTTP (access.log) [6]. El archivo de interés en la solución planteada es el de *log* de transacciones HTTP o *log* de acceso, debido a que cada una de sus entradas contiene información útil sobre una petición HTTP realizada por el cliente de Squid (en este caso el navegador web utilizado por un usuario de la red interna). Esta información será procesada para extraer la información de navegación web requerida en las historias de usuario.

La cantidad de información que puede ser registrada en un *log* de acceso depende del formato del mismo. En esta solución se utiliza el formato que se

muestra en el Código 3.6. Cada campo de este formato [6] [35] se describe a continuación, además se menciona su aporte en la extracción de información de navegación web.

```
Timestamp MAC_Address Result/Status_Code Method URL Type
```

Código 3.6 Formato de un *log* de acceso de Squid

Timestamp: Almacena el tiempo, utilizando la marca de tiempo Unix[4], cuando inició la transacción HTTP. El valor de este campo es utilizado para la estimación de tiempo de navegación web y obtener la fecha y hora de la petición HTTP.

MAC Address: Almacena la dirección MAC del *host* cliente. Este campo permite identificar el *host* (dispositivo asignado a un perfil) que fue utilizado para realizar una petición HTTP.

Result: Almacena una etiqueta (e. g., TCP_MISS, TCP_DENIED), provista por Squid, que describe la respuesta HTTP servida; las etiquetas permiten identificar si las respuestas servidas son de peticiones HTTP permitidas, denegadas, etc.

Status Code: Almacena el código que identifica el estado de la respuesta HTTP, e. g., 200, 404, 500, etc. El valor de 200 en este campo corresponde a una petición HTTP recibida correctamente, entendida y aceptada (RFC 2616); este valor es utilizado para identificar *logs* con información de páginas web visitadas exitosamente por los usuarios de la red interna.

Method: Almacena el método de la petición HTTP o HTTPS, e. g., POST, GET, CONNECT. Squid no registra cierta información de la URL de peticiones HTTPS procesadas: protocolo (e. g., http://, https://), path (e. g., /index.html), *string* de consulta (e. g., ?a=b&c=d). El protocolo en la URL es necesario para componer un hiperenlace de cada página web o sitio web perteneciente a un reporte de información de navegación que se presenta en la interfaz de administración. Así, mediante el campo Method se identifican *logs* con información de peticiones HTTPS que tienen la URL incompleta y se añade el campo protocolo.

URL: Almacena la URL de la petición HTTP. De este campo se extrae la URL solicitada por el *host* de la red interna.

---

[4] Es el número de segundos que han transcurrido desde Enero 1 de 1970 (*midnight* UTC/GMT) [51]

Type: contiene el valor del tipo de contenido de la respuesta HTTP, e. g., text/html, application/js, image/jpeg. Un navegador web para cargar una página web debe realizar, por lo general, más de una petición HTTP: una petición HTTP para descargar el documento HTML y las demás para descargar los objetos (e. g., imágenes, hojas de estilo, script, etc.) que conforman la página web. De los *logs* de acceso de estas peticiones HTTP, el que corresponde a la petición al documento HTML aporta con los datos suficientes para extraer información requerida sobre la página web visitada; básicamente, porque la petición HTTP al documento HTML es la única que siempre será realizada cuando el navegador web descarga una página web. Así que el campo Type es utilizado para identificar *logs* de peticiones HTTP a contenido HTML.

3.2.2.3 Configuración del control de acceso en Squid

En el capítulo anterior se presentó en la Figura 2.5 un diagrama de flujo que describe el filtrado de peticiones HTTP que debe realizar el servidor proxy para cumplir con las funcionalidades descritas en las historias de usuario. En esta sección, se explica la codificación de ese diagrama de flujo con elementos ACL y listas de acceso de Squid.

Convenciones para nombrar los elementos ACL

El nombre de los elementos ACL para permitir el acceso web por horario a ciertos dispositivos, bloquear el acceso web a ciertos dispositivos, bloquear el acceso a sitios web específicos, y permitir el acceso a sitios web específicos (estos dos últimos tipos de control son aplicados al perfil) está formado por dos elementos, un número y una palabra, e. g., **35**dispositivo, **22**perfil.

El número corresponde al valor del atributo **id** de la tupla, de la tabla Dispositivo ó Perfil, que contiene los datos del dispositivo ó perfil al que se le aplica el control de acceso. La palabra guarda relación con el tipo de control de acceso aplicado.

El nombre de un elemento ACL, formado por estos dos elementos, es único, y permite identificar las líneas del archivo de configuración que deben ser borradas cuando se cambia el tipo de control de acceso.

Por otro lado, los elementos ACL de los demás tipos de control de acceso son nombrados con una o más palabras separadas por un guion bajo, e. g.,

dispos_no_asignados. De igual forma, las palabras utilizadas guardan relación con el tipo de control de acceso aplicado. El control de acceso que utiliza elementos ACL con este tipo de nombre permanece estático en el archivo de configuración de Squid.

3.2.2.3.1 Permitir el acceso a la aplicación web del prototipo

Squid también intermedia las peticiones HTTP dirigidas hacia el servidor web que hospeda la aplicación web del prototipo. El acceso a esta aplicación web debe estar siempre disponible, por lo tanto se configura por defecto una lista de acceso que permita las peticiones HTTP dirigidas a la dirección IP del servidor web.

La configuración de esta lista de acceso puede ser observada en el Código 3.7. Primero se define el elemento acl web_server_addr con el valor de la dirección IP, 192.168.8.1 (de la interfaz de red interna para conectar la red interna). Luego se define la lista de acceso que permite peticiones HTTP dirigidas a la dirección IP de web_server_addr.

```
acl web_server_addr dst 192.168.8.1
http_access allow web_server_addr
```

Código 3.7 Control de acceso al servidor web

3.2.2.3.2 Permitir acceso a dispositivos no asignados a un perfil

Por defecto, los dispositivos registrados en el prototipo y que no están asignados a un perfil tienen acceso web sin restricciones. La lista de acceso para aplicar este control de acceso se encuentra en el Código 3.8. Primero, se define el elemento ACL dispos_no_asignados con el *path* del archivo que almacenará las direcciones MAC de los dispositivos no asignados a un perfil. Después, se encuentra la lista de acceso que permite las peticiones HTTP identificadas por el elemento ACL dispositivos_no_asignados.

```
acl dispos_no_asignados arp '/etc/squid/listas/dispos_no_asignados'
http_access allow dispos_no_asignados
```

Código 3.8 Permitir acceso web a dispositivos no asignados a un perfil

Este código será fijo y estático en el archivo de configuración de Squid. Cuando un dispositivo es asignado a un perfil, su dirección MAC es removida del archivo y se le aplica un control de acceso diferente, dependiendo de la configuración realizada por el administrador.

3.2.2.3.3 Bloquear el acceso web a ciertos dispositivos

El bloqueo de acceso web a ciertos dispositivos está descrito en la historia de usuario No. 11. Este control de acceso puede ser aplicado individualmente a los dispositivos asignados a un perfil.

Un ejemplo del código para configurar este control de acceso se muestra en el Código 3.9. Primero, se define el elemento ACL 35dispositivo con la dirección MAC del dispositivo al que se le aplica el bloqueo. Después se especifica, con la directiva deny_info, la página web de error (_err_bloqueo) a ser servida cuando se deniegue el acceso; con esta página web se informará al usuario que se deniega el acceso web por que el dispositivo utilizado tiene el acceso bloqueado. Finalmente, se define la lista de acceso que deniega peticiones HTTP provenientes del dispositivo con la dirección MAC definida en 35dispositivo.

```
acl 35dispositivo arp a4:17:31:e4:b7:bd
deny_info _err_bloqueo_dispositivo 35dispositivo
http_access deny 35dispositivo
```

Código 3.9 Bloquear acceso a ciertos dispositivos

Este código es borrado cuando se permite el acceso sin restricción de horario, se aplica el control de acceso por horario o en el caso de que el registro del dispositivo sea eliminado.

3.2.2.3.4 Permitir el acceso web por horario a ciertos dispositivos

El permitir el acceso web por horario a ciertos dispositivos está descrito en la historia de usuario No. 10. Este control de acceso, al igual que en el anterior control, es aplicado individualmente a los dispositivos asignados a un perfil.

En el Código 3.10 se muestra un ejemplo de la configuración para permitir el acceso web por horario. Primero se define el elemento ACL 35dispositivo con la dirección MAC del dispositivo al que se le aplica el control. A continuación, se define el elemento ACL 35horario con el horario permitido por el administrador para cada día de la semana. Después se especifica, con la directiva deny_info, la página web de error (_err_horario) a ser servida cuando se deniegue el acceso; con esta página web se informará al usuario que se deniega el acceso web porque no el horario no es el permitido. Finalmente, se define la lista de acceso que deniega peticiones HTTP provenientes de dispositivos con la dirección MAC

definida en el 35dispositivo y cuyo tiempo no está en el horario definido en 35horario, para especificar esto se utiliza el carácter **!** que significa “no”.

```
acl 35dispositivo arp a4:17:31:e4:b7:bd
acl 35horario time M 14:00-16:00
acl 35horario time T 14:00-16:00
acl 35horario time W 14:00-16:00
acl 35horario time H 14:00-16:00
acl 35horario time F 14:00-16:00
acl 35horario time A 14:00-16:00
acl 35horario time S 14:00-16:00
deny_info _err_horario 35horario
http_access deny 35dispositivo !35horario
```

Código 3.10 Ejemplo de control de acceso por horario

Este código es borrado cuando se bloquea el acceso web, cuando se permite el acceso sin restricción de horario o en el caso de que el registro del dispositivo sea eliminado.

3.2.2.3.5 Permitir el acceso a sitios web específicos

El permitir el acceso a sitios web específicos está descrito en la historia de usuario No. 8. El administrador debe aplicar este control de acceso al perfil, pero internamente se configura el control para aplicarlo a los dispositivos asignados al perfil.

Un ejemplo del código para este control puede ser observado en el Código 3.11. Primero, se define el elemento acl 22perfil con el *path* del archivo que contiene las direcciones MAC de los dispositivos asignados al perfil. Después, se define el elemento ACL 22sitiospermitidos con el *path* del archivo que contiene los nombres de los sitios web permitidos. Finalmente, se define la lista de acceso para permitir peticiones HTTP provenientes de los dispositivos con direcciones MAC definidas en 22perfil y con destino a uno de los sitios web definidos en 22sitiospermitidos. Este código es borrado cuando se elimina un perfil. Si se elimina el registro de un sitio web permitido, el nombre del mismo es borrado del archivo correspondiente.

3.2.2.3.6 Bloquear el acceso a sitios web específicos

El bloqueo de acceso a sitios web específicos está descrito en la historia de usuario No. 7. El procedimiento para realizar este control es bastante similar al del control anterior.

```
acl 22perfil arp "/etc/squid/listas/22perfil
acl 22sitiospermitidos dstdom_regex "/etc/squid/listas/22sitiospermitidos"
http_access allow 22sitiospermitidos 22perfil
```

Código 3.11 Permitir el acceso a sitios web específicos

Un ejemplo del código para este control puede ser observado en el Código 3.12. Para este control se utiliza el elemento ACL 22perfil del Código 3.11. Luego se define el elemento ACL 22sitiosbloqueados con el *path* del archivos que contiene los nombres de los sitios web bloqueados. Finalmente, se define la lista de acceso para bloquear peticiones HTTP provenientes de los dispositivos con direcciones MAC están definidas en 22perfil, y con destino a uno de los sitios web definidos en 22sitiosbloqueados.

```
acl 22sitiosbloqueados dstdom_regex "/etc/squid/listas/22sitiosbloqueados"
http_access deny 22perfil 22sitiosbloqueados
```

Código 3.12 Bloquear sitios web específicos

Este código es borrado cuando se elimina un perfil. Si se elimina el registro de un sitio web bloqueado, el nombre del mismo es borrado del archivo correspondiente.

3.2.2.3.7 Bloquear el acceso a sitios web pertenecientes a categorías

El bloqueo del acceso a sitios web pertenecientes a ciertas categorías está descrito en la historia de usuario No. 6. Este control de acceso también es aplicado un perfil. Para configurar este control de acceso para cada categoría se define un elemento ACL con los nombres de los sitios web de la categoría, otro elemento ACL con las direcciones MAC de los dispositivos asignados a los perfiles que se les ha bloqueado el acceso a los sitios web de la categoría, y la lista de acceso respectiva.

En el Código 3.13 se presenta la configuración para bloquear el acceso a sitios web de la categoría Armas. En la primera línea se define el elemento ACL armas_sitiosweb con el *path* del archivo que contiene los nombres de los sitios web de la categoría Armas. En la segunda línea se define el elemento ACL armas_dispositivos_controlados con el *path* del archivo que contiene las direcciones MAC de los dispositivos asignados a los perfiles que se les ha bloqueado el acceso a los sitios web de la categoría Armas. En la tercera línea se especifica la página web de error (_err_armas) a ser servida cuando se deniegue el acceso; con esta página web de error se informará al usuario que se deniega el

acceso web porque se está solicitando una página web que pertenece a uno de los sitios web de categoría Armas. Finalmente, en la cuarta línea se encuentra la lista de acceso que deniega peticiones HTTP provenientes de los dispositivos con direcciones MAC definidas en armas_dispositivos_controlados, y que están dirigidas sitios web definidos en armas_sitiosweb.

```
acl armas_sitiosweb dstdomain "/etc/squid/listas/armas/dominios"
acl armas_dispositivos_controlados arp  \
    "/etc/squid/listas/_armas_dispositivos_controlados"
deny_info _err_armas armas_sitiosweb
```

Código 3.13 Elementos acl para el control de acceso a sitios web de Armas

Para configurar el control de acceso a los sitios web de las demás categorías se utiliza un código muy similar al presentado en el Código 3.13. El código para este control de acceso permanece estático en el archivo de configuración de Squid, lo que cambia es el contenido de los archivos que contienen las direcciones MAC de los dispositivos asignados a los perfiles que se les bloquea el acceso.

3.2.2.3.8 Listas de acceso por defecto

Si una petición HTTP proveniente de un dispositivo asignado a algún perfil no es denegada por las listas de acceso que bloquean el acceso a sitios web específicos y pertenecientes a categorías bloqueadas, debe ser permitida explícitamente. La lista de acceso para este control se muestra en el Código 3.14. Primero se define el elemento ACL dispositivos_asignados con el *path* del archivo que contiene las direcciones MAC de dispositivos asignados a algún perfil. Luego, se define la lista de acceso para permitir peticiones HTTP que provengan de dispositivos con las direcciones definidas en dispositivos_asignados.

```
acl dispositivos_asignados arp "/etc/squid/listas/dispositivos_asignados'
http_access allow dispositivos_asignados
```

Código 3.14 Control de acceso por defecto a dispositivos asignados a un perfil

La última lista de acceso (ver Código 3.15) que se verifica es la que deniega peticiones HTTP que no han sido identificadas en las listas de acceso definidas anteriormente (lista de acceso para controlar el acceso de los dispositivos registrados en el sistema). En otras palabras, esta última lista de acceso prohíbe el acceso web a dispositivos no registrados en el sistema del prototipo.

En el Código 3.15, primero se define el elemento ACL todo con el valor 0.0.0.0/0 (estos caracteres son utilizados para representar todas las direcciones IP),

después se especifica la página de error a ser enviada cuando se deniegue el acceso web, finalmente se define la lista de acceso que deniega peticiones HTTP identificadas por el elemento ACL todo.

```
acl todo src 0.0.0.0/0
deny_info _err_dispos_no_registrados todo
http_access deny todo
```

Código 3.15 Bloquear el acceso a dispositivos no registrados

3.2.2.4 Configuración de *log* de acceso de Squid

En el Código 3.16 se encuentra la configuración de *log* de acceso. Con la directiva `access_log` se especifica el *path* del archivo que almacenamiento de *log*. Así mismo, con la directiva `logformat` se especifica el formato del *log* de acceso.

```
access_log /var/log/squid/access.log squid
logformat squid %ts.%03tu %>eui %Ss/%03>Hs %rm %ru %mt
```

Código 3.16 Configuración de *log* de acceso de Squid

3.2.2.5 Configuración del almacenamiento en *cache*

Se configura almacenamiento en *cache* (ver Código 3.17*)* para almacenar y reutilizar páginas web solicitadas con frecuencia, con el propósito de mejorar los tiempos de respuesta de futuras solicitudes al mismo contenido.

```
cache_dir aufs /var/spool/squid 100 16 256
maximum_object_size 8 MB
cache_swap_low 90
cache_replacement_policy heap LFUDA
```

Código 3.17 Almacenamiento en *cache*

A continuación se describe las líneas del Código 3.17. La primera línea configura con la directiva `cache_dir` el formato aufs para crear en el directorio /var/spool/squid un cache de 100 MB, dividido en jerarquías de 16 directorios subordinados, con hasta 256 niveles cada uno.

En la segunda línea, con la directiva `maximun_object_size`, se configura el tamaño máximo, 8 MB, de los objetos almacenados. Es recomendado establecer un nivel máximo para evitar que se almacenen objetos de gran tamaño que seguramente serán aprovechados solamente por unos pocos usuarios.

La directiva `cache_swap_low` es utilizada en la tercera línea para configurar la limpieza automática de *cache* cuando se alcance el 90% de su capacidad.

Finalmente, en la cuarta y última línea, se utiliza la directiva `cache_replament_policy` para configurar el algoritmo encargado de reemplazar los recursos no utilizados. El algoritmo LFUDA (*Least Frequently Used with Dynamic Aging),* es recomendado por demostrar un buen desempeño en escenarios de alta carga de trabajo [36].

3.2.2.6 Clasificación de sitios web y descripción de pruebas realizadas

Las categorías predefinidas en el sistema del prototipo corresponden a las categorías de sitios web con un alto nivel de preocupación de acuerdo a los resultados de la encuesta.

Existen listas "negras" de nombres de *host*, nombres de dominios, URL y expresiones regulares de varias categorías de sitios web, que pueden ser descargadas de forma gratuita o mediante un pago. Algunas de estas listas negras fueron probadas, como las de [48] (pagada, pero la primera descarga es gratis), [49] y [39]. Las listas descargadas de nombres de sitios web que corresponden a la clasificación de categorías de esta solución estuvieron formadas por un gran número (hasta miles) de entradas. Al reiniciar el servicio de Squid para cargar estas listas, el tiempo que fue necesario para completar este proceso fue de alrededor de 5 minutos, durante el cual Squid no procesó transacciones HTTP. Además, una vez restablecido el servicio de Squid, el tiempo de carga de páginas web de antes de cargar las listas negras se incrementó.

Squid carga toda su configuración en la memoria principal, así que la demora en reiniciar su servicio es producida por la limitada capacidad de memoria del Raspberry Pi utilizado.

Cabe resaltar que también se realizaron pruebas filtrando expresiones regulares (palabras relacionadas con las categorías) y nombres de *host* en conjunto. El resultado fue un gran número de falsos positivos. Por eso, el filtrado web por destino aplicado, finalmente es únicamente basado en nombre de *host*.

Otra desventaja de las listas negras disponibles que se probó es que gran porcentaje de las URLs que incluían ya no eran válidas. Por eso se buscó una clasificación fidedigna, y la cual corresponde a la clasificación de sitios web más visitados por categoría del sitio web de Alexa Inc. [40].

Para las pruebas de funcionamiento, y por las razones expuestas previamente, se utilizó listas negras pequeñas, de 10 a 15 nombres de sitios web, formadas de la clasificación de los sitios web más visitados por categoría de Alexa Inc.

3.2.3 FIREWALL

El firewall se implementó en el prototipo usando el programa iptables. Con este programa se añaden reglas en las tablas de filtrado de paquetes del *kernel* de Raspbian para implementar la historia de usuario No.18 y No.19. Además, con iptables se configura NAT para que, cuando sea necesaria una conexión directa, los *hosts* de la red interna utilicen la dirección IP de la interfaz de red externa del prototipo y puedan comunicarse con el router doméstico para acceder a Internet.

Un resumen del manual de usuario de iptables se encuentra en el Anexo H.

3.2.3.1 Reenvío de paquetes IP

Como el prototipo es el *gateway* de la red interna (ver Figura 2.2) para acceder a Internet, el sistema operativo del prototipo debe tener la capacidad de enrutamiento de paquetes IP: desde la interfaz de red interna hacia la interfaz de red externa. Esto será necesario cuando los usuarios requieran acceder en Internet a servicios diferentes de web (e. g., correo electrónico).

Por defecto, en el *kernel* del sistema operativo Raspbian, el reenvío de paquetes IP está deshabilitado. Para habilitarlo, en el archivo /etc/sysctl.conf se reemplaza `0` con `1` en la línea `net.ipv4.ip_forward` (ver Código 3.18).

```
net.ipv4.ip_forward = 0
```

Código 3.18 Parámetro para habilitar el reenvío IP

Para activar el cambio del archivo sysctl.conf se ejecuta el comando del Código 3.19.

```
#sysctl -p /etc/sysctl.conf
```

Código 3.19 Activar el reenvío IP

3.2.3.2 Configuración del firewall para filtrado de paquetes

En la configuración del firewall se encuentran las reglas para establecer la política de filtrado por defecto, permitir paquetes de servicios de red que serán provistos

por el prototipo a la red interna, y permitir el reenvío de paquetes de servicios de Internet. Las reglas para el reenvío de paquetes son utilizadas para implementar la historia de usuario No. 18 y No. 19.

Los comandos de iptables para añadir las reglas para filtrado de paquetes son escritos en un *script*, el cual es modificado y ejecutado por la aplicación web. La modificación constituye la adición o eliminación de reglas de reenvío de paquetes.

A continuación se describe el contenido del *script* con comandos de iptables.

3.2.3.2.1 Variables

Como ayuda, se definen variables para almacenar el valor de la interfaz de red interna (INTERNAL_IFACE) y externa (EXTERNAL_IFACE) del prototipo, número de puerto (WEB_SERVER_PORT) y dirección IP del servidor web (WEB_SERVER_IP), y número de puerto del servidor proxy HTTP (PROXY_PORT). Esta definición de variables puede ser observada en el Código 3.20.

```
INTERNAL_IFACE='eth0'
EXTERNAL_IFACE='eth1'
WEB_SERVER_IP='192.168.8.1'
WEB_SERVER_PORT='80'
PROXY_PORT='3128'
```

Código 3.20 Definición de variables

3.2.3.2.2 Eliminación de reglas de las tablas filter y nat

El *script* con comandos de iptables es actualizado y ejecutado cada vez que se cambia la configuración del firewall. Para evitar añadir nuevamente las mismas reglas se eliminan todas las reglas previamente configuradas con los comandos del Código 3.1. Se eliminan las reglas de las tablas filter y nat.

```
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
```

Código 3.21 Limpieza de cadenas de la tabla filter y nat

3.2.3.2.3 Política de filtrado de paquetes

Como ya fue mencionado en el capítulo anterior, la política de filtrado de paquetes es: lo que no está explicitamente permitido, está prohibido. Los comandos de

iptables para configurar esta política de filtrado se muestran en el Código 3.22. Las dos primeras reglas descartan todos los paquetes entrantes y salientes en todas las interfaces de red del prototipo. La tercera regla deniega el reenvío de paquetes por defecto.

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Código 3.22 Política de filtrado de paquetes por defecto

3.2.3.2.4 Permitir paquetes de servicios provistos por el prototipo

El prototipo provee a la red interna los servicios: web, utilizado para acceder a la aplicación web del prototipo; proxy HTTP, encargado de controlar el acceso web de los usuarios de la red interna; y DHCP, para proveer automáticamente la configuración TCP/IP a los *host* conectados a la red interna.

Estos servicios siempre deben estar siempre disponibles en la red interna. Es decir, se debe permitir paquetes enviados y recibidos por estos servicios en la interfaz de red interna del prototipo. En el Código 3.23 se muestran las reglas para este filtrado de paquetes. Los paquetes de los servicios web, proxy y DHCP son identificados por el número puerto TCP y UDP que es utilizado.

```
# DHCP
iptables -A INPUT -i $INTERNAL_IFACE -p udp --dport 67 -j ACCEPT
iptables -A OUTPUT -o $INTERNAL_IFACE -p udp --sport 67 -j ACCEPT
# Proxy HTTP
iptables -A INPUT -i $INTERNAL_IFACE -p tcp --dport $PROXY_PORT \
         -j ACCEPT
iptables -A OUTPUT -o $INTERNAL_IFACE -p tcp --sport $PROXY_PORT \
         -j ACCEPT
# Servidor web
iptables -A INPUT -i $INTERNAL_IFACE -p tcp --dport $WEB_SERVER_PORT \
         -j ACCEPT
iptables -A OUTPUT -o $INTERNAL_IFACE -p tcp --sport $WEB_SERVER_PORT
         -j ACCEPT
iptables -A OUTPUT -p tcp --sport $WEB_SERVER_PORT -d $WEB_SERVER_IP \
         -j ACCEPT
iptables -A INPUT -p tcp --dport $WEB_SERVER_PORT -s $WEB_SERVER_IP \
         -j ACCEPT
iptables -A OUTPUT -p tcp --dport $WEB_SERVER_PORT -s $WEB_SERVER_IP \
         -j ACCEPT
iptables -A INPUT -p tcp --sport $WEB_SERVER_PORT -d $WEB_SERVER_IP \
         -j ACCEPT
```

Código 3.23 Reglas que permiten servicios de red provistos por el prototipo

3.2.3.2.5 Permitir paquetes de servicios de Internet

Permitir paquetes de servicios de Internet registrados por el administrador está descrito en la historia de usuario No. 18 y No 19. Este filtrado de tráfico constituye aceptar el reenvío de paquetes de los servicios permitidos desde la interfaz de red interna hacia la red externa del prototipo. En el Código 3.24 se muestra, como ejemplo, las reglas para aceptar el reenvío de paquetes de SSH (*Secure SHell*).

Además, se observa un comentario de línea al final de cada regla, el cual permite identificar las reglas que deben ser borradas del *script* de configuración cuando el registro del servicio de Internet de la interfaz de administración sea eliminado o deshabilitado.

```
iptables -A FORWARD -i $EXTERNAL_IFACE -o $INTERNAL_IFACE -p \
     tcp --sport 22 -j ACCEPT #regla-usuario
iptables -A FORWARD -i $INTERNAL_IFACE -o $EXTERNAL_IFACE -p \
     tcp --dport 22 -j ACCEPT #regla-usuario
```

Código 3.24 Reglas para aceptar el reenvío de paquetes de servicios de Internet

3.2.3.2.6 Permitir paquetes del servidor proxy HTTP

El servidor proxy HTTP actúa en representación de la red interna para navegar en la Web, por lo tanto, las peticiones web del servidor proxy serán enviadas por la interfaz de red externa. En el Código 3.25 se muestran las reglas para permitir el envío y recepción de paquetes de peticiones HTTP y HTTPS en la interfaz de red externa.

```
iptables -A INPUT -i $EXTERNAL_IFACE -p tcp --sport 80 -j ACCEPT
iptables -A OUTPUT -o $EXTERNAL_IFACE -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -i $EXTERNAL_IFACE -p tcp --sport 443 -j ACCEPT
iptables -A OUTPUT -o $EXTERNAL_IFACE -p tcp --dport 443 -j ACCEPT
```

Código 3.25 Reglas para permitir paquetes del servidor proxy HTTP dirigidos a Internet

3.2.3.2.7 Permitir paquetes de red del servicio de DNS

Paquetes del servicio de DNS se permiten por defecto. Las reglas para permitir este servicio se encuentra en el Código 3.26. Se añaden reglas para aceptar el envío y recepción de paquetes de DNS en la interfaz de red externa, y aceptar el reenvío de paquetes de DNS desde la interfaz de red interna hacia la interfaz de red externa.

```
iptables -A INPUT -i $EXTERNAL_IFACE -p tcp --sport 53 -j ACCEPT
iptables -A OUTPUT -o $EXTERNAL_IFACE -p tcp --dport 53 -j ACCEPT
iptables -A INPUT -i $EXTERNAL_IFACE -p udp --sport 53 -j ACCEPT
iptables -A OUTPUT -o $EXTERNAL_IFACE -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -i $EXTERNAL_IFACE -o $INTERNAL_IFACE -p \
         tcp --sport 53 -j ACCEPT
iptables -A FORWARD -i $INTERNAL_IFACE -o $EXTERNAL_IFACE -p \
         tcp --dport 53 -j ACCEPT
iptables -A FORWARD -i $EXTERNAL_IFACE -o $INTERNAL_IFACE -p \
         udp --sport 53 -j ACCEPT
iptables -A FORWARD -i $INTERNAL_IFACE -o $EXTERNAL_IFACE -p \
         udp --dport 53 -j ACCEPT
```

Código 3.26 Reglas para aceptar paquetes de DNS

3.2.3.3 Configuración de NAT

El comando de iptables para configurar NAT se encuentra en el Código 3.27.

```
iptables -t nat -A POSTROUTING -o $EXTERNAL_IFACE -j MASQUERADE
```

Código 3.27 Configuración de NAT

3.2.3.4 Ejecución del *script* con comandos de iptables

El *script* con comandos de iptables es ejecutado por la aplicación web cada vez que el administrador cambia la configuración del firewall y por Raspbian en su inicio de ejecución para cargar la configuración. La programación de la ejecución del *script* cuando inicia Raspbian, se realiza añadiendo la línea del Código 3.28 en el archivo /etc/rc.local.

```
/bin/bash /home/project/iptables.sh
```

Código 3.28 Programación de ejecución del script con comandos de iptables

3.2.4 RECOLECTOR DE INFORMACIÓN DE NAVEGACIÓN WEB

Este componente es implementado con un programa codificado en Python. El programa es ejecutado periódicamente para procesar en segundo plano el archivo *de log de acceso* de Squid y obtener información diaria sobre el tiempo de navegación web, las páginas web visitadas, los sitios web más visitados y los intentos bloqueados de acceso web.

3.2.4.1 Obtención de información de navegación web

A continuación se describe el método heurístico utilizado para extraer cada tipo de información de navegación web.

3.2.4.1.1 Tiempo de navegación web

La navegación web de un usuario puede considerar esencialmente 2 etapas: una etapa en la que genera peticiones HTTP para acceder a los recursos web, y otra etapa en la que el usuario observa la información servida (texto, imágenes) y que, teóricamente, sería navegación también. Estrictamente hablando, el tiempo de navegación web debería considerar estas dos etapas; sin embargo, desde Squid no es posible detectar la duración de la segunda etapa (pasiva) ya que no se generan eventos explícitos de navegación (peticiones HTTP). Por esa razón, el tiempo de navegación se calcula solamente a partir de los eventos explícitos de acceso generados por el usuario y aquellos derivados hacia entidades de terceros.

Para calcular este tiempo de navegación, se va sumando el tiempo transcurrido entre dos eventos consecutivos de navegación registrados en el archivo de *log* de acceso de Squid y que corresponden a las peticiones HTTP explícitas del usuario y sus derivadas (e. g., peticiones para el despliegue de anuncios). Finalmente, si el tiempo transcurrido entre dos eventos es mayor a 20 segundos, este tiempo no se utiliza para el cálculo, pues este período se considera como tiempo de inactividad web. Este tiempo de referencia (20 segundos) es el tiempo máximo que se observó en varias pruebas realizadas para la carga de un recurso web junto con todos sus componentes.

3.2.4.1.2 Páginas web visitadas

Las páginas web visitadas se obtienen de *logs* de acceso que tienen el valor `200` en el campo Status Code y el valor `type/HTML` en el campo Type. Si una página web es visitada más de una vez en un día, se registra la última visita, que se detecta en base a la marca de tiempo del campo Timestamp.

3.2.4.1.3 Sitios web más visitados

De *logs* de acceso que tienen el valor `200` en el campo Status Code y el valor `type/HTML` en el campo Type, se obtiene el nombre de *host* de destino del campo URL y se cuenta el número de veces que aparece una misma URL para determinar el número de visitas realizadas.

3.2.4.1.4 Intentos de acceso bloqueados

Los intentos de acceso bloqueados se obtienen de *logs* de acceso que tienen el valor `200` en el campo Status Code, el valor `type/HTML` en el campo Type y la etiqueta `TCP_DENIED` (indica peticiones HTTP denegadas por Squid). Al igual que en las páginas web visitadas se registra el último intento de acceso web a una misma URL

Para todos los tipos de información de navegación web, la fecha y hora se obtienen del campo Timestamp y la dirección MAC del dispositivo utilizado del campo MAC Address.

3.2.4.2 Algoritmo del programa recolector

El programa recolector de información de navegación web realiza lo siguiente:

a. Lectura del *archivo de log de acceso*.
b. Almacenamiento temporal de *logs.*
c. Limpieza del archivo de *log* para evitar duplicidad de información en un próximo procesamiento.
d. Consulta de las direcciones MAC de dispositivos asignados a perfil.
e. Cálculo del tiempo de navegación web por cada dirección MAC consultada en el literal d.
f. Filtrado de los registros para obtener *logs* de respuestas HTTP de contenido HTML y almacenamiento del resultado.
g. Identificación de *logs* de transacciones HTTP permitidas y denegadas en los registros almacenados en el literal f y extracción de información sobre las páginas web visitadas e intentos de acceso por cada dirección MAC consultada en el literal d.
h. Obtención de información sobre los sitios web más visitados por cada dirección MAC consultada en el literal d.

Para cada paso relacionado con la extracción de información web se realiza una inserción de nuevos registros o actualización de atributos de registros existentes en la base de datos. Esta última tarea consiste en actualizar el horario de acceso o intento de acceso a una URL al más reciente en un día, sumar el número de visitas calculado al total almacenado, o sumar el tiempo calculado al total de

tiempo de navegación web. El código del programa recolector de información se encuentra en el Anexo C.

3.2.4.3 Programación de la ejecución del programa recolector

La programación de la ejecución periódica del programa se realiza en el archivo de configuración `crontab` del servicio `cron`[5] de Linux (ver Código 3.29); el programa se ejecutará cada minuto.

```
* * * * * /usr/bin/python /home/project/procesar_squid_logs.py
```

Código 3.29 Programación de la ejecución programa recolector

Durante la etapa de pruebas se programó una sola ejecución diaria del programa recolector. Esto no permitió tener información actualizada sobre la navegación web que se realiza en entre dos ejecuciones del programa recolector. Para solucionar este problema, se programó ejecutar el programa recolector cada vez que acceden a la aplicación web. Esto incremento el tiempo de espera para acceder a la aplicación web; este tiempo depende de la cantidad de *logs* de acceso que se procesan, pero en promedio fue de alrededor de 3 minutos.

Para solucionar estos inconvenientes, se programó ejecutar el programa recolector cada minuto. Con esto se evita procesar gran cantidad de *logs*, incrementar el tiempo de espera para acceder a la aplicación web y no tener información de navegación web actualizada.

3.2.5 INTERFAZ DE ADMINISTRACIÓN

La interfaz de administración está conformada por una aplicación web y una base de datos. La aplicación web es desarrollada con el *framework* web Django. Para la base de datos se utiliza SQLite. A continuación se describen características generales de estas herramientas, la instalación y configuración de Django y se del código la interfaz de administración.

3.2.5.1 Lenguaje de programación Python

Python es un lenguaje de programación interpretado *open source* con gran cantidad de librerías, de propósito general y específico. La simplicidad y claridad

[5] Servicio de Linux que permite ejecutar en *background* tareas programadas.

en la sintaxis del lenguaje permiten un desarrollo rápido y eficiente. El ambiente que provee Python es adecuado para el desarrollo de un prototipo inicial [41].

Para el desarrollo de la aplicación web del prototipo se utilizó un *framework web*[6] que utilice Python y con patrón de arquitectura MVC[7], con el propósito de conseguir una estructura efectiva para mantener el código ordenado y que facilite el desarrollo. El *framework web* elegido fue Django, debido a que utiliza Python como lenguaje de programación, maneja el patrón de arquitectura MCV y existe buena documentación.

3.2.5.2 *Framework* web Django

Django es un *framework web open source* con una infraestructura completa que permite un desarrollo rápido y simple. Se encarga de tareas complicadas y repetitivas en el desarrollo web, e. g., conexión con la base de datos, sesión cliente-servidor, autenticación, seguridad, etc. [42]. Otra característica importante de Django es que incluye un servidor web ligero para realizar pruebas durante el desarrollo.

Django promueve el desarrollo independiente de las piezas que conforman una aplicación web. Se puede realizar cambios a una pieza de software en particular sin afectar a las demás.

3.2.5.2.1 Patrón de Arquitectura [43]

El patrón de arquitectura de Django es bastante similar al patrón de arquitectura MVC. La diferencia en Django es que éste mismo realiza las funciones del Controlador (encargado de realizar un enrutamiento de las peticiones HTTP del cliente para que sean procesadas), de tal forma que el patrón de arquitectura cambia a MPV (Modelo-Plantilla-Vista). El Modelo define cómo se accede a los datos, así como la validación y relación entre ellos; la Plantilla es la encargada de la presentación de los datos; y la Vista contiene la lógica de la aplicación que permite relacionar Modelo y Plantilla.

---

[6] Un *framework web* proporciona una infraestructura para desarrollar aplicaciones sin tener que manipular detalles de bajo nivel como protocolos, sockets, administración de procesos, etc. [52]

[7] MVC (Modelo-Vista-Controlador): Es un patrón de arquitectura de software que permite separar datos, la lógica del negocio y la interfaz de usuario de forma independiente para facilitar el desarrollo de aplicaciones.

La Figura 3.2 muestra un diagrama que ilustra el funcionamiento de Django cuando sirve una página web. Las acciones son las siguientes:

1. A través del navegador web se envía una petición HTTP para acceder a la página web.
2. El URLconf realiza el mapeo entre la URL de la petición HTTP y la vista correspondiente.
3. La vista se encarga de procesar la petición HTTP, y de ser necesario manipular el modelo de datos.
4. La vista toma la plantilla para enviarla como respuesta a la petición realizada.
5. La plantilla es presentada en el navegador web.

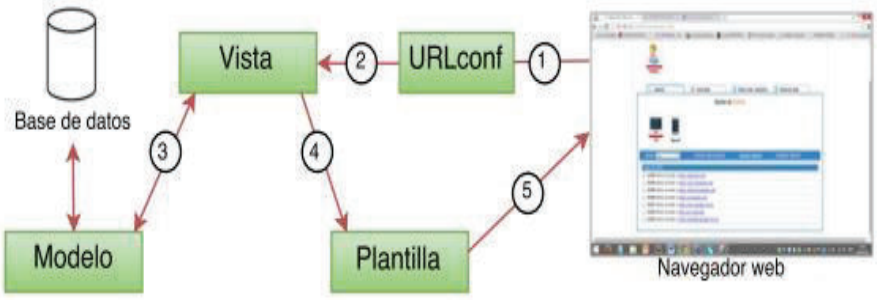

Figura 3.2 Funcionamiento de Django

Conceptos fundamentales que se manejan en Django se describen a continuación.

Modelos

Django define a un modelo como la única fuente para acceder a la información de una tabla de la base de datos. Los modelos son clases que permiten manipular las tablas de la base de datos como objetos; una clase se refiere a una tabla y cada atributo se refiere a un campo.

Plantillas

Son documentos HTML (*HyperText Markup Language*) con variables y etiquetas propias de Django, que pueden ser utilizadas para presentar datos de los modelos. Django utiliza el concepto de herencia de plantillas, el cual es muy útil para estructurar una página HTML mediante bloques (definidos con etiquetas).

Esto significa que de una plantilla se puede heredar una estructura base e ir modificándola de acuerdo a los bloques definidos.

Vistas

Una vista es una función de Python que recibe como argumento una petición HTTP y retorna una respuesta. La respuesta puede ser código HTML (plantillas), redirección a otra vista, etc. En una vista se puede utilizar todas las funcionalidades que Python ofrece, e. g., escritura de archivos, ejecutar programas externos.

URLconf

Para decidir qué vista utilizar en función de la URL de la petición HTTP, Django utiliza un esquema muy ordenado y flexible que permite mapear patrones de URL y el nombre de la vista que se encarga de procesar la petición.

3.2.5.3 Base de datos SQLite

SQLite cuenta con características suficientes para el almacenamiento de información del prototipo, entre las cuales se puede mencionar las siguientes [44] [45]:

- Adecuada para sistemas embebidos o sitios web con baja cantidad de tráfico.
- Transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
- No necesita configuración inicial ni administración.
- Multi-plataforma (funciona en Android, iOS, Linux, Solaris y Windows).
- API fácil de utilizar.
- El código y documentación son parte del dominio público.
- Implementación de casi todas las operaciones con SQL[8]

3.2.5.4 Instalación y configuración de Django

Para desarrollar la aplicación web se utilizó Django 1.7, el cual necesita Python 2.7. En Raspbian ya viene instalado esta versión de Python, por lo que no fue necesario instalar paquetes adicionales. Para instalar Django se utilizó `pip`

[8] SQL (*Structure Query Language*): Lenguaje de alto nivel que permite manipular datos relacionales [53].

(herramienta para instalar paquetes de Python), con el comando del Código 3.30. `pip` ya viene instalado con la versión 2.7 de Python.

```
# pip install django==1.7
```

Código 3.30 Comando para instalar Django

A continuación, se explica la creación de un proyecto y aplicación de Django y la organización del directorio del proyecto para desarrollar la interfaz de administración.

3.2.5.4.1 Creación de un proyecto

Un proyecto en Django es una colección de configuraciones y aplicaciones, que juntos forman un sitio web. Para la creación de un proyecto de Django se ejecuta el comando del Código 3.31. El proyecto para desarrollar la interfaz de administración tiene el nombre `project`.

```
root@sistema:/home/project# django-admin.py startproject project
```

Código 3.31 Comando para crear un proyecto de Django

Como resultado de la ejecución del comando se crea un directorio con el nombre `project` (directorio del proyecto). Este directorio contiene lo siguiente:

- `manage.py`: Utilitario de línea de comandos para realizar tareas administrativas en el proyecto, e. g., crear aplicaciones, ejecutar el servidor web de desarrollo, etc.
- `project`: Directorio con el mismo nombre del directorio del proyecto que contiene varios *script* de Python para la configuración del proyecto. Para continuar con la explicación, este directorio será identificado como el directorio de configuración del proyecto.

El directorio de configuración del proyecto contiene los siguientes *scripts*:

- `__init__.py`: Es un archivo vacío, el cual es necesario para indicar que el directorio que lo contiene debe ser tratado como un paquete de Python.
- `settings.py`: Contiene la configuración del proyecto, e. g., nombres de las aplicaciones instaladas, parámetros para acceder a base de datos, configuración para servir contenido estático, entre otros.

- `urls.py`: Almacena el mapeo entre la URL raíz de cada aplicación y su respectivo archivo urls.py.
- `wsgi.py`: Es un script utilizado como punto de acceso para los servidores web compatibles con WSGI[9] para servir el proyecto.

3.2.5.4.2 Creación de una aplicación

Una aplicación en Django es una aplicación web encargada de realizar una tarea específica. Para la creación de una aplicación se ejecuta el comando del Código 3.32. Para la interfaz de administración, se creó una sola aplicación llamada `prototipo`.

```
root@sistema:/home/project$ python manage.py startapp prototipo
```

Código 3.32 Comando para crear una aplicación de Django

El comando crea un directorio llamado `prototipo` con los siguientes *script* de Python para configurar la funcionalidad de la interfaz de administración:

- `models.py`: Es un *script* para almacenar la definición de los modelos.
- `views.py`: Es un *script* para escribir las funciones que implementan la lógica de las funcionalidades.
- `tests.py`: Es un *script* para escribir pruebas unitarias.

La aplicación `prototipo` debe ser instalada en el proyecto. En INSTALLED_APPS del archivo settings.py se escribe el nombre de aplicación web, tal como se observa en el Código 3.33.

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'prototipo'
)
```

Código 3.33 Registro de la aplicación `prototipo`

En un proyecto de Django, por defecto, ya vienen instaladas varias aplicaciones: sessions, auth, staticfiles, entre otras. Las funciones de estas aplicaciones son

[9] WSGI (*Web Server Gateway Interface*): Especificación estándar de Python que describe el método de comunicación entre un servidor web y una aplicación web.

utilizadas en la aplicación prototipo, para controlar el acceso a la aplicación web, servir archivos estáticos, manejar automáticamente la sesión cliente-servidor.

Así mismo, se debe configurar el acceso a la base de datos que es utilizada, en el archivo settings.py. Por defecto, Django configura el acceso a una base de datos SQLite (ver Código 3.34).

```
import os
BASE_DIR = os.path.dirname(os.path.dirname(__file__))
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Código 3.34 Definición de parámetros de la base de datos SQLite

3.2.5.4.3 Directorio para desarrollo de la interfaz de administración

La estructura del directorio del proyecto para desarrollar la interfaz de administración puede ser observada en la Figura 3.3.

Además de los archivos y directorios descritos previamente, se crearon los siguientes archivos y directorios:

- `prototipo/urls.py`: Es un script de Python que almacena el mapeo entre URLs de la aplicación web y las vistas que las procesan.
- `static`: Es un directorio que almacena: hojas de estilo[10], scripts de JavaScript[11], e imágenes.
- `templates/prototipo`: Es un directorio que almacena las plantillas HTML (interfaces de usuario) de la aplicación web.
- `uwsgi_nginx_conf`: Es un directorio que almacena los archivos de configuración de las aplicaciones `uwsgi` y `nginx` para servir archivos del directorio `static` y la aplicación web del prototipo.
- `procesar_squid_logs.py`: Programa recolector de información de navegación web.
- `iptables.sh`: Es un *script* con comandos de iptables para configurar el firewall del prototipo.

[10] Se utilizan hojas de estilo para dar apariencia a una plantilla HTML, por ejemplo: color, posición de elementos, etc.
[11] Se utilizan programas de JavaScript para validar datos en el lado del cliente y manipular eventos ocurridos en una página web.

- `iptables.sh.inicial`: Es un *script,* con comandos de iptables, utilizado para restablecer la configuración inicial del firewall.

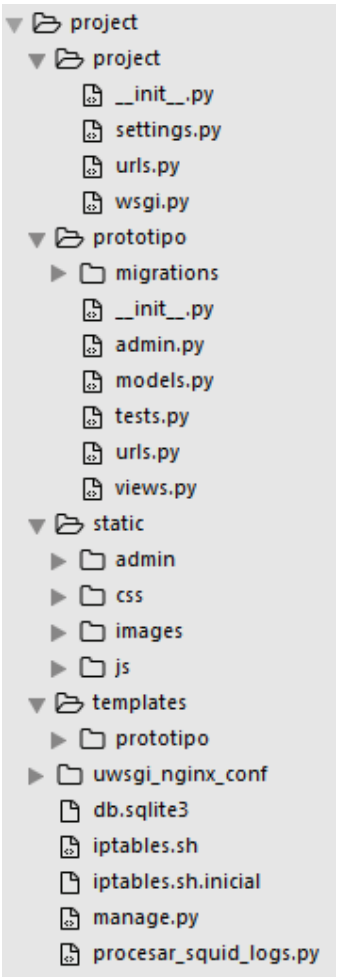

Figura 3.3 Estructura del directorio para desarrollar la interfaz de administración

3.2.5.5 Codificación

3.2.5.5.1 Modelos

En el archivo project/prototipo/models.py, que se incluye en el Anexo D, se encuentran los modelos. Como ejemplo de los modelos, en el Código 3.35 se muestra el modelo `CategoriaSitioWeb`.

En el modelo `CategoriaSitioWeb` cada atributo corresponde a un campo de la tabla en la base de datos. Se define el tipo de campo (`CharField`, `ManyToManyField`, `SlugField`), la longitud máxima de caracteres permitida (`max_length`), si se permite la ausencia de datos (`blank`) o si el campo debe

ser único en la tabla (`unique`). Si no se especifica el campo de clave primaria, Django automáticamente añade un campo del tipo `IntegerField` para la clave primaria.

```
class CategoriaSitioWeb (models.Model):
    perfil=models.ManyToManyField(Perfil,null=True,blank=True,unique=False)
    nombre=models.CharField(max_length=30,unique=False,blank=True)
    slug = models.SlugField(unique=False)
    def save(self, *args, **kwargs):
        self.slug = slugify(self.nombre)
        super(CategoriaSitioWeb , self).save(*args, **kwargs)
```

Código 3.35 Clases de los modelos `Perfil` y `CategoriaSitioWeb`

Django provee los siguientes tipos de campo para definir las relaciones entre modelos/tablas:

- `ForeignKey`: permite crear una relación uno-a-muchos.
- `OneToOneField`: permite definir una estricta relación de uno-a-uno.
- `ManyToManyField`: permite definir una relación de muchos a muchos.

Cada uno de estos tipos de campo requiere como argumento el modelo con el cual se relaciona. Adicionalmente, se pueden definir métodos en cada modelo. Por ejemplo, en el modelo `CategoriaSitioWeb` se define un método para cambiar los espacios de la entrada del atributo slug por guiones.

Creación y actualización de tablas

Para que las tablas representadas en los modelos sean creadas en la base de datos, se debe ejecutar el comando del Código 3.36.

```
root@sistema:/home/project$ python manage.py syndb
```

Código 3.36 Comando para crear las tablas de la base de datos en Django

En el caso de editar los modelos, se debe ejecutar los comandos del Código 3.37, para aplicar los cambios realizados, y que estos se vean reflejados en las tablas de la base de datos.

```
root@sistema:/home/project$ python manage.py makemigrations
root@sistema:/home/project$ python manage.py migrate
```

Código 3.37 Comandos para aplicar cambios en los modelos

3.2.5.5.2 Plantillas

Las plantillas HTML, de cada interfaz de administración (página web), se encuentran en el directorio project/templates/prototipo, que se incluye en el Anexo

D. Se utilizó la herencia de plantillas de Django. Se creó una plantilla base (`base.html`), la cual tiene el encabezado que se repite en todas las páginas web, y un bloque (contenido) que puede ser modificado en las plantillas hijas (el resto de plantillas).

3.2.5.5.3 Vistas

En el Código 3.38 y Código 3.39 se muestra la vista `regla_filtrar_sitios_web`, las demás vistas se encuentran en el archivo project/prototipo/views.py, que se incluye en el Anexo D. Esta vista configura el filtrado de sitios web de categorías, filtrado de sitios web específicos y sirve contenido estático.

```
@login_required
def regla_filtrar_sitios_web(request, id_perfil):
    if request.is_ajax(): #aplicar regla web
        resultado = '0'
        perfil = ''
        id_perfil = request.POST.get('id_perfil','')
        sitios_permitidos = request.POST.getlist('sitios_permitidos[]')
        sitios_bloqueados = request.POST.getlist('sitios_bloqueados[]')
        eliminar_sitios_permitidos =
            request.POST.getlist('eliminar_sitios_permitidos[]','')
        eliminar_sitios_bloqueados =
            request.POST.getlist('eliminar_sitios_bloqueados[]','')
        categorias_bloqueadas =
            request.POST.getlist('categorias_bloqueadas[]','')
        perfil = Perfil.objects.filter(id = id_perfil)
        if perfil:
            perfil = Perfil.objects.get(id = id_perfil)
            establecer_reglas_sitiosweb(SitioWebPermitido,
                 perfil,'sitiospermitidos',sitios_permitidos)
            establecer_reglas_sitiosweb(SitioWebBloqueado,
                 perfil,'sitiosbloqueados',sitios_bloqueados)
            eliminar_reglas_sitiosweb(SitioWebBloqueado,
                 perfil,'sitiosbloqueados',eliminar_sitios_bloqueados)
            eliminar_reglas_sitiosweb(SitioWebPermitido,
            perfil,'sitiospermitidos',eliminar_sitios_permitidos)
            SitioWebPermitido.objects.filter(perfil = None).delete()
            SitioWebBloqueado.objects.filter(perfil = None).delete()
            categorias_a_bloquear = []
             for id_cat in categorias_bloqueadas:
                slug_categoria =
                     CategoriaSitioWeb.objects.get(id = id_cat).slug
                categorias_a_bloquear.append(slug_categoria)
            establecer_reglas_categorias_squidconf(id_perfil,
                categorias_a_bloquear)
            reiniciar_servicio_squid()
            resultado = '1'
        return  HttpResponse(resultado)
```

Código 3.38 Vista `regla_filtrar_sitios_web`

```
# Enviar contenido de la pagina web
else:
        context_dict = {}
        perfil = ''
        perfiles = Perfil.objects.all()

        if perfiles:
            if id_perfil == str(0):
                perfil = perfiles[0]
            else:
                perfil = Perfil.objects.get(id = int(id_perfil))
            context_dict["perfiles"] = perfiles
            context_dict['perfil'] = perfil
            context_dict['categorias02'] =
                   consultar_categorias(perfil)[:3]
            context_dict['categorias35'] =
                   consultar_categorias(perfil)[3:6]
            context_dict['categorias69'] =
                   consultar_categorias(perfil)[6:9]
            context_dict['categorias911'] =
                   consultar_categorias(perfil)[9:]
            context_dict['listablanca'] =
                   SitioWebPermitido.objects.filter(perfil = perfil)
            context_dict['listanegra'] =
                   SitioWebBloqueado.objects.filter(perfil = perfil)
            context_dict['perfiles_sin_dispositivos']=
                   perfiles_sin_dispositivos()
            nombre_dispositivo_asignado =
                   request.session.get('nombre_dispositivo_asignado','')
            nombre_perfil_nuevo_seleccionado =
                request.session.get('nombre_perfil_nuevo_seleccionado','')
            nombre_perfil_seleccionado =
                   request.session.get('nombre_perfil_seleccionado','')
            regla_guardada = request.session.get('regla_guardada','')

return render(request,"prototipo/reglas_filtrado_paginas_web.html",
context_dict)
```

Código 3.39 Vista `regla_filtrar_sitios_web` (continuación)

Una breve descripción del código de la vista `regla_filtrar_sitios_web` se presenta a continuación. Primero, se utiliza la directiva `@login_required` para controlar el acceso a la vista, esta verifica que la petición HTTP provenga de un cliente con acceso concedido (nombre usuario y contraseña válidos). Después, se identifica si la petición HTTP es síncrona o asíncrona.

En el caso de ser la petición HTTP asíncrona[12], se obtiene los datos de la petición. Luego, se verifica si perfil, al que se le aplica la regla, existe. Con un

[12] Peticiones HTTP enviadas en segundo plano. El procesamiento en el servidor toma tiempo, y mientras se espera la respuesta, la aplicación web no puede ser utilizada, se bloquea. Al utilizar peticiones asíncronas no se espera la respuesta del servidor, es decir se puede continuar utilizando la aplicación web mientras se recibe la respuesta.

resultado positivo de la verificación realizada, se continúa a actualizar las tablas de los modelos `CategoriaSitioWeb`, `Perfil`, `SitioWebPermitido` y `SitioWebBloqueado`; después, se añade o eliminan las respectivas listas de acceso en el archivo `squid.conf`. Finalmente, se reinicia el servicio de Squid y se envía una respuesta al cliente.

En el caso contrario, la petición HTTP es síncrona, se consulta la información de los perfiles, las categorías de sitios web bloqueadas y no bloqueadas, sitios web permitidos y sitios web bloqueados para armar el diccionario `context_dict` que será devuelto junto con la plantilla `regla_filtrar_sitios_web.html`, la cual contiene etiquetas especiales para iterar en el diccionario `context_dict` e ir presentando la información consultada.

3.2.5.5.4 URLconf

En el archivo project/prototipo/urls.py, que se incluye en el Anexo D, se encuentra la definición de las URLs correspondientes a las páginas web (interfaces de administración) de la aplicación web y las respectivas vistas que las procesan.

En el Código 3.40 se muestran 2 ejemplos de las entradas del archivo urls.py. Se define la URL y la vista. Se puede utilizar expresiones regulares o definir variables en la URL. Por ejemplo, en la expresión `(?P<id_perfil>[0-9]+)`, se indica el nombre de la variable (`P<id_perfil>`), y el tipo valores (`[0-9]`) que son aceptados.

```
from django.conf.urls import patterns, url
from prototipo import views
urlpatterns = patterns('',
        url(r'^reglas-hogar/filtrar-sitios-web/\
           (?P<id_perfil>[0-9]+)/$',views.regla_filtrar_sitios_web),
        url(r'^reglas-hogar/acceso-por-horario/(?P<id_perfil>[0-9]+)/\
           (?P<id_dispo>[0-9]+)/$',views.regla_acceso_por_horario),
        )
```

Código 3.40 Ejemplos de entradas del archivo urls.py

3.2.5.5.5 Interfaces de usuario

A continuación se presenta el resultado del desarrollo de las interfaces de usuario que conforman la interfaz de administración. Cada una de ellas cuenta con títulos descriptivos y breves descripciones sobre cómo utilizar sus diferentes opciones.

Configuración Inicial

En la Figura 3.4 se muestra la interfaz de configuración inicial del prototipo. Desde esta interfaz se puede acceder a todas las interfaces de administración del sistema. La opción “Dispositivos Registrados” y “Perfiles Creados” despliegan sus respectivos registros. La opción “Cuenta administrador” muestra un formulario con el nombre de usuario, la contraseña, pregunta y respuesta para cambiar la contraseña; toda esta información es editable. La opción “Restablecer configuración inicial” despliega un cuadro de diálogo, en el que se informa sobre las consecuencias de restablecer la configuración y se pide confirmación para realizar la operación. La opción “Avanzado” permite acceder a una nueva interfaz para registrar servicios de Internet.

Figura 3.4 Interfaz Configuración inicial

Registro y asignación de dispositivos a un perfil

Para registrar un dispositivo y asignarlo a un perfil, el administrador debe acceder a la interfaz de administración desde el dispositivo en cuestión y escoger la opción “Dispositivos registrados” de la interfaz de la Figura 3.4. En este proceso la interfaz provee las facilidades para guiar al administrador. Primero, se despliega un cuadro de diálogo (ver Figura 3.5) para ingresar información del dispositivo a

registrar. Después, se despliega inmediatamente un cuadro de diálogo para asignar el dispositivo, recién registrado, a un perfil existente o crear uno nuevo (ver Figura 3.6).

Figura 3.5 Interfaz Registrar dispositivo

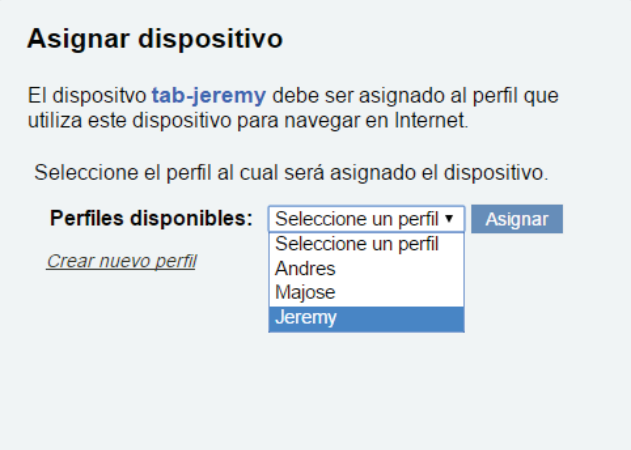

Figura 3.6 Interfaz Asignar dispositivo registrado a un perfil

Para aplicar el control de acceso web con Squid se necesita conocer la dirección MAC de cada dispositivo. Para evitar que el administrador tenga que ingresar directamente este parámetro en el registro del dispositivo, se propuso realizar el procedimiento descrito en el párrafo anterior. Al realizar ese procedimiento, detrás de escena sucede lo siguiente: la aplicación web recibe la petición HTTP para registrar el dispositivo, extrae la dirección IP del cliente, y la utiliza para consultar la dirección MAC del dispositivo a registrar en los registros de arrendamiento de direcciones IP del servidor DHCP; con la dirección MAC consultada y con el nombre ingresado por el administrador se registra el dispositivo.

Aplicación de reglas de control de acceso web

En la Figura 3.7 se muestra la interfaz en donde se puede aplicar la regla para bloquear el acceso a sitios web específicos, bloquear el acceso a sitios web pertenecientes a determinadas categorías e permitir el acceso a un sitio web. Esta última opción es útil cuando se quiere permitir el acceso a un sitio web perteneciente a una categoría con acceso bloqueado. Para aplicar una regla se debe seleccionar el perfil y a continuación se despliegan las opciones antes mencionadas.

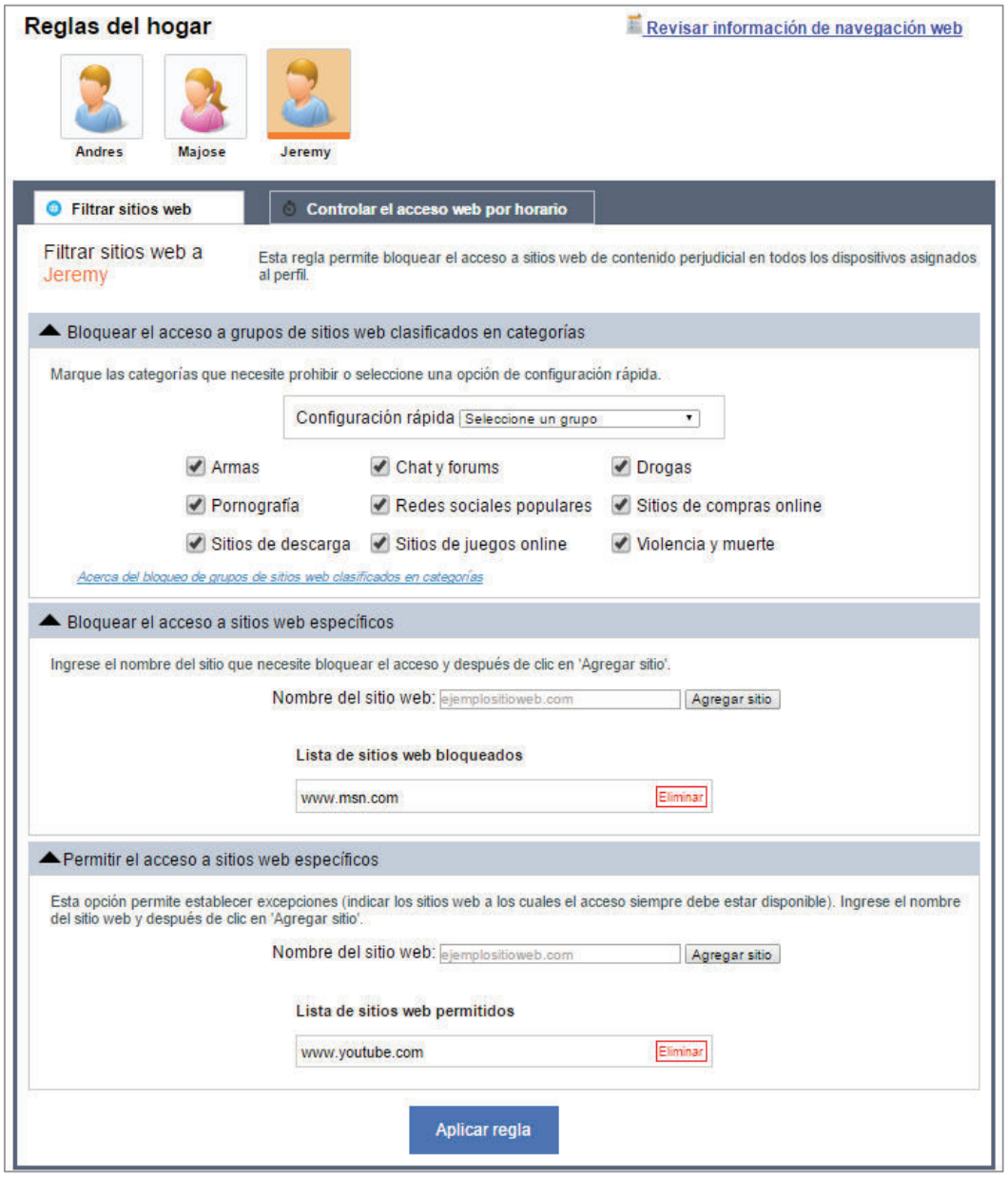

Figura 3.7 Interfaz Reglas del hogar opción Filtrar sitios web

En la Figura 3.7 se observa que se ha seleccionado el perfil “Jeremy” para bloquearle, a todos los dispositivos de este perfil, el acceso a los sitios web de todas las categorías. Además, se ha ingresado el nombre del sitio web “www.msn.com” para bloquear su acceso. Así mismo, se ha ingresado el nombre del sitio web “www.youtube.com” para permitir su acceso, el cual pertenece a la categoría “Redes sociales populares” con acceso bloqueado.

En la Figura 3.8 se muestra la interfaz en donde se puede aplicar la regla para establecer el horario de acceso web permitido, bloquear el acceso o permitir el acceso a los dispositivos que se encuentran asignados a un perfil. Para aplicar una regla se debe seleccionar el perfil y a continuación se debe escoger un dispositivo. Se puede observar que se ha seleccionado el dispositivo “cel-jeremy” del perfil “Jeremy” para asignarle un horario de acceso (ver Figura 3.8).

Visualización de información sobre la navegación web de los usuarios

A continuación se presentan las interfaces para visualizar la información sobre la navegación web realizada con cada dispositivo asignado a un perfil.

En la Figura 3.9 se muestra la interfaz que despliega los registros de los intentos bloqueados de acceso web que fueron realizados con el dispositivo “tab-jeremy” del perfil “Jeremy”. Se visualizan los registros por fecha, y en cada uno se muestra la URL que fue solicitada y la hora del intento de acceso. Como en todas las interfaces de información de navegación web, se puede filtrar los registros por período, eliminar registros y actualizar la información.

En la Figura 3.10 se muestra la interfaz que despliega los registros del tiempo de navegación web del dispositivo “tab-jeremy” que se encuentra asignado al perfil “Jeremy”. Se visualizan los registros por fecha, y en cada uno se muestra la cantidad de tiempo.

En la Figura 3.11 se muestra la interfaz para visualizar la lista de sitios web más visitados con el dispositivo “tab-jeremy” del perfil “Jeremy”. Se visualizan los registros por fecha, y en cada uno se muestra el nombre del sitio web y el número de veces que fue visitado.

Para terminar con la presentación de las interfaces de información navegación web, en la Figura 3.12 se muestra la interfaz que despliega los registros de las

páginas web visitadas con el dispositivo “tab-jeremy” del perfil “Jeremy”. Se visualizan los registros por fecha, y en cada uno se muestra la URL de la página web visitada y la hora de la visita.

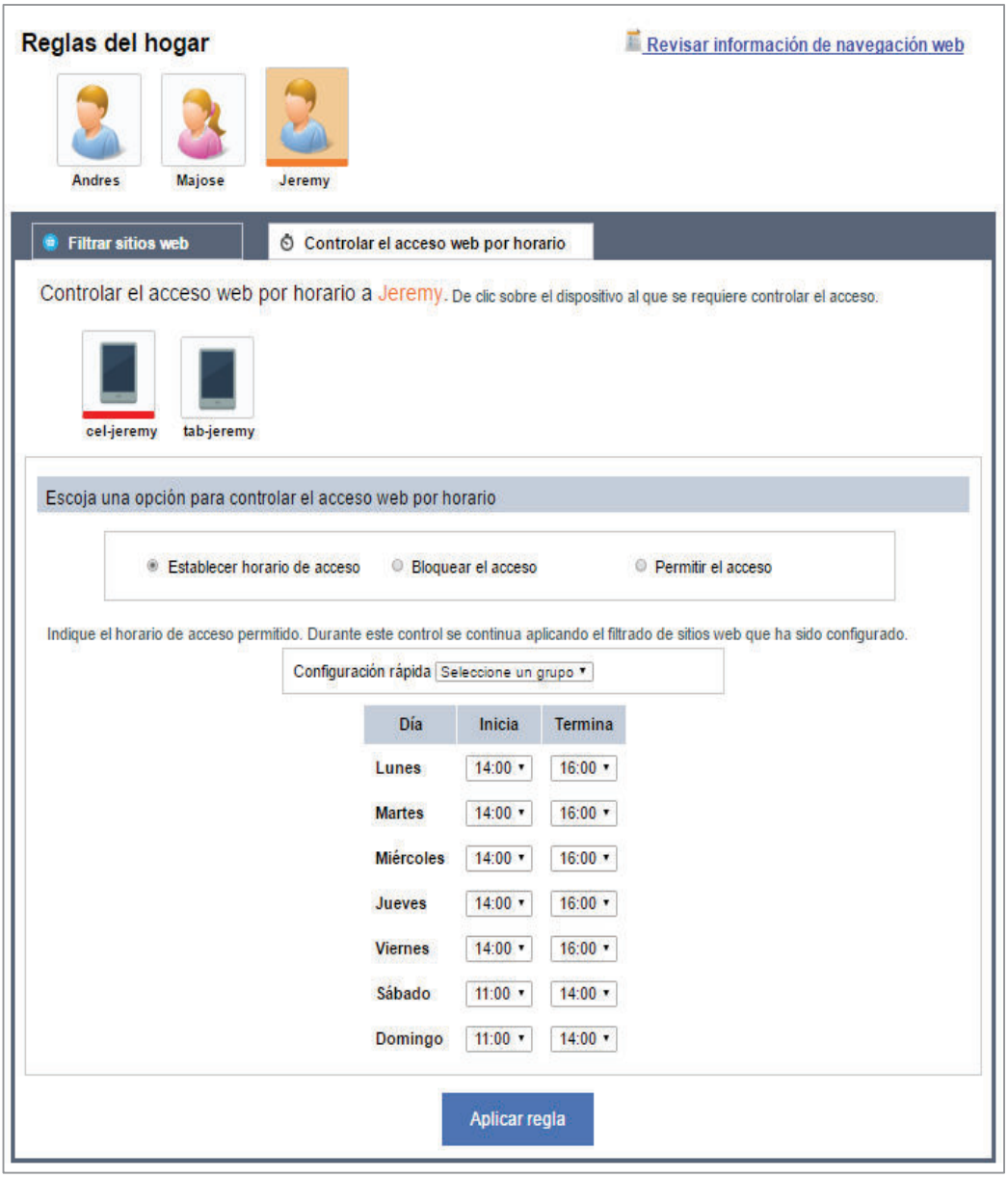

Figura 3.8 Interfaz Reglas del hogar opción Control de acceso por horario

Control de tráfico a nivel de red

En la Figura 3.13 se presenta la interfaz para permitir tráfico de servicios de Internet en capa de red. Se puede escoger ciertos servicios, de una lista predefinida, o se puede registrar el nombre del servicio y el número de puerto que utiliza. Se observa que se encuentra registrada la aplicación “thunderbird” (cliente

de correo electrónico) y que su tráfico se encuentra permitido (registro con estado habilitado). Se puede cambiar el estado del registro a deshabilitado. Esto permite mantener registrada la información del servicio, pero no permitir su tráfico. También se puede eliminar el registro.

Figura 3.9 Información de navegación web opción Alertas

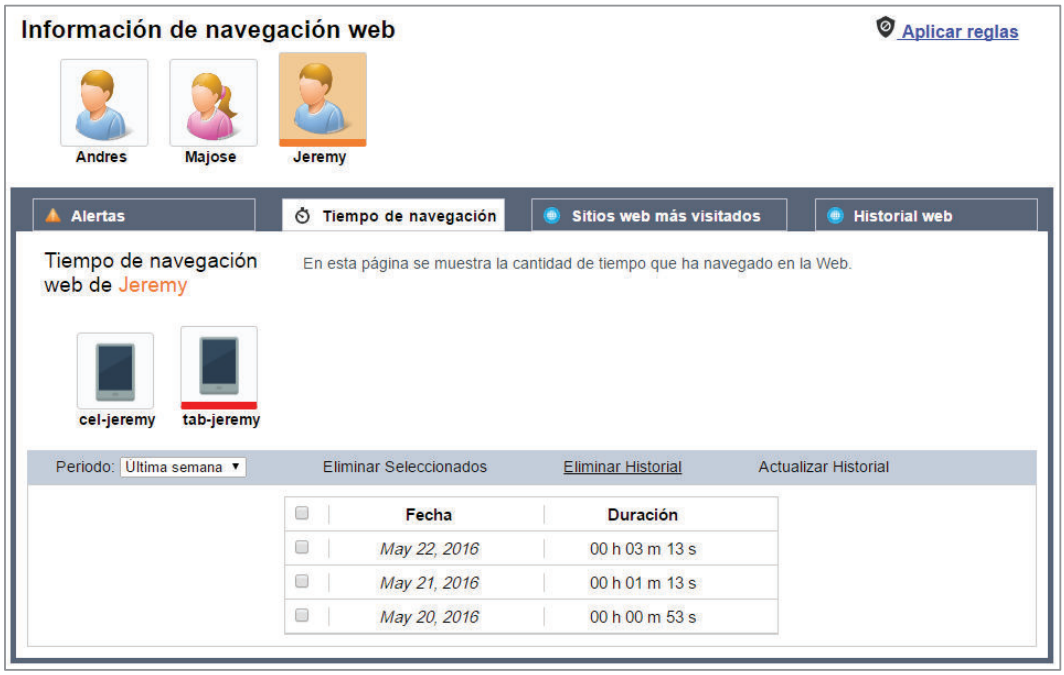

Figura 3.10 Interfaz Información de navegación web opción Tiempo de navegación web

Figura 3.11 Interfaz Información de navegación web opción Sitios web más visitados

Figura 3.12 Interfaz Información de navegación web opción Historial web

Figura 3.13 Interfaz Permitir tráfico de servicios de Internet

Editar perfil

En la opción “Perfiles creados” de la Figura 3.4 se listan los registros de perfiles, escoger la opción “Editar perfil” de uno de ellos. A continuación, se accede a la interfaz “Editar Perfil”, en donde se puede cambiar el nombre del perfil, liberar dispositivos o asignar dispositivos pertenecientes a otros perfiles o sin asignar. Cuando se asigna un dispositivo de otro perfil se muestra un aviso (ver Figura 3.14).

Mensajes y avisos

Cuando el administrador toma una acción en la interfaz que cambia la configuración del prototipo, se pide una confirmación, de esta manera se advierte al administrador sobre el resultado de dicha acción. Por ejemplo, en la Figura 3.15 se muestra al administrador la solicitud de confirmación, mediante un cuadro de diálogo, cuando está a punto de eliminar un dispositivo del sistema.

De igual manera, se proporciona mensajes de resultado a las acciones realizadas por el administrador. En la Figura 3.16 se observa el mensaje mostrado cuando un perfil no tiene asignados dispositivos. Además, en la Figura 3.17 se observa el mensaje de respuesta después de aplicar una regla.

Figura 3.14 Interfaz Editar perfil

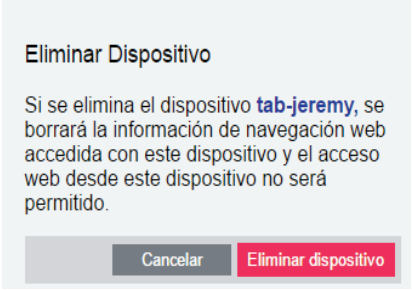

Figura 3.15 Cuadro de diálogo para eliminar un dispositivo

Figura 3.16 Aviso de estado de perfiles sin dispositivos

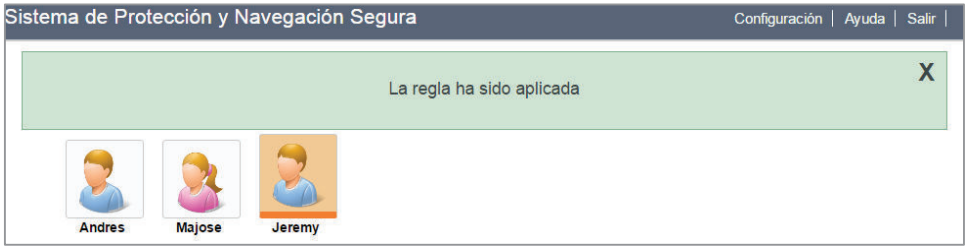

Figura 3.17 Mensaje de resultado de una transacción exitosa

Cuando se accede a la interfaz de administración por primera vez, se muestra un mensaje con información sobre los primeros pasos que deben ser realizados para utilizar las funcionalidades del sistema, tal como se muestra en la Figura 3.18.

Figura 3.18 Cuadro de diálogo sobre los primeros pasos en el sistema

3.3 PRUEBAS

Para verificar el funcionamiento del prototipo se realizaron diferentes pruebas: pruebas de carga en procesador y memoria, evaluación de usabilidad de la interfaz de administración, pruebas unitarias y pruebas de aceptación.

Para las pruebas de carga y aceptación y, en general, para utilizar las funcionalidades de control de acceso web del prototipo, es necesario configurar el navegador web de cada *host* de la red interna con la dirección IP y puerto del servidor proxy HTTP.

3.3.1 PRUEBAS UNITARIAS

Durante el desarrollo de la aplicación web se escribieron pruebas unitarias para: verificar que el código nuevo funcione como se espera y verificar que la funcionalidad del código escrito no fue afectada cuando se realizó una reestructuración del mismo.

Django permite escribir pruebas unitarias utilizando módulos embebidos, y algunos *framework* de prueba (e. g., unittest). Las pruebas unitarias en Django permiten probar código de los distintos niveles del patrón Modelo-Plantilla-Vista, e. g., verificar el mapeo entre URL y vista, verificar el contenido de la página

HTML enviada como respuesta, verificar el resultado de una vista. Adicionalmente, Django tiene un intérprete de comandos, el cual es muy útil para *debugging.*

Las pruebas unitarias fueron desarrolladas utilizando unittest (la utilización de este *framework* de prueba está documentada en el Proyecto Django). Además se utilizó constantemente el intérprete de comandos para probar código.

Prueba unitaria de la vista `restablecer_configuracion`

Las pruebas unitarias deben ser escritas en el archivo `test.py`, que se encuentra en el directorio de la aplicación web, para su ejecución. En el Código 3.41 se presenta como ejemplo de pruebas unitarias el código de la clase `RestablecerConfiguracionTest`, la cual verifica el resultado de la vista `restablecer_configuracion`. Esta vista se encarga de procesar peticiones HTTP dirigidas a la URL `/restablecer-configuracion/`. Los resultados que se esperan de esta vista son: restablecer el archivo de configuración inicial de Squid y el script con la configuración inicial de iptables, así como eliminar los archivos de perfiles que fueron creados para almacenar las direcciones MAC de sus dispositivos asignados.

```
class RestablecerConfiguracionTest(unittest.TestCase):
    def setUp(self):
        self.client = Client()
        self.client.post('/restablecer-configuracion/')
    def test_reset_iptables_script(self):
        archivos_iguales                                                    =
cmp('/home/project/iptables.sh','/home/project/iptables.sh.inicial')
        self.assertEqual(archivos_iguales,True)
    def test_reset_squidconf(self):
        archivos_iguales                                                    =
cmp('/etc/squid/squid.conf','/etc/squid/squid.conf.inicial')
        self.assertEqual(archivos_iguales,True)
    def test_eliminacion_archivos_de_perfil(self):
        archivos_eliminados = True
        lista_archivos = listdir('/etc/squid/listas')
        for var in lista_archivos:
            if re.search('[0-9]',var):
                archivos_eliminados = False
        self.assertEqual(archivos_eliminados, True)
```

Código 3.41 Clase `RestablecerConfiguracionTest`

Ejecución de pruebas unitarias

El comando del Código 3.42 es utilizado para ejecutar pruebas unitarias. Se pueden ejecutar todas las pruebas escritas en el archivo `test.py` pasando como

argumento el nombre de la aplicación, o se puede ejecutar una clase (de la prueba unitaria) en particular pasando como argumento el nombre de la misma.

```
root@sistema:/home/project$ python manage.py test <nombre_aplicacion |
nombre de clase>
```

Código 3.42 Comando para ejecutar las pruebas de código

En la Figura 3.19 se puede observar el resultado de la ejecución de la clase `RestablecerConfiguracionTest`. El resultado es afirmativo (OK), lo que significa que se ha verificado el resultado esperado.

```
root@sistema:/home/project# python manage.py test prototipo.tests.RestablecerConfiguracionTest
Creating test database for alias 'default'...
...
----------------------------------------------------------------------
Ran 3 tests in 0.672s

OK
Destroying test database for alias 'default'...
```

Figura 3.19 Ejecución de la clase `RestablecerConfiguracionTest`

Cabe mencionar que Django crea una base de datos especial para las pruebas unitarias. Esta base de datos se encuentra vacía, por tanto, para probar el código de un modelo, se debe crear primero una instancia del modelo.

3.3.2 PRUEBAS DE ACEPTACIÓN

Después de finalizar cada iteración se realizaron las respectivas pruebas de aceptación para verificar y validar la implementación de las historia de usuario. Por una historia de usuario se puede realizar más de una prueba de aceptación, donde se determinan los posibles escenarios para llegar a un resultado válido para el usuario y que cumpla con la definición de la historia de usuario [46].

Las pruebas de aceptación fueron realizadas y documentadas de acuerdo a la plantilla presente en la Tabla 3.2. Los elementos de esta plantilla se describen a continuación:

- ***Historias de usuario:*** Número que identifica a la o las historias de usuario involucradas en la prueba de aceptación.
- ***Descripción:*** Información sobre la funcionalidad que se está verificando en la prueba de aceptación.

- ***Condiciones de ejecución*:** Tareas que se deben cumplir para la ejecución de la prueba.
- ***Entrada:*** Procedimiento que produce el resultado esperado.
- ***Resultado esperado:*** Salida producida por el prototipo según la entrada.
- ***Evaluación de la prueba:*** Valoración del resultado con la caracterización de aprobado o fallido.

Prueba de aceptación	**Número:**
Historias de usuario:	
Descripción:	
Condiciones de ejecución:	
Entrada:	
Resultado esperado:	
Evaluación de la prueba:	

Tabla 3.2 Plantilla de una prueba de aceptación [46]

3.3.2.1 Primera Iteración

Al final de esta iteración se ejecutaron las pruebas de aceptación: Acceso web bloqueado a dispositivos no registrados (ver Tabla 3.3), Registro y asignación de un dispositivo a un perfil (ver Tabla 3.4), Bloquear el acceso a sitios web de categorías y permitir el acceso a sitio web (ver Tabla 3.6 y Tabla 3.6), Bloquear el acceso a un sitio web (ver Tabla 3.7).

Prueba de aceptación	**Número:** 1
Historias de usuario: 1	
Descripción: Acceso web bloqueado a dispositivo no registrado en el sistema	
Condiciones de ejecución: Dispositivo conectado a la red interna	
Entrada: Generar una petición HTTP para acceder a un recurso web	
Resultado esperado: - Petición HTTP denegada, y se obtiene como respuesta la página web de error “acceso no permitido a dispositivos no registrados”.	
Evaluación de la prueba: Aprobada	

Tabla 3.3 Prueba de aceptación Acceso web bloqueado a dispositivos no registrados

Prueba de aceptación	**Número:** 2
Historias de usuario: 1,3	
Descripción: Registro y asignación de un dispositivo a un perfil y verificación de acceso web sin restricciones a este dispositivo	
Condiciones de ejecución: Acceder a la aplicación web desde el dispositivo a registrar	
Entrada: - Acceder a la aplicación web. Después, escoger la opción “Dispositivos registrados” y seguir el proceso de registro provisto por la aplicación web. - Generar una petición HTTP para acceder a un recurso web desde el dispositivo registrado y asignado al perfil.	
Resultado esperado: - En la aplicación web, se redirige a la interfaz de “Reglas del hogar” y muestra un mensaje de aviso que indica el éxito de la operación. - Petición HTTP permitida.	
Evaluación de la prueba: Aprobada	

Tabla 3.4 Prueba de aceptación Registro y asignación de un dispositivo a un perfil

Prueba de aceptación	**Número:** 3
Historias de usuarios: 6, 8	
Descripción: A un perfil bloquearle el acceso a sitios web pertenecientes a la categoría “Redes sociales populares” y permitirle el acceso al sitio web “www.facebook.com” (perteneciente a la categoría a bloquear)	
Condiciones de ejecución: Tres dispositivos registrados y asignados a un perfil	
Entrada: - Acceder a la aplicación web. Después, acceder a la interfaz “Reglas del hogar opción Filtrar sitios web”. Luego, seleccionar el perfil al cual se le aplicará la regla. A continuación, marcar la categoría de “Redes sociales populares” e ingresar “www.facebook.com” en la casilla nombre de sitio web permitido y dar clic en “Agregar sitio web”. Finalmente, dar clic en el botón “Aplicar regla”. - Generar una petición HTTP para acceder a un sitio web perteneciente a la categoría bloqueada desde los dispositivos asignados al perfil. - Generar una petición HTTP para acceder a un sitio web no perteneciente a la categoría bloqueada desde los dispositivos asignados al perfil. - Generar una petición HTTP para acceder al sitio web “www.facebook.com” desde los dispositivos asignados al perfil.	

Tabla 3.5 Prueba de aceptación Bloquear el acceso a sitios web de categorías y permitir el acceso a sitio web

Prueba de aceptación	**Número:** 3
Resultado esperado: - En la aplicación web, se muestra un mensaje de aviso que indica que la regla ha sido aplicada. - La petición HTTP dirigida a un sitio web perteneciente a la categoría bloqueada es denegada y se obtiene como respuesta la página web de error “acceso no permitido a sitios de categorías bloqueadas”. - La petición HTTP dirigida a un sitio web no perteneciente a la categoría bloqueada es permitida. - La petición HTTP dirigida al sitio web “www.facebook.com” es permitida.	
Evaluación de la prueba: Aprobada	
Observaciones: Cuando Squid sirve una página web de error en respuesta a la denegación de una petición HTTPS, el navegador web (Chrome, Firefox, Explorer) no despliega esta página web, en su lugar presenta un mensaje de error de conexión, el cual es provisto por el propio navegador. El navegador web realiza una petición HTTPS y espera la respectiva respuesta HTTPS, pero en su lugar, Squid le envía una respuesta HTTP. Es por esta razón que el navegador interpreta que ha ocurrido un error en la conexión. Esta operación del navegador web no permite informar al usuario de la red interna cuando Squid bloquea peticiones HTTPS.	

Tabla 3.6 Prueba de aceptación Bloquear el acceso a sitios web de categorías y permitir el acceso a sitio web (continuación)

Prueba de aceptación	**Número:**4
Historias de usuario: 7	
Descripción: A un perfil bloquearle el acceso al sitio web “www.taringa.net”	
Condiciones de ejecución: Tres dispositivos registrados y asignados a un perfil.	
Entrada: - Acceder a la aplicación web. Después, acceder a la interfaz “Reglas del hogar opción Filtrar sitios web”. Luego, seleccionar el perfil al cual se le aplicará la regla. A continuación, ingresar ““www.taringa.net”” en la casilla nombre de sitio web bloqueado y dar clic en el botón “Agregar sitio web”. Finalmente, dar clic en el botón “Aplicar regla”. - Generar una petición HTTP para acceder al sitio web “www.taringa.net” desde los dispositivos asignados al perfil.	
Resultado esperado: - En la aplicación web, se muestra un mensaje de aviso que indica que la regla ha sido aplicada. - La petición HTTP dirigida al sitio web “www.taringa.net” y se obtiene como respuesta la página web de error “sitio web con acceso no permitido”.	
Evaluación de la prueba: Aprobada	

Tabla 3.7 Prueba de aceptación Bloquear el acceso a un sitio web

3.3.2.2 Segunda Iteración

Al finalizar esta iteración se realizaron las pruebas de aceptación: Control de acceso web por horario y por origen (ver Tabla 3.8), Filtrado de sitios web y control de acceso por horario y por origen (ver Tabla 3.9).

Prueba de aceptación	**Número:** 5
Historias de usuario: 10 , 11, 12	
Descripción: Se tiene tres dispositivos (PC1, PC2, PC3) asignados a un perfil. A PC1 asignarle un horario de acceso web. A PC2 permitirle el acceso web sin restricción de horario. A PC3 bloquearle el acceso web.	
Condiciones de ejecución: - Tres dispositivos registrados y asignados a un perfil.	
Entrada: - Acceder a la aplicación web. Después, acceder a la interfaz “Reglas del hogar opción Controlar el acceso web por horario”. Luego, seleccionar el perfil al cual se le aplicará la regla. A continuación, seleccionar el dispositivo al que se le configurará el control de acceso. - Para PC1, escoger la opción “Establecer horario de acceso”. Después, establecer el horario de acceso permitido de lunes a viernes de 12:00 a 15:00. Finalmente, dar clic en Aplicar reglar. - Para PC2, escoger la opción “Bloquear acceso” y dar clic en Aplicar reglar. - Para PC3, escoger la opción “Permitir acceso” y dar clic en Aplicar reglar. - Generar una petición HTTP para acceder un recurso web desde PC1 en el horario permitido. - Generar una petición HTTP para acceder un recurso web desde PC1 fuera del horario permitido. - Generar una petición HTTP para acceder un recurso web desde PC2 en el cualquier horario. - Generar una petición HTTP para acceder un recurso web desde PC3 en el cualquier horario.	
Resultado esperado: - En la aplicación web, para cada configuración realizada se muestra un mensaje de aviso que indica que la regla ha sido aplicada. - La petición HTTP dirigida al recurso web desde PC1 en el horario permitido es permitida. - La petición HTTP dirigida al recurso web desde PC1 fuera del horario permitido es denegada y se obtiene como respuesta la página web de error “acceso fuera de horario permitido”. - La petición HTTP dirigida al recurso web desde PC2 es denegada. - La petición HTTP dirigida al recurso web desde PC3 es permitida.	
Evaluación de la prueba: Aprobada	

Tabla 3.8 Prueba de aceptación Control de acceso web por horario y por origen

Prueba de aceptación	**Número:** 6
Historias de usuario: 6 , 7, 8, 10 ,11,12	
Descripción: Verificar que el bloqueo y permiso de acceso a sitios web se mantenga aplicado cuando se aplica el control de acceso web por horario y se permite el acceso web sin restricción de horario.	
Condiciones de ejecución: - Pruebas de aceptación No. 3, No.4 y No.5 aprobadas	
Entrada: - Generar una petición HTTP para acceder un sitio web perteneciente a la categoría bloqueada desde PC1 en el horario permitido. - Generar una petición HTTP para acceder un sitio web no perteneciente a la categoría bloqueada desde PC1 en el horario permitido. - Generar una petición HTTP para acceder al sitio web “www.facebook.com” (con acceso permitido) desde PC1 en el horario permitido. - Generar una petición HTTP para acceder al sitio web www.taringa.net (con acceso bloqueado) desde PC1 en el horario permitido. - Generar una petición HTTP para acceder un sitio web perteneciente a la categoría bloqueada desde PC3 en cualquier horario. - Generar una petición HTTP para acceder un sitio web no perteneciente a la categoría bloqueada desde PC3 en cualquier horario. - Generar una petición HTTP para acceder al sitio web “www.facebook.com” (con acceso permitido) desde PC3 en cualquier horario. - Generar una petición HTTP para acceder al sitio web “www.taringa.net” (con acceso bloqueado) desde PC3 en cualquier horario.	
Resultado esperado: - Las peticiones HTTP dirigidas a un sitio web perteneciente a la categoría bloqueada desde PC1, en el horario permitido, y desde PC3, en cualquier horario, son denegadas y se obtiene como respuesta la página web de error “acceso no permitido a sitios de categorías bloqueadas”. - Las peticiones HTTP dirigidas a un sitio web no perteneciente a la categoría bloqueada desde PC1, en el horario permitido, y desde PC3, en cualquier horario, son permitidas. - Las peticiones HTTP dirigidas al sitio web “www.taringa.net” desde PC1, en el horario permitido, y desde PC3, en cualquier horario, son denegadas y se obtiene como respuesta la página web de error “sitio web con acceso no permitido”. - Las peticiones HTTP dirigidas al sitio web “www.facebook.com” desde PC1, en el horario permitido, y desde PC3, en cualquier horario, son permitidas.	
Evaluación de la prueba: Aprobada	

Tabla 3.9 Prueba de aceptación Filtrado de sitios web y control de acceso por horario y por origen

3.3.2.3 Tercera Iteración

Al finalizar esta iteración se realizó la prueba de aceptación Buscar registros de historial web que se muestra en la Tabla 3.10.

Prueba de aceptación	**Número:** 7
Historias de usuario: 15	
Descripción: Buscar registros de Historial web	
Condiciones de ejecución: - Un dispositivo registrado y asignado a un perfil. - Registros de historial web del dispositivo.	
Entrada: Acceder a la aplicación web. Después, acceder a la interfaz “Información de navegación web”. Luego, seleccionar el perfil. Después, seleccionar el “Historial web”. A continuación, seleccionar el dispositivo cuya información de navegación web que accedida utilizándolo requiere ser revisada. Finalmente, ingresar la palabra de búsqueda en la casilla correspondiente y dar clic el botón “Buscar en historial”.	
Resultado esperado: - Despliegue de los registros que contengan la palabra de búsqueda	
Evaluación de la prueba: Aprobada	

Tabla 3.10 Prueba de aceptación Buscar registros de historial web

3.3.2.4 Cuarta Iteración

Al finalizar esta iteración se realizaron las pruebas de aceptación: Visualizar información de navegación web (ver Tabla 3.11) y Eliminar de registros de información de navegación web (ver Tabla 3.12).

3.3.2.5 Quinta Iteración

Al finalizar esta iteración se realizaron las pruebas de aceptación: Permitir tráfico de correo electrónico (ver Tabla 3.13), Bloquear tráfico de un servicio de Internet registrado (ver Tabla 3.14), Editar cuenta de administrador (ver Tabla 3.15), Cambiar contraseña (ver Tabla 3.16), Centro de ayuda (ver Tabla 3.17), Asignar dispositivo asignado a otro perfil (ver Tabla 3.18), Eliminar registro de dispositivo (ver Tabla 3.19), Eliminar registro de perfil (ver Tabla 3.20), Restablecer configuración inicial (ver Tabla 3.21).

Prueba de aceptación	**Número:** 8
Historias de usuario: 13, 14, 16, 17	
Descripción: Revisar a un perfil el historial web, intentos de acceso bloqueados, tiempo de navegación web y sitios web más visitados, utilizando los filtros: hoy, última semana, último mes y todo.	
Condiciones de ejecución: - Un dispositivo registrado y asignado a un perfil - Registros de historial web, intentos de acceso bloqueados, tiempo de navegación web y sitios web más visitados del dispositivo	
Entrada: Acceder a la interfaz "Información de navegación web" y seleccionar el perfil. A continuación, seleccionar el tipo de información de navegación web y luego seleccionar el dispositivo. Finalmente, utilizar los filtros: hoy, últimos 7 días, últimos 30 días y todo.	
Resultado esperado: Despliegue de registros de intentos de acceso web bloqueados, tiempo de navegación web y sitios web más visitados de hoy, última semana, último mes y todo.	
Evaluación de la prueba: Aprobada	

Tabla 3.11 Prueba de aceptación Visualizar información de navegación web

Prueba de aceptación	**Número:** 9
Historias de usuario: 13, 14, 16, 17	
Descripción: Eliminar registros por período y registros específicos de historial web, intentos de acceso web bloqueados, tiempo de navegación web y sitios web más visitados.	
Condiciones de ejecución: - Un dispositivo registrado y asignado a un perfil - Registros de historial web, intentos de acceso web bloqueados, tiempo de navegación web y sitios web más visitados del dispositivo	
Entrada: Acceder a la interfaz "Información de navegación web" y seleccionar el perfil. A continuación, seleccionar el tipo de información de navegación web y luego seleccionar el dispositivo. Después, dar clic en "Eliminar historial", escoger el periodo a eliminar, y confirmar la eliminación. Finalmente, seleccionar los registros específicos que requieran se eliminados y dar clic en "Eliminar registros seleccionados".	
Resultado esperado: - Eliminación de los registros de la base de datos. - En la aplicación web se muestra un mensaje de aviso que indica que los registros del periodo seleccionado han sido eliminados. - En la aplicación web se muestra un mensaje de aviso que indica que los registros seleccionados han sido eliminados.	
Evaluación de la prueba: Aprobada	

Tabla 3.12 Prueba de aceptación Eliminar de registros de información de navegación web

Prueba de aceptación	**Número:**10
Historia de usuario: 19	
Descripción: Permitir tráfico correspondiente la aplicación Thunderbird	
Condiciones de ejecución: - Thunderbird instalada y configurada	
Entrada: - Acceder a la aplicación web. Después escoger la opción “Permitir tráfico de servicios de Internet”. A continuación, registrar la aplicación con el nombre “Thunderbird”, el número de puerto “993” y con estado en “habilitado”. - Descargar correo electrónico con Thunderbird.	
Resultado esperado: - En la aplicación, se muestra un mensaje de aviso que indica que el servicio ha sido registrado. - Correo electrónico descargado con Thunderbird.	
Evaluación de la prueba: Aprobada	

Tabla 3.13 Prueba de aceptación Permitir tráfico de correo electrónico

Prueba de aceptación	**Número:** 11
Historia de usuario: 20	
Descripción: Cambiar el estado de habilitado a deshabilitado de un registro de un servicio de Internet.	
Condiciones de ejecución: - Prueba de aceptación No. 10 aprobada	
Entrada: - Acceder a la aplicación web. Después, escoger la opción “Permitir tráfico de servicios de Internet”. A continuación, seleccionar el registro de “Thunderbird” y cambiar el estado de “habilitado” a “deshabilitado”. - Descargar correo electrónico con Thunderbird.	
Resultado esperado: - En la aplicación, se muestra un mensaje de aviso que indica que el registro ha sido modificado correctamente. - Thunderbird no se puede conectar con el servidor de correo.	
Evaluación de la prueba: Aprobada	

Tabla 3.14 Prueba de aceptación Bloquear tráfico de un servicio de Internet registrado

Prueba de aceptación	**Número:**12
Historia de usuario: 22	
Descripción: Modificar cuenta para administrador	
Condiciones de ejecución: Ninguna	
Entrada: - Acceder a la interfaz de “Configuracion inicial” y dar clic en la opción “Cuenta de administrador” para desplegar la información de esta cuenta. Luego, modificar el nombre de usuario, la contraseña, la pregunta (para cambiar contraseña) y su respectiva respuesta. El cambio de contraseña se realiza en un cuadro de diálogo, en el cual se pide confirmación del nombre de usuario y contraseña actual antes de proceder al cambio. - Ingresar a la aplicación web con el nuevo nombre de usuario y nueva contraseña.	
Resultado esperado: - Se valida que se ingrese una contraseña de al menos 8 caracteres. - Se valida que el nombre de usuario es el registrado. - Se valida que se ingresa un nombre de usuario o la respuesta a la pregunta. - Se permite el ingreso al sistema con las nuevas credenciales (nombre de usuario y contraseña).	
Evaluación de la prueba: Aprobada	

Tabla 3.15 Prueba de aceptación Editar cuenta de administrador

Prueba de aceptación	**Número:**13
Historia de usuario: 23, 24	
Descripción: Cambio de contraseña	
Condiciones de ejecución: Ninguna	
Entrada: - Acceder a la interfaz de “Ingreso” a la aplicación web. Después, dar clic sobre la opción “Contraseña olvidada”. A continuación, se despliega la interfaz “Cambio de contraseña olvidada” para ingresar el nombre de usuario y la respuesta a la pregunta para cambiar contraseña; el sistema valida esta información para permitir el cambio de contraseña.	
Resultado esperado: - Se valida la información ingresada y después se pide el ingreso de la nueva contraseña. - Se cambia la contraseña. - Se permite el acceso al sistema si el nombre de usuario y la nueva contraseña son válidos, caso contrario se muestra un aviso que son incorrectas.	
Evaluación de la prueba: Aprobada	

Tabla 3.16 Prueba de aceptación Cambiar contraseña

Prueba de aceptación	**Número:**14
Historia de usuario: 25	
Descripción: Visualizar en la aplicación web el manual de usuario que contiene información sobre cómo configurar las funcionalidades del prototipo.	
Condiciones de ejecución: Ninguna	
Entrada: Acceder a la aplicación web. En la parte superior de la interfaz “Configuración inicial” dar clic en “Ayuda”. A continuación, se despliega un menú con dos opciones: primeros pasos y centro de ayuda. Escoger centro de ayuda.	
Resultado esperado: - Despliegue de la interfaz “Centro de ayuda”.	
Evaluación de la prueba: Aprobada	

Tabla 3.17 Prueba de aceptación Centro de ayuda

Prueba de aceptación	**Número:**15
Historias de usuario: 4	
Descripción: Asignar el dispositivo del “Perfil-1” al “Perfil-2”.	
Condiciones de ejecución: - Dos perfiles, cada uno con un dispositivo asignado. - Registros de información de navegación web de cada dispositivo.	
Entrada: Acceder a la aplicación web. Después, acceder a la interfaz “Configuración inicial”. A continuación, seleccionar la opción “Perfiles creados”. Después, seleccionar el “Perfil-2”. A continuación, se despliega la información del perfil y los dispositivos que tienen asignados así como los dispositivos asignados a otros perfiles (en cada registro de dispositivo se muestra el nombre del perfil al cual está asignado). Después, escoger el dispositivo que pertenece al “Perfil-1”para asignarlo; se muestra un aviso de que la información de navegación web y las reglas aplicadas al dispositivo serán eliminadas si se completa la modificación. Finalmente, dar clic en el botón “Guardar cambios”. Generar una petición HTTP para acceder a cualquier recurso web.	
Resultado esperado: - Borrado de registros de información de navegación del dispositivo. - Se redirige a la interfaz Configuración Inicial y se muestra un aviso que indica la modificación realizada. - El “Perfil-1” queda sin dispositivos, y en la interfaz Configuración inicial, Reglas del hogar e Información de navegación web se muestra un aviso que lo indica. - El nuevo dispositivo tiene el acceso web permitido sin restricciones. - Petición HTTP permitida (el dispositivo ahora tiene acceso web sin restricciones)	
Evaluación de la prueba: Aprobada	

Tabla 3.18 Prueba de aceptación Asignar dispositivo asignado a otro perfil

Prueba de aceptación	**Número:**16
Historia de usuario: 2	
Descripción: Eliminar el registro de dispositivo	
Condiciones de ejecución: Un dispositivo registrado y asignado a un perfil.	
Entrada: - Acceder a la aplicación web. Después, seleccionar la opción Dispositivos registrados y dar clic en el botón “Eliminar dispositivo” en el registro del dispositivo. A continuación, se muestra un cuadro de diálogo que pide confirmar la operación. Finalmente, confirmar la eliminación. - Generar una petición HTTP a un recurso web desde el dispositivo cuyo registro fue eliminado.	
Resultado esperado: - Se muestra un aviso de que el registro del dispositivo ha sido eliminado. - En la interfaz Configuración inicial, Reglas del hogar e Información de navegación web se muestra un aviso que indica que el perfil no tiene dispositivos asignados. - Petición HTTP denegada, y se obtiene como respuesta la página web de error “acceso no permitido a dispositivos no registrados”.	
Evaluación de la prueba: Aprobada	

Tabla 3.19 Prueba de aceptación Eliminar registro de dispositivo

Prueba de aceptación	**Número:**17
Historia de usuario: 5	
Descripción: Eliminar el registro de perfil	
Condiciones de ejecución: Un dispositivo registrado y asignado a un perfil.	
Entrada: - Acceder a la aplicación web. Después, seleccionar la opción “Perfiles creados” y dar clic en “Eliminar perfil” en el registro del perfil. A continuación, se muestra un cuadro de diálogo que pide confirmar la operación. Finalmente, confirmar la eliminación. - Generar una petición HTTP a cualquier recurso web desde el dispositivo que estaba asignado al perfil cuyo registro fue eliminado.	
Resultado esperado: - Se muestra un aviso que el registro del perfil ha sido eliminado. - Liberar el dispositivo que estaba asignado al perfil eliminado y borrar registros de información de navegación web asociados. - Petición HTTP del dispositivo liberado es permitida.	
Evaluación de la prueba: Aprobada	

Tabla 3.20 Prueba de aceptación Eliminar el registro de perfil

Prueba de aceptación	**Número:**18
Historia de usuario: 21	
Descripción: Restablecer configuración inicial	
Condiciones de ejecución: - Un dispositivo registrado y asignado a un perfil - Información de navegación web recolectada - Reglas de filtrado web aplicadas - Servicios de Internet registrados para permitir su tráfico	
Entrada: - Acceder a la aplicación web. Después, dar clic sobre la opción “Restablecer configuración”. A continuación, se muestra un cuadro de diálogo que informa las consecuencias de esta operación y se pide confirmación. - Generar una petición HTTP a cualquier recurso web desde el dispositivo que estaba asignado al perfil antes de restablecer la configuración.	
Resultado esperado: - Se muestra un aviso que indica que se ha restablecido la configuración inicial. - Se carga el archivo de configuración inicial de Squid. - Se carga el *script* de configuración inicial de iptables. - Se borran todos los registros: perfiles, dispositivos, información de navegación web, servicios de Internet y reglas de filtrado web. - Petición HTTP denegada, y se obtiene como respuesta la página web de error “acceso no permitido a dispositivos no registrados”.	
Evaluación de la prueba: Aprobada	

Tabla 3.21 Prueba de aceptación Restablecer configuración inicial

3.3.3 EVALUACIÓN DE USABILIDAD

Al terminar la fase de desarrollo de iteraciones se realizó una prueba de usuario, con el propósito de observar el desempeño de los usuarios en la interfaz de administración y recolectar la información necesaria, para luego analizarla y determinar si la interfaz de administración es fácil de utilizar.

Los resultados de esta prueba de usuario permitirán evaluar los siguientes atributos de la interfaz de administración:

- Facilidad de aprendizaje
- Eficiencia
- Satisfacción

3.3.3.1 Participantes

Los participantes de la prueba fueron jefes de hogar, sin experiencia en la utilización de herramientas de software similares al prototipo desarrollado (pero con habilidad para utilizar dispositivos electrónicos para navegar en la Web) y con interés en participar. Los participantes fueron contactados personalmente por el autor. No se proporcionó remuneración de ningún tipo a los participantes por asistir a la prueba.

El número de participantes de la prueba fue determinado en base a la propuesta en [47], en la que se establece que realizar una prueba de usuario con 5 participantes, representantes de la audiencia objetivo, debería exponer la mayoría de problemas de usabilidad, cerca del 80% de ellos. Por eso, se seleccionó 5 participantes.

3.3.3.2 Metodología

La prueba de usuario estuvo compuesta de tres sesiones de prueba, con un período entre cada una de 48 horas. En cada sesión de prueba se pidió a potenciales usuarios del prototipo utilizar la interfaz de administración para realizar varias actividades (configurar funcionalidades del filtrado web y visualizar información de navegación web).

En cada sesión de prueba se midió el tiempo que les toma a los participantes realizar cada actividad. El propósito de realizar tres sesiones de prueba, es verificar si los participantes realizan las actividades en menor tiempo en cada sesión de prueba; así como verificar que tan rápido (en cuantas sesiones) el participante puede mejorar su rendimiento (realizar las actividades en menor tiempo).

En la primera sesión, a cada participante se le informó sobre las funcionalidades del sistema y después se dio lectura y se proporcionó una copia impresa de las actividades a realizar (una a la vez) mediante el uso de la interfaz de administración. Al terminar con las actividades, se pidió a cada participante responder un cuestionario para conocer el nivel de satisfacción experimentado luego de utilizar la interfaz.

En las dos sesiones de prueba siguientes, a cada participante se le proporcionó directamente una copia impresa de las actividades a realizar.

En cada sesión de prueba, a cada participante se le proporcionó las siguientes instrucciones:

- Realizar las actividades sin preocuparse de cometer errores, relajados como si estuviera en su hogar
- Si requiere aclaraciones o ayuda durante el desarrollo de las actividades puede pedirlas
- Considerar que este computador portátil es el computador utilizado por su hijo o sobrino para navegar en Internet
- Avisar cuando crea que ha terminado la actividad

La prueba fue realizada en un lugar confortable y calmado. El elemento primordial utilizado para recolectar información (grabar audio y capturar pantalla, aunque también se tomó apuntes) y acceder a la aplicación web fue un computador portátil con sistema operativo Windows 8 y navegador web Chrome versión 47.

La interfaz de administración fue desplegada en la plataforma de cómputo descrita en la sección 3.1; se utilizó el servidor web de prueba de Django para acceder a la interfaz de administración. Durante esta prueba de usuario no se habilitó el componente de filtrado de tráfico ni el componente de recolección de información de navegación web.

3.3.3.3 Escenarios de las actividades

Cada actividad a realizar fue descrita en un escenario lo más real posible, con el fin de comprometer al participante a realizar la actividad. Las actividades están relacionadas con una función de filtrado web y visualización de información de navegación web. Los escenarios de las actividades se muestran en la Tabla 3.22, cada escenario tiene como título el nombre de la función del sistema que fue utilizada; este título no fue presentado en la copia impresa de cada escenario que fue entregada a los participantes en las sesiones de prueba.

En cada sesión de prueba, las actividades fueron realizadas en un orden diferente, con el propósito de evitar que los participantes utilicen la interfaz de administración de forma automatizada (de memoria), y de comprobar que en

realidad han aprendido a utilizarla. En la Tabla 3.23 se presenta el orden de las actividades durante cada sesión de prueba.

Actividad 1: Bloquear el acceso a sitios web de categorías Se siente preocupado cuando su [hijo, hija, sobrina] navega en Internet porque puede acceder a sitios web de contenido perjudicial (por ejemplo: violencia, drogas, entre otros). Utilice este sistema para prohibir el acceso a esa clase de sitios web y así proteger a su [hijo, sobrino].
Actividad 2: Bloquear el acceso a un sitio web Su [hijo, sobrino] pasa mucho tiempo viendo videos en www.youtube.com y no realiza las tareas de la escuela. Utilice el sistema para prohibir el acceso a ese sitio web y así evitar que su [hijo, sobrino] se distraiga.
Actividad 3: Establecer horario de acceso web Su [hijo, sobrino] pasa demasiado tiempo navegando en Internet y ha dejado de ejercitarse jugando pelota con sus amigos. Como [padre, tío] Ud. desea evitar este problema. Utilice el sistema para establecer un horario de acceso adecuado que ayude a programar las actividades de su [hijo, sobrino].
Actividad 4: Revisar historial web Este sistema ha recolectado información sobre la actividad web de su [hijo, sobrino]. A Ud. le interesa acceder al sistema para revisar las páginas web que ha visitado en la última semana.

Tabla 3.22 Escenarios de las actividades

Sesión	Orden de realización
1	Actividad 1, Actividad 2, Actividad 3, Actividad 4
2	Actividad 1, Actividad 3, Actividad 2, Actividad 4
3	Actividad 1, Actividad 4, Actividad 3, Actividad 2

Tabla 3.23 Orden de realización de actividades

3.3.3.4 Resultados

Se documentan las mediciones del tiempo promedio para realizar cada una de las 4 actividades en cada sesión de prueba y el número de peticiones de ayuda realizadas por los participantes; en base a estos resultados se evalúa la eficiencia y facilidad de aprendizaje. También, se presentan las respuestas del cuestionario realizado para medir el nivel de satisfacción.

3.3.3.4.1 Eficiencia y facilidad de aprendizaje

Peticiones de ayuda realizadas por los participantes

En la primera sesión de prueba, dos participantes solicitaron ayuda para completar las actividades 1 y 2. Para realizar la actividad 1, cada uno de ellos

solicitó ayuda una vez. Para realizar la actividad 2, uno de estos participantes solicitó ayuda una vez. Para realizar las actividades 3 y 4 no solicitaron ayuda.

En las dos siguientes sesiones de prueba, ningún participante solicitó ayuda. Además, cabe mencionar que todos los participantes completaron cada una de las 4 actividades en las tres sesiones de prueba.

Tiempo promedio para realizar las actividades

La medición del tiempo para realizar cada actividad fue efectuada desde el instante en que el participante ingresó a la interfaz de administración y hasta el momento cuando el participante informó que ha terminado la actividad. En la Figura 3.20 se muestra una representación gráfica del tiempo promedio para realizar cada una de las cuatro actividades en función del número de sesión de prueba.

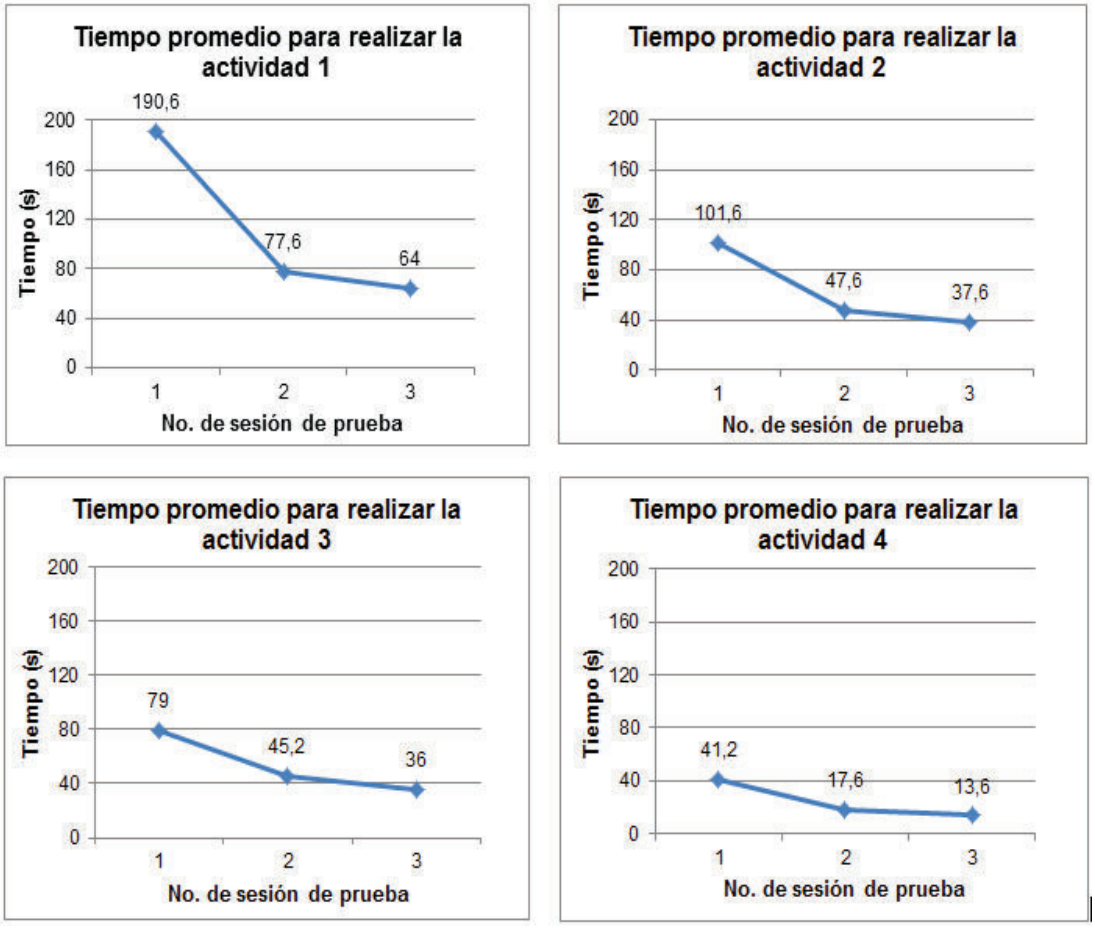

Figura 3.20 Representación gráfica del tiempo promedio para realizar las actividades

Análisis de resultados

Dos participantes solicitaron ayuda para realizar las actividades 1 y 2 en la primera sesión. En las dos siguientes sesiones de prueba, ningún participante pidió ayuda. Para los participantes de la prueba, fue la primera vez que utilizaron una herramienta de software similar al prototipo desarrollado, por lo que es razonable que para algunos de ellos sea necesario algún tipo de ayuda. Pero, se resalta que la ayuda fue solicitada para las dos primeras actividades y solo en la primera sesión, lo que permite concluir que los diseños de las interfaces de usuario son fáciles de aprender a utilizar.

Se puede observar en la Figura 3.20 que la curva de tiempo promedio de cada actividad decrece a medida que avanza el número de sesión de prueba. Esto significa, que los participantes cada vez que utilizaron la interfaz de administración realizaron las actividades en menor tiempo. También, se puede observar en la Figura 3.20, que en la segunda vez que los participantes utilizan la interfaz (sesión de prueba No.2) ya mejoran considerablemente su rendimiento (menor tiempo para realizar las actividades); en la sesión de prueba No.3, los participantes nuevamente realizan las actividades en menor tiempo. Como las interfaces de usuario permiten mejorar el rendimiento del participante en poco tiempo (tres sesiones de prueba) se puede concluir que son eficientes y fáciles de aprender a utilizar.

3.3.3.4.2 Satisfacción

Cuestionario para medir el nivel de satisfacción

En la primera sesión, después de terminar las actividades, se pidió a los participantes que respondan las preguntas que se presentan en la Tabla 3.24 para conocer el nivel de satisfacción que experimentaron al utilizar la interfaz de administración.

¿Qué tan satisfecho se sintió con el uso de este sistema? Nada satisfecho 1 2 3 4 5 Muy satisfecho
¿Qué tan probable es que usted pueda recomendar este sistema a un familiar o conocido? Nada probable 1 2 3 4 5 Muy probable

Tabla 3.24 Preguntas para medir el nivel de satisfacción

Las respuestas a las preguntas de la Tabla 3.24 indicaron que casi todos (4 de 5) los participantes se sintieron muy satisfechos al utilizar el sistema. Por otro lado, es muy probable que todos los participantes recomienden el sistema a un familiar o conocido.

Análisis de resultados

Los participantes sintieron satisfacción al utilizar la interfaz de administración del prototipo. A pesar que el nivel de satisfacción es una medida subjetiva, tiene relación con las métricas objetivas (e. g., tiempo), ya que el diseño de una interfaz que es más fácil de utilizar tiende a ser más preferido por los usuarios [48].

3.3.4 PRUEBA DE CARGA

Una prueba de carga fue realizada para medir la cantidad de memoria RAM y procesamiento del hardware del prototipo, así como medir el tiempo de carga de páginas web cuando varios usuarios acceden a la Web de forma concurrente. Cuando se realizó esta prueba todos los componentes del prototipo ya fueron implementados e instalados en la plataforma de cómputo descrita en la Tabla **3.1** Características de la plataforma del prototipo

. Los resultados de la prueba permitieron determinar si la capacidad del hardware del modelo de Raspberry Pi utilizado es suficiente para la solución presentada.

3.3.4.1 Metodología

La prueba consistió en incrementar paulatinamente la carga sobre el prototipo, aumentando el número de usuarios (simulados) concurrentes desde 1 hasta 6, obteniendo en total 6 experimentos. Para simular el comportamiento de navegación de los usuarios en cada experimento, que duró 5 minutos, se programan instancias que representan a cada usuario y que generan peticiones a un conjunto predeterminado de páginas web. Una nueva petición web es generada por cada instancia cada 30 segundos.

La infraestructura de red que fue utilizada fue la red de un hogar con acceso a Internet, cuyo servicio de Internet contratado cuenta con un ancho de banda de 2.5 Mbps de bajada y compartición de 8:1. En la Figura 3.21 se muestra la topología de red implementada para realizar las pruebas de carga.

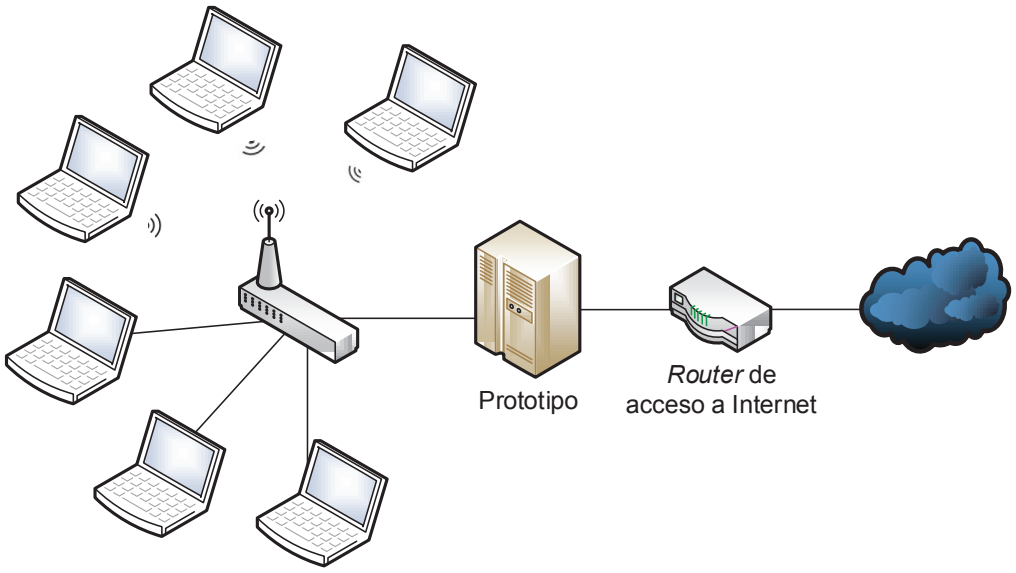

Figura 3.21 Topología de red en la prueba de carga

Para medir la cantidad de memoria RAM y procesamiento se utilizó el software de monitoreo de red PRTG. Para medir el tiempo de carga de una página web se utilizó el Add-on app.telemetry Page Speed Monitor para el navegador web Firefox que fue utilizado para ejecutar el programa generador de carga en cada *host* (computador portátil) de la red interna.

Como se utiliza un mismo conjunto de páginas web en todas las instancias programadas (usuarios simulados), se deshabilitó el almacenamiento en *cache* en Squid, y lo mismo fue realizado en el navegador web Firefox. Así, se asegura que Squid procese todas las peticiones HTTP generadas por los usuarios simulados.

La configuración del prototipo y las herramientas de software utilizadas para el desarrollo de la prueba de carga se encuentran en el Anexo G.

3.3.4.2 Programa generador de carga

En el navegador web de cada *host* de la red interna, se ejecutó un programa de JavaScript embebido en una página web que genera una petición HTTP para acceder a una página web cada 30 segundos. Se asumió que un usuario de un entorno doméstico genera una petición HTTP para acceder a una página web y visualiza su contenido, mientras se carga o una vez completa, durante 30 segundos.

Las páginas web que fueron solicitadas durante la prueba fueron un grupo de las páginas web más visitadas en Ecuador de acuerdo a la clasificación de sitios web más visitados por país que entrega Alexa Inc. [49].

3.3.4.3 Resultados

Procesamiento y tiempo promedio de carga de páginas web

En la Figura 3.22 se muestra una representación gráfica de las mediciones de procesamiento y tiempo promedio de carga de páginas web en función del número de usuarios concurrentes.

El tiempo de carga de páginas web fue medido en el *host* con el que se generó el primer incremento de carga. En cada incremento de carga se cargaron las mismas 10 páginas web, y en los cuales se midió el tiempo de carga de este grupo de páginas web.

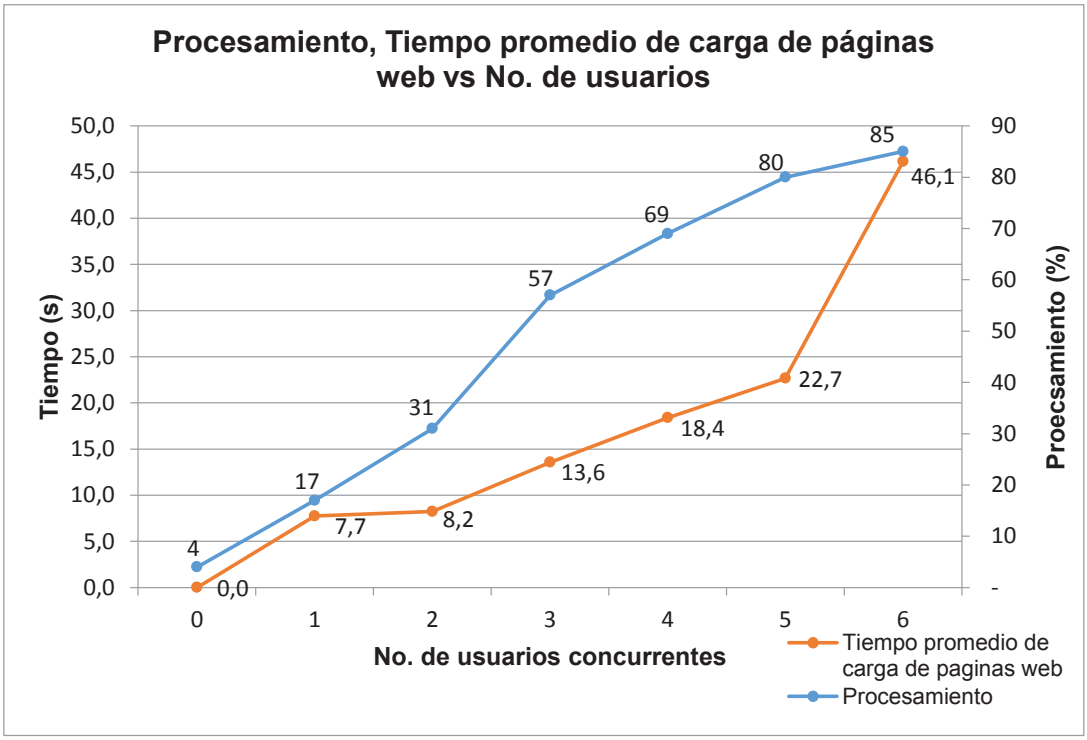

Figura 3.22 Procesamiento, Tiempo de carga de páginas web vs No. de usuarios

Memoria RAM

En la Figura 3.23 se muestra una representación gráfica de la cantidad promedio de RAM utilizada en función del número de usuarios concurrentes.

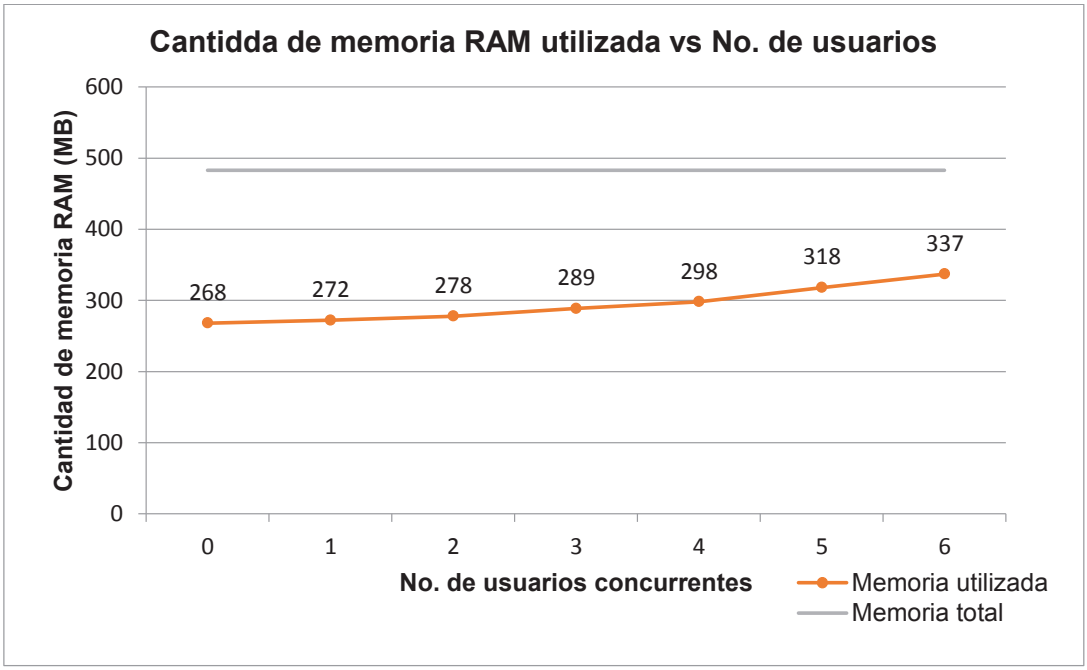

Figura 3.23 Cantidad de memoria RAM utilizada vs No. de usuarios

Análisis de resultados

De acuerdo a las mediciones de procesamiento y RAM se puede concluir que un computador Raspberry Pi modelo B+ utilizado en esta solución soporta la navegación web concurrente de 6 usuarios.

Por otro lado, cuando se generó la carga de 5 y 6 usuarios, el tiempo de carga de páginas web se incrementó considerablemente, lo que significa para el usuario una navegación web lenta. Por eso, se puede determinar que para que el usuario experimente una navegación web fluida, utilizando el prototipo, no se debería superar un nivel de concurrencia de 4 usuarios.

Cabe mencionar que en esta prueba de carga no se configuró almacenamiento en *cache* en Squid ni en el navegador web de cada *host*, por lo que se puede asumir, que al habilitar esta funcionalidad, el rendimiento de la plataforma de computó del prototipo puede mejorar.

3.4 COSTO REFERENCIAL DE DESARROLLO

En el costo referencial se considera el costo de los elementos de hardware y el costo del tiempo dedicado al desarrollo del prototipo. En la Tabla 3.25 se detalla el total de este del costo referencial.

Cant.	Elemento	Características	Costo Unidad	Total (USD)
1	Raspberry Pi modelo B+	Ver Tabla 1.1	65	65
1	Cargador con cable micro USB	Salida: 5 V, 1000 mA	10	10
1	Memoria micro SD	Clase 10 Capacidad de 16 GB	10	15
1	Adaptador USB a Ethernet	Velocidad hasta 100 Mbps	10	10
1	*Router* multifuncional D-Link, modelo N 150	IEEE 802.11b/g, IEEE 802.3 4 interfaces LAN 10/100 Mbps	30	30
2	*Patch Cord*	Cable UTP categoría 5e	5	10
392	Horas de desarrollo	Tiempo dedicado al desarrollo del prototipo	5	1960
			Total (USD)	2100

Tabla 3.25 Costo referencial de desarrollo del prototipo

De acuerdo al plan de entregas que se muestra en la Tabla 2.30, se estimó 49 días (8 horas cada día) para desarrollar todas las historias de usuario; a partir de esta estimación se calculó el número total de horas de desarrollo.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Se desarrolló un prototipo de un sistema de protección perimetral y navegación segura que puede ser instalado en cualquier entorno doméstico de red con acceso a Internet, y le permite al jefe del hogar, de forma sencilla, aplicar un control de acceso web personalizado: bloquear el acceso a sitios web, establecer un horario de acceso y bloquear el acceso completamente a cada *host* de la red; aplicar un control básico del tráfico a nivel de capa de red; y visualizar información de navegación web: intentos de acceso web bloqueados, tiempo de navegación web, sitios web más visitados y páginas web visitadas.

- Se evaluó el rendimiento del hardware del prototipo, mediante una prueba de carga, en la cual se simuló la generación continua de tráfico web de varios usuarios al mismo tiempo. Los resultados permitieron determinar que el Raspberry Pi modelo B+, utilizado en la solución, soporta el acceso web concurrente de 6 usuarios, alcanzo un nivel de procesamiento máximo de 85% de su capacidad; además, se observó que, con este nivel de concurrencia, el tiempo de carga de las páginas web se incrementa considerablemente respecto al tiempo que se tiene cuando navegan 4 usuarios, el incrementó del tiempo en promedio fue de 18.4 s a 46.1 s. Por ello, se determina que el prototipo puede soportar correctamente una navegación web de máximo 4 usuarios al mismo tiempo.

- Se evaluó la usabilidad de la interfaz de administración mediante una prueba de usuario, los participantes fueron un grupo de personas que representaron a la audiencia objetivo del prototipo, jefes de hogar, y los resultados obtenidos permitieron determinar que las interfaces de usuario son fácil de aprender a utilizar y eficientes: cada vez que los participantes

utilizaron la interfaz de administración, en cada sesión de prueba ejecutada, configuraron las funcionalidades del prototipo en menor tiempo.

- El desarrollo de aplicaciones para uso doméstico implica grandes retos que tienen que ver con la usabilidad y el costo de los sistemas. Un mayor esfuerzo en ofrecer usabilidad seguramente incrementará el costo de la solución, así como los recursos de hardware necesarios para servirla.

- Un Raspberry Pi modelo B+ ofrece recursos suficientes para soluciones telemáticas, como el prototipo desarrollo, donde el número de usuarios concurrentes es reducido. Sin embargo, ciertas funcionalidades (como la de listas negras/bloqueo) deben simplificarse significativamente, por la gran cantidad de recursos que pueden llegar a requerir.

- Se configuró almacenamiento en *cache* en Squid para almacenar recursos web solicitados con frecuencia, y que peticiones a estos recursos sean servidos directamente por Squid, con el propósito de mejorar el tiempo de carga de las páginas web y en general mejorar la experiencia del usuario en la navegación web.

- Se utilizaron herramientas de software libre para desarrollar cada componente del prototipo. Esto no solo evitó el pago de licencias y la correspondiente disminución del costo de desarrollo, sino que también ayudó en la solución de problemas que se presentaron en la implementación, por la documentación de calidad que existe y que es mantenida por usuarios avanzados que promueven el software libre.

- Squid es una herramienta efectiva y fácil de configurar para aplicar un control de acceso web personalizado, así como para registrar información detallada sobre el acceso web. Esta herramienta permitió implementar con éxito la mayoría de los requerimientos del prototipo. Además, dada su amplia versatilidad puede ser utilizada en entornos más complejos que un entorno doméstico.

- El *framework* web Django es una herramienta muy poderosa para desarrollar aplicaciones y sitios web en poco tiempo. Esto es debido a su patrón de patrón de arquitectura Modelo-Vista-Plantilla que permite mantener el código ordenado, reutilizar aplicaciones para evitar desarrollar nuevamente una misma funcionalidad y el lenguaje de programación Python. En este proyecto, la simplicidad de la sintaxis de Python permitió una codificación rápida y el ahorro de gran número de líneas de código; además la gran variedad de librerías de Python facilitaron la lectura y escritura de los archivos de configuración de los componentes del prototipo, así como la ejecución de las aplicaciones que los implementan.

- El método heurístico diseñado para calcular el tiempo de navegación web, en base a la información de las entradas del archivo de *log* de acceso de squid es de gran utilidad: permitió cumplir con requerimientos del prototipo y además permite concluir que en una solución perimetral se debe desarrollar aplicaciones clientes para complementar la recolección de información sobre la navegación web realizada. En el prototipo se puede mejorar el cálculo del tiempo de navegación web realizado, en base solo a eventos que generan peticiones HTTP, desarrollando una aplicación cliente (para el navegador web) que se encargue de determinar el tiempo que el usuario visualiza la información servida (sin generar peticiones HTTP).

- En el prototipo no se intermedian peticiones HTTPS; la aplicación Squid simplemente deja establecer o bloquear el túnel de extremo a extremo SSL o TLS, por lo que no puede revisar el contenido de las peticiones que atraviesan el túnel y no registra en el archivo de *log* de acceso cierta información de las peticiones HTTPS procesadas, como la URL de cada petición. Esto, unido a que cada vez más sitios web utilizan HTTPS, limita la capacidad de Squid para recolectar información de navegación web más detallada.

- Para aplicar el control de acceso web personalizado del prototipo es necesario configurar manualmente cada navegador web del *host* cliente

con la dirección IP y puerto del servidor proxy HTTP. Este proceso puede ser tedioso e incluso complicado para un usuario de un entorno doméstico, pero es necesario para redirigir las peticiones HTTPS al servidor proxy HTTP, porque este tipo de peticiones no pueden ser intermediadas de forma transparente.

- Utilizar la metodología de desarrollo de software Extreme Programming ayudó a organizar el proceso de desarrollo del prototipo y proporcionó las pautas para saber qué actividades realizar para que se consiga un producto de software de calidad en un corto período.

- No demorar en el diseño de una interfaz de usuario. Realizar prototipos, en papel o con herramientas de software, incluso implementarlos y probarlos continuamente con usuarios finales para encontrar problemas de usabilidad y corregirlos a tiempo para obtener un diseño de alta calidad.

- Para abordar la usabilidad en un sistema se debe considerar las características de la audiencia objetivo y el contexto de utilización.

4.2 RECOMENDACIONES

- Evaluar la usabilidad del sistema utilizando otras técnicas, como grupos focales, comparación de versiones de interfaces en una prueba A/B, etc.; además, utilizar herramientas de *click tracking* para recolección y análisis de información.

- Desarrollar una aplicación cliente para automatizar el registro de dispositivos en el sistema del prototipo y la configuración del navegador web.

- Desarrollar un *plugin* para el navegador web del *host* cliente para que se encargue de calcular el tiempo que el usuario observa la información servida, y así complementar el cálculo de tiempo de navegación web realizado en este proyecto.

- Desarrollar un mecanismo para actualizar periódicamente el contenido de las listas negras que se manejan en el prototipo.

- Eliminar aplicaciones que vienen instaladas en Raspbian y que no son utilizadas para aumentar la disponibilidad de recursos para las funcionalidades del prototipo.

- El gran potencial del *framework* web Django implica un gran consumo de recursos, por lo que se recomienda utilizar una alternativa más ligera cuando se utilice hardware de limitada capacidad.

- Utilizar un método complementario a la encuesta para recopilar requerimientos, e. g., entrevistas, casos de uso, prototipos. Una encuesta permite recolectar información de un gran número de personas, pero la información obtenida no puede ser verificada y se pueden obviar detalles esenciales de los requerimientos.

- Desarrollar un libreto para conducir la prueba de usuario para la evaluación de usabilidad. Esto ayuda a evitar problemas y permite dar las mismas instrucciones a cada participante de la prueba.

REFERENCIAS BIBLIOGRÁFICAS

[1] "Políticas Públicas aplicadas en Telecomunicaciones permiten bajar costo de Internet." [Online]. Available: http://www.telecomunicaciones.gob.ec/politicas-publicas-aplicadas-en-telecomunicaciones-permiten-bajar-costo-de-internet/. [Accessed: 21-Oct-2015].

[2] "Plan Nacional de Desarrollo de Banda Ancha." [Online]. Available: http://www.telecomunicaciones.gob.ec/plan-nacional-de-desarrollo-de-banda-ancha/. [Accessed: 21-Oct-2015].

[3] C. Semeria, "Internet Firewalls and Security," *3Com Techinical Papers*. U. S. A., 1996.

[4] "What is a proxy server?," *Knowledge Base*. [Online]. Available: https://kb.iu.edu/d/ahoo. [Accessed: 28-Oct-2015].

[5] "netfilter/iptables project homepage - The netfilter.org project." [Online]. Available: http://www.netfilter.org/index.html. [Accessed: 13-Aug-2015].

[6] D. Wessels, *Squid: the definitive guide*. U. S. A., 2004.

[7] "What is DansGuardian?" [Online]. Available: http://dansguardian.org/?page=whatisdg. [Accessed: 22-Oct-2015].

[8] "ClearOS – OS for your Server, Network, and Gateway Systems." [Online]. Available: https://www.clearos.com/. [Accessed: 23-Oct-2015].

[9] "DistroWatch.com: ClearOS." [Online]. Available: http://distrowatch.com/table.php?distribution=clearos. [Accessed: 23-Oct-2015].

[10] "What is a Raspberry Pi," 2015. [Online]. Available: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/. [Accessed: 30-Jul-2015].

[11] "Raspberry Pi Firewall and Intrusion Detection System." [Online]. Available: http://www.instructables.com/id/Raspberry-Pi-Firewall-and-Intrusion-

Detection-Syst/. [Accessed: 22-Oct-2015].

[12] “Open Source Media Center.” [Online]. Available: https://osmc.tv/. [Accessed: 22-Oct-2015].

[13] “Raspberry Pi - SMS Garage Door Butler.” [Online]. Available: http://www.instructables.com/id/Raspberry-Pi-SMS-Garage-Door-Butler/. [Accessed: 22-Oct-2015].

[14] INTECO, “Ingeniería del Software: Metodologías y Ciclos de Vida,” España, 2009.

[15] “The New Methodology.” [Online]. Available: http://www.martinfowler.com/articles/newMethodology.html. [Accessed: 25-Oct-2015].

[16] K. Beck, *Extreme Programming Explained: Embrace Change*, no. c. 1999.

[17] “Extreme Programming (XP) FAQ.” [Online]. Available: http://www.jera.com/techinfo/xpfaq.html. [Accessed: 13-Aug-2015].

[18] M. Cohn, *User Stories Applied: For Agile Software Development*, vol. 1. 2004.

[19] “Build your Project using Extreme Programming.” [Online]. Available: http://www.asapm.org/articles/A2_AboutXP.pdf. [Accessed: 06-Nov-2015].

[20] “BeckDesignRules.” [Online]. Available: http://www.martinfowler.com/bliki/BeckDesignRules.html. [Accessed: 21-Oct-2015].

[21] “Usability 101: Introduction to Usability.” [Online]. Available: http://www.nngroup.com/articles/usability-101-introduction-to-usability/. [Accessed: 11-Aug-2015].

[22] T. Grossman, G. Fitzmaurice, and R. Attar, “A survey of software learnability,” *Proc. 27th Int. Conf. Hum. factors Comput. Syst. - CHI 09*, pp. 649–658, 2009.

[23] M. A. Khan and M. Nasir, *Human Errors and Learnability Evaluation of Authentication System*, no. September. 2011.

[24] “Usability Evaluation Methods | Usability Body of Knowledge.” [Online]. Available: http://www.usabilitybok.org/usability-evaluation-methods. [Accessed: 21-Oct-2015].

[25] H. Petrie and N. Bevan, “Petrie_ Bevan_The_evaluation_of_accessibility_usability_and_user_experience,” 2009.

[26] “Recruiting Test Participants for Usability Studies.” [Online]. Available: http://www.nngroup.com/articles/recruiting-test-participants-for-usability-studies/. [Accessed: 11-Aug-2015].

[27] “Task Scenarios for Usability Testing.” [Online]. Available: http://www.nngroup.com/articles/task-scenarios-usability-testing/. [Accessed: 11-Aug-2015].

[28] “Resultados provinciales Pichincha.” [Online]. Available: http://www.ecuadorencifras.gob.ec/wp-content/descargas/Manu-lateral/Resultados-provinciales/pichincha.pdf. [Accessed: 11-Aug-2015].

[29] “¿Qué tamaño de muestra necesito? | Blog de Netquest.” [Online]. Available: http://www.netquest.com/blog/es/que-tamano-de-muestra-necesito/. [Accessed: 11-Aug-2015].

[30] “Usability guidelines.” [Online]. Available: http://guidelines.usability.gov/. [Accessed: 26-Nov-2015].

[31] “10 Heuristics for User Interface Design: Article by Jakob Nielsen.” [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/. [Accessed: 13-May-2016].

[32] “Raspbian.” [Online]. Available: https://www.raspbian.org/. [Accessed: 12-Jan-2016].

[33] “Dnsmasq.” [Online]. Available: https://help.ubuntu.com/community/Dnsmasq. [Accessed: 19-Jan-2016].

[34] “Access Controls in Squid.” [Online]. Available: http://wiki.squid-cache.org/SquidFaq/SquidAcl. [Accessed: 28-Jan-2016].

[35] “squid : logformat configuration directive.” [Online]. Available:

http://www.squid-cache.org/Doc/config/logformat/. [Accessed: 21-Oct-2015].

[36] “Squid.” [Online]. Available: http://www.alcancelibre.org/staticpages/index.php/19-0-como-squid-general. [Accessed: 29-Jul-2015].

[37] “Squidblacklist.org - Domain Blacklist Database - Web Filtering For Squid Proxy & More.” [Online]. Available: http://www.squidblacklist.org/. [Accessed: 21-Feb-2016].

[38] “Shalla Secure Services KG.” [Online]. Available: http://www.shallalist.de/. [Accessed: 21-Feb-2016].

[39] “URLBlacklist.com.” [Online]. Available: http://urlblacklist.com/?sec=download. [Accessed: 21-Feb-2016].

[40] “Alexa - Top Sites by Category: Top.” [Online]. Available: http://www.alexa.com/topsites/category. [Accessed: 04-Feb-2016].

[41] “Python Advocacy HOWTO — Python v2.7.4 documentation.” [Online]. Available: https://docs.python.org/2/howto/advocacy.html. [Accessed: 26-Oct-2015].

[42] “The Web framework for perfectionists with deadlines | Django.” [Online]. Available: https://www.djangoproject.com/. [Accessed: 02-Aug-2015].

[43] “Chapter 5: Models.” [Online]. Available: http://www.djangobook.com/en/2.0/chapter05.html. [Accessed: 01-Oct-2015].

[44] “Features Of SQLite.” [Online]. Available: https://www.sqlite.org/features.html. [Accessed: 03-Oct-2015].

[45] “Appropriate Uses For SQLite.” [Online]. Available: https://www.sqlite.org/whentouse.html. [Accessed: 14-Aug-2015].

[46] J. Ibrobo, C. Tipán, and T. Calle, *Desarrollo de una aplicación móvil bajo la plataforma iOS para la consulta de un catálogo de productos de una tienda deportiva*. Quito, Ecuador, 2015.

[47] “Why You Only Need to Test with 5 Users.” [Online]. Available:

http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/. [Accessed: 07-Aug-2015].

[48] “User Satisfaction vs. Performance Metrics.” [Online]. Available: https://www.nngroup.com/articles/satisfaction-vs-performance-metrics/. [Accessed: 12-Sep-2015].

[49] “Alexa - Top Sites in Ecuador.” [Online]. Available: http://www.alexa.com/topsites/countries/EC. [Accessed: 19-Dec-2015].

[50] “Platform for Internet Content Selection (PICS)Platform for Internet Content Selection (PICS).” [Online]. Available: http://www.w3.org/PICS/. [Accessed: 22-Oct-2015].

[51] “Unix Time Stamp.” [Online]. Available: http://www.unixtimestamp.com/. [Accessed: 24-Mar-2016].

[52] “WebFrameworks - Python Wiki.” [Online]. Available: https://wiki.python.org/moin/WebFrameworks/. [Accessed: 27-Sep-2015].

[53] N. Sharma, L. Perniu, R. F. Chong, A. Iyer, C. Nandan, A. Mitea, M. Nonvinkere, and M. Danubianu, “Database Fundamentals DB2,” *Group*, p. 282, 2010.

ANEXOS

ANEXO A: ENCUESTA PARA RECOLECCIÓN DE REQUERIMIENTOS

ANEXO B: INSTALACIÓN Y CONFIGURACIÓN DE RASPBIAN

ANEXO C: CÓDIGO DEL PROGRAMA RECOLECTOR DE INFORMACIÓN DE NAVEGACIÓN WEB

ANEXO D: CÓDIGO DE LA INTERFAZ DE ADMINISTRACIÓN

ANEXO E: SCRIPT DE CONFIGURACIÓN INICIAL DE IPTABLES

ANEXO F: ARCHIVOS PARA CONTROL DE ACCESO EN SQUID

ANEXO G: INSTALACIÓN Y CONFIGURACIÓN DE HERRAMIENTAS PARA LA PRUEBA DE CARGA

ANEXO H: MANUAL DE USUARIO DE IPTABLES

ANEXO I: INSTALACIÓN DEL PROTOTIPO EN UNA RED DOMÉSTICA

Los Anexos se encuentran en el CD adjunto.