Aplicación web construida con el framework Laravel integrada a routers MikroTik para administrar clientes, corte de ventas y generar fichas hotspot

Web application built with the Laravel framework integrated with MikroTik routers to manage clients, sales report, and generating hotspot vouchers

Arturo Cuevas Aldana
Tecnológico Nacional de México, Instituto Tecnológico Superior de Huauchinango, México
m23315016@huauchinango.tecnm.mx
ORCID: 0000-0002-9219-3406

Carmen Jeannette Sampayo Rodríguez
Tecnológico Nacional de México, Instituto Tecnológico Superior de Huauchinango, México
carmen.sr@huauchinango.tecnm.mx ✉
ORCID: 0000-0001-8844-6055

Aldo Hernández Luna
Tecnológico Nacional de México, Instituto Tecnológico Superior de Huauchinango, México
aldo.hl@huauchinango.tecnm.mx
ORCID: 0000-0002-7717-5314

Hugo Hernández Cabrera
Tecnológico Nacional de México, Instituto Tecnológico Superior de Huauchinango, México
hugo.hc@huauchinango.tecnm.mx
ORCID: 0000-0002-7172-9734

https://doi.org/10.36825/RITI.13.30.004

Recibido: Junio 09, 2025
Aceptado: Septiembre 18, 2025

Resumen: El uso de redes *hotspot* para ofrecer acceso controlado a Internet es común en entornos públicos y comunitarios. Sin embargo, su administración suele implicar procesos manuales que dificultan el control eficiente de usuarios, cortes de servicio y generación de fichas. Este estudio propone que el uso de una metodología ágil, como Scrum, junto con herramientas web modernas, permite desarrollar una aplicación efectiva para automatizar estos procesos. Para ello, se aplicó Scrum como marco de trabajo de desarrollo, dividiendo las funcionalidades en *sprints* planificados a partir de historias de usuario. El sistema fue construido con Laravel, Livewire, MySQL y otras tecnologías web, integrando funcionalidades específicas para gestionar clientes, fichas *hotspot*, cortes y máquinas expendedoras conectadas mediante NodeMCU y routers MikroTik. Las pruebas funcionales, de integración y de rendimiento confirmaron la estabilidad, eficiencia y adecuación del sistema a los objetivos planteados. Se observó una mejora operativa de hasta el 96 % en tiempo de ejecución de tareas clave y una

© RITI. CC BY-NC

reducción significativa del error humano. Se concluye que la combinación de desarrollo ágil y automatización especializada ofrece una solución efectiva y adaptable para la administración técnica de redes *hotspot*.

Palabras clave: *Hotspot, Scrum, MikroTik, Aplicación Web, Automatización de Redes*.

Abstract: The use of hotspot networks to provide controlled internet access is common in public and community environments. However, their administration often involves manual processes that hinder efficient user management, sales reporting, and voucher generation. This study demonstrates how applying an agile methodology such as Scrum, combined with modern web tools, enables the development of an effective application to automate these processes. Scrum was adopted as the development framework, organizing functionality into planned sprints based on user stories. The system was built using Laravel, Livewire, MySQL, and other web technologies, integrating features for managing clients, hotspot vouchers, sales reports, and vending machines connected via NodeMCU and MikroTik routers. Functional, integration, and performance testing confirmed the system's stability, efficiency, and alignment with project goals. Operational time improvements of up to 96% were observed in key tasks, along with a significant reduction in human error. These findings suggest that the combination of agile development and specialized automation provides an effective and adaptable solution for the technical administration of hotspot networks.

Keywords: *Hotspot, Scrum, MikroTik, Web Application, Network Automation*.

1. Introducción

Informes de Statista destacan que, a nivel mundial, cerca de dos tercios de la población mundial está conectada a Internet [1]. En México, hasta febrero de 2025 se estimaba un total de 110 millones de usuarios conectados, solo por debajo de Brasil en las estadísticas de América Latina [2].

Específicamente hablando de redes inalámbricas, esta situación impulsa el uso de sistemas *hotspot* como una solución efectiva para ofrecer acceso a Internet en entornos públicos y comerciales con un control robusto y eficiente. Sin embargo, la administración de estos servicios aún suele depender de configuraciones manuales o herramientas dispersas que dificultan acciones esenciales como el control eficiente de usuarios, cortes de servicio y generación de accesos. Esta situación se agrava cuando se requiere integrar diferentes dispositivos de red, como *routers* MikroTik, con aplicaciones que gestionan usuarios y ventas desde máquinas vending u otros puntos automatizados.

En este contexto, el presente artículo documenta el proceso de desarrollo de una aplicación web orientada a la administración de redes *hotspot*, con funcionalidades específicas como el registro de clientes, gestión de credenciales de acceso, cortes, reportes y exportación de datos, todo ello integrado con *routers* MikroTik y máquinas expendedoras de credenciales de acceso. El sistema fue desarrollado aplicando la metodología ágil Scrum, la cual permitió organizar el trabajo en *sprints* iterativos enfocados en la entrega progresiva de funcionalidades. Esta metodología no solo facilita una mayor organización entre los desarrolladores, sino que también permite obtener resultados en corto plazo mediante entregas parciales [3].

El propósito de este trabajo fue construir una solución web que facilitara la administración técnica y operativa de redes *hotspot* desde una plataforma centralizada, optimizando el tiempo de operación, reduciendo errores manuales y mejorando la trazabilidad de los procesos. Para validar la funcionalidad y rendimiento del sistema, se implementaron pruebas manuales basadas en las historias de usuario, así como pruebas de integración con los dispositivos externos y análisis de desempeño mediante la herramienta Clockwork.

Este artículo se organiza de la siguiente manera: la Sección 2 presenta el estado del arte sobre sistemas de gestión *hotspot*, aplicaciones web administrativas e integración con MikroTik; la Sección 3 describe los materiales y métodos aplicados durante el desarrollo; la Sección 4 detalla los resultados obtenidos por sprint; finalmente, la Sección 5 discute las conclusiones derivadas del trabajo y las posibles líneas de mejora.

2. Estado del arte

En el ámbito del desarrollo de software, la metodología ágil Scrum se ha consolidado como un marco de trabajo ampliamente adoptado con el fin de optimizar la productividad y la eficiencia de los equipos de desarrollo de

© RITI. CC BY-NC

proyectos. Fue creado por Jeff Sutherland y Ken Schwaber en 1993 como una respuesta a las limitaciones del método en cascada. Scrum se basa en ciclos cortos llamados *sprints*, en los cuales los equipos organizan una serie de tareas según su prioridad y trabajan de manera colaborativa para presentar avances frecuentes para así obtener retroalimentación que beneficie el resto del desarrollo. Su éxito radica en la capacidad de inspeccionar y ajustar constantemente los procesos, asegurando que los productos finales respondan a las necesidades reales de los usuarios y clientes [4].

Por su parte, las redes *hotspot* o puntos de acceso, constituyen una modalidad de acceso a Internet en la que los usuarios pueden conectarse con sus dispositivos generalmente mediante una red inalámbrica autenticándose a través de un portal cautivo con un usuario y contraseña proporcionados según algún modelo de negocio [5]. Estas credenciales de acceso se suelen distribuir como tarjetas impresas o digitales conocidas popularmente como fichas *hotspot*. La conexión en este tipo de servicios suele ser medida, controlando factores como el ancho de banda, los sitios a los que se tiene acceso y la duración de la conexión.

MikroTik, es una compañía fundada en 1996 enfocada en las telecomunicaciones. Ofrece su línea de *routers* con el sistema operativo RouterOS que se encarga de administrar las configuraciones que el *router* puede ofrecer para la conexión de los usuarios [6]. Entre estas configuraciones, existe la capacidad de crear servidores *hotspot* administrables por estos dispositivos y fáciles de implementar en un sentido técnico.

La administración de redes *hotspot*, por tanto, implica una serie de tareas técnicas y operativas orientadas al control de usuarios, distribución de fichas de acceso, y monitoreo del servicio brindado, tanto en esquemas temporales como residenciales. La gestión de clientes requiere registrar usuarios finales o puntos de venta, asignar identificadores únicos, y mantener un historial de servicios, ventas y pagos. Particularmente, la generación de fichas *hotspot* demanda la creación de credenciales personalizadas con reglas estrictas de nomenclatura y seguridad, las cuales deben exportarse e ingresarse manualmente en dispositivos MikroTik mediante comandos específicos.

Estas operaciones, junto con el cálculo de cortes de fichas, control de comisiones y seguimiento de clientes residenciales, suelen realizarse de forma manual con herramientas no integradas, como hojas de cálculo y configuraciones directas en WinBox. Esta situación genera una alta carga operativa, propensa a errores, difícil de escalar, y con escasa trazabilidad. Por ello, se identifica la necesidad de sistemas que automaticen estos procesos, centralicen la información, y proporcionen mecanismos de control eficientes que mejoren la toma de decisiones, reduzcan el esfuerzo administrativo y aseguren la calidad del servicio ofrecido.

Diversas investigaciones han abordado la gestión de redes *hotspot* y la integración de herramientas de software para su administración. En el ámbito académico, Castillo Estrada *et al.* [7] desarrollaron una aplicación web para la gestión del Curriculum Vitae del personal docente, utilizando una arquitectura orientada a servicios y tecnologías web como Vue.js, CodeIgniter y API REST. Si bien el dominio de aplicación es diferente, este antecedente resulta relevante debido a su enfoque de diseño modular, integración de interfaces dinámicas y uso de metodologías estructuradas de desarrollo, elementos que también están presentes en el proyecto aquí descrito. Además, comparten la necesidad de exponer datos mediante una API para ser consumidos por otros sistemas o dispositivos externos, lo cual guarda relación con la integración que este trabajo realiza con máquinas expendedoras basadas en NodeMCU ESP8266.

MikroTik ofrece a sus usuarios una aplicación de escritorio para el sistema operativo Windows llamada Winbox. Esta aplicación es ampliamente usada y es la oficial para modificar todas las configuraciones de los *router* con el sistema operativo RouterOS, no solo las relacionadas con el *hotspot*. Permite el acceso a dispositivos tanto en red local como aquellos configurados para el acceso remoto [8]. Esta aplicación presenta limitaciones para la automatización de ciertas funciones administrativas avanzadas del *hotspot*, lo que obliga a los administradores a realizar procesos de forma manual o a complementar con otros sistemas.

Frente a estas limitaciones, Purwanto y Amin [9] propusieron un sistema de autenticación con código QR para redes *hotspot* basado en las propias capacidades del propio RouterOS para administrar usuarios y complementado el portal de inicio de sesión con HTML5 y JavaScript. Su implementación resolvió el problema de acceso mediante credenciales tradicionales, pero no contempló una gestión integral de usuarios ni funciones administrativas avanzadas.

Por otra parte, Amarudin [10] implementó el plugin MikroTik User Manager junto con herramientas como FreeRadius y MySQL, con el objetivo de autenticar, autorizar y contabilizar (AAA) el uso de redes *hotspot* en entornos públicos. La creación de cuentas controladas mejoró significativamente la experiencia del usuario, aunque el sistema requiere conocimientos técnicos avanzados para su puesta en marcha.

© RITI. CC BY-NC

Con un enfoque similar, Nainggolan y Putra [11] aplicaron User Manager en cibercafés, mostrando su utilidad para generar ingresos mediante vales *hotspot* personalizados. A pesar de su éxito, los investigadores advierten sobre la dificultad de escalar o adaptar la solución en otros entornos, debido a su complejidad operativa.

En contextos comunitarios, Hendrawan y Saputra [12] diseñaron una red RT/RW con claves generadas automáticamente mediante User Manager, facilitando la gestión para administradores y usuarios. No obstante, la cobertura técnica de la red fue limitada, restringiéndose a distancias cortas.

Suryanto [13] analizó el impacto de la implementación de User Manager en entornos institucionales, mejorando la calidad de servicio mediante políticas automatizadas de acceso y consumo de ancho de banda. Aun así, remarcó que la sostenibilidad de estas redes depende no solo del software, sino también del respaldo normativo de los proveedores del servicio.

En un enfoque más empresarial, Aguilar Alvarado *et al.* [14] desarrollaron un software para ISP basado en Java y PostgreSQL, con funcionalidades como control de banda, desconexiones automatizadas, ubicación GPS y reconexión de servicios. La herramienta demostró mejorar la productividad, la relación cliente-empresa y la toma de decisiones estratégicas, al integrar distintos procesos de gestión bajo una sola plataforma.

Hamid y Al Farizi [15] presentaron Mikhmon como una alternativa para comercializar acceso a redes privadas mediante vales con límites personalizados de tiempo y velocidad. La solución incluye autenticación vía código QR y una interfaz sencilla de administración basada en etapas estructuradas de análisis, diseño, simulación, implementación y monitoreo.

Finalmente, Rodríguez Ávila *et al.* [16] diseñaron e implementaron un sistema *hotspot* institucional con un *router* CCR-1036 y *access points* Ubiquiti. Utilizando portal cautivo personalizado, perfiles diferenciados y pruebas de rendimiento con herramientas especializadas, lograron validar una solución económicamente viable, con buen control de ancho de banda y monitoreo en tiempo real, aunque con retos al escalar para cargas de tráfico muy altas.

A continuación, se presenta la Tabla 1 con una comparación de las principales características de las investigaciones e implementaciones descritas anteriormente.

Tabla 1. Comparativa de soluciones administrativas.

Autor(es)	Tecnología principal	Funcionalidades destacadas	Limitaciones identificadas
MikroTik (Winbox) [8]	RouterOS, Winbox	Configuración completa de *routers*	Requiere configuración manual, sin automatización
Purwanto y Amin [9]	HTML5, JavaScript, RouterOS	Acceso vía código QR en portal cautivo	No gestiona usuarios ni administra fichas
Amarudin [10]	User Manager, FreeRadius, MySQL	AAA en redes *hotspot* públicas	Complejidad técnica alta
Nainggolan y Putra [11]	User Manager	Generación de vales *hotspot* en cibercafés	Difícil de escalar en otros contextos
Hendrawan y Saputra [12]	User Manager	Claves generadas automáticamente en redes RT/RW	Cobertura física limitada
Suryanto [13]	User Manager	Control de acceso y ancho de banda institucional	Dependencia de respaldo normativo
Aguilar Alvarado *et al.* [14]	Java, PostgreSQL	Software ISP con control de servicios y GPS	Complejidad de desarrollo e integración inicial
Hamid y Al Farizi [15]	Mikhmon	Vales *hotspot* con límites de tiempo y velocidad	No apto para escenarios de alta demanda
Rodríguez Ávila *et al.* [16]	MikroTik, Ubiquiti	Portal cautivo, perfiles diferenciados, pruebas QoS	Escalabilidad limitada bajo alta carga

Fuente: Elaboración propia.

La novedad principal de la aplicación desarrollada frente a las soluciones revisadas radica en su carácter integral: no se limita a la gestión de usuarios mediante *vouchers*, como ocurre en la mayoría de las alternativas, sino que combina en una misma plataforma web la administración de clientes *hotspot* y residenciales, la integración con máquinas expendedoras, y la generación de reportes y cortes de ventas. Además, incorpora un enfoque de diseño orientado a la usabilidad y la reducción de la complejidad técnica, lo que permite que pequeños y medianos

© RITI. CC BY-NC

proveedores de servicios de Internet adopten la herramienta sin requerir conocimientos avanzados de redes. Estas características distintivas marcan una diferencia sustancial con respecto a los sistemas existentes y constituyen el aporte original de la presente investigación.

3. Materiales y métodos

Para llevar a cabo el desarrollo de la aplicación web, se adoptó el marco de trabajo Scrum, una metodología ágil que permite la entrega incremental y colaborativa de software.

Se definieron los roles fundamentales de Scrum, teniendo a un experto del área en los roles de *Product Owner* y *Scrum Master*, garantizando una dirección clara del producto y facilitando el cumplimiento de los principios del marco ágil. En la Figura 1 se muestran las fases basadas en Scrum, utilizadas en esta investigación para desarrollar la aplicación.

Figura 1. Fases de la metodología de desarrollo.

3.1. Planificación, análisis y diseño de la aplicación

Las fases iniciales de análisis de requisitos y diseño de la aplicación se apoyaron en un enfoque de diseño centrado en el usuario. Esta metodología se integra perfectamente con Scrum permitiendo evaluar desde etapas tempranas la aceptabilidad de la aplicación por parte de los usuarios.

3.2. Planificación de Sprint

Las historias de usuario desempeñaron un papel central en la planificación de los *sprints*. Cada uno se basó directamente en desarrollar las historias de usuario distribuidas de forma eficiente según su prioridad y el esfuerzo requerido. A esta planificación se le conoce como Product Backlog. En la Tabla 2 a continuación, se describen los *sprints* definidos, junto con las historias de usuario asignadas a cada uno y las fechas de realización en que fueron programados.

Tabla 2. Historias de usuario para la aplicación web y el sprint en el que se planificó.

ID	Título	Prioridad
Sprint 1: Fundamentos del Sistema (03/06/2024 – 05/07/2024)		
HU-01	Registro de usuarios	Alta
HU-02	Iniciar sesión	Alta
HU-03	Modificar datos personales	Media
Sprint 2: Gestión de Clientes (08/07/2024 – 23/08/2024)		
HU-04	Registrar nuevos clientes	Alta
HU-05	Modificar clientes	Media
HU-06	Consultar y buscar clientes	Media
HU-07	Gestionar roles y permisos	Alta
HU-19	Gestión de configuraciones del sistema	Media
HU-22	Gestión de clientes residenciales	Alta
Sprint 3: Gestión de Fichas y Cortes (26/08/2024 – 04/10/2024)		
HU-08	Generar lotes de fichas *hotspot*	Alta

© RITI. CC BY-NC

HU-09	Administrar lotes de fichas *hotspot*	Alta
HU-10	Generar cortes de fichas *hotspot*	Media
Sprint 4: Integración con MikroTik (07/10/2024 – 15/11/2024)		
HU-13	Generar comandos para *routers* MikroTik	Media
HU-14	Exportar fichas hotspot a MikroTik	Alta
HU-15	Registrar y gestionar máquinas vending	Media
HU-16	Generar fichas *hotspot* para máquinas vending	Media
HU-17	Gestionar ventas de máquinas vending	Media
Sprint 5: Exportación y Reportes (18/11/2024 – 20/12/2024)		
HU-11	Exportar fichas hotspot a PDF	Media
HU-12	Exportar cortes de fichas a PDF o JPG	Baja
HU-18	Alertas y notificaciones	Baja
HU-23	Importación y exportación de datos del sistema	Media
Sprint 6: Seguridad y Mantenimiento (23/12/2024 – 24/01/2025)		
HU-20	Registro y gestión de logs	Media
HU-21	Eliminación segura de datos del sistema	Baja

Fuente: Elaboración propia.

3.3. Desarrollo de la aplicación web

Para el desarrollo del sistema, se utilizó el *framework* Laravel 9, basado en el lenguaje PHP 8.0.28, seleccionado por su robustez, amplia comunidad de soporte y por implementar el patrón de arquitectura Modelo–Vista–Controlador (MVC), lo que facilita la separación de responsabilidades y la escalabilidad de aplicaciones web modernas [17]. Se complementó con Livewire, que permitió mejorar la experiencia del usuario mediante componentes dinámicos sin necesidad de recargar la página, reduciendo la dependencia de código JavaScript adicional y manteniendo la lógica dentro del ecosistema de Laravel y Blade [18].

La interfaz gráfica se diseñó con Bootstrap, aprovechando sus componentes predefinidos y características responsivas para asegurar un correcto despliegue en dispositivos móviles y de escritorio [19]. Asimismo, se emplearon librerías como Select2 y SweetAlert2, que facilitaron la interacción avanzada con formularios, listas desplegables y mensajes emergentes, contribuyendo a una experiencia de usuario más fluida.

Para la gestión de datos se utilizó el sistema de gestión de bases de datos MySQL 5.7.4, ampliamente reconocido por su estabilidad, eficiencia en el manejo de grandes volúmenes de información y soporte a transacciones ACID [20]. La comunicación entre la aplicación y los dispositivos externos se implementó mediante API REST, lo que garantizó un intercambio de datos sin estado y con suficiente flexibilidad para integrar distintos clientes, como las máquinas expendedoras de fichas [21].

Para la exportación de credenciales de acceso *hotspot*, se trabajó con routers MikroTik de la línea RouterBoard con el sistema operativo RouterOS con una configuración de servidor hotspot previamente realizada. Además, se integró soporte para la gestión de máquinas vending, basadas en el módulo NodeMCU ESP8266, conectadas a través de red local.

El entorno de desarrollo elegido incluye el editor de código Visual Studio Code combinado con el entorno de desarrollo local Laragon. Para control de versiones se utilizó Git y Github como respaldo. Finalmente, para simular las peticiones de las máquinas expendedoras en el entorno de desarrollo se utilizó Postman.

Durante el desarrollo del sistema también se aplicaron varios patrones de diseño reconocidos para garantizar una mayor mantenibilidad y escalabilidad del sistema. Entre los patrones utilizados se encuentran: Singleton, para instanciar clases auxiliares y de formateo de datos disponibles desde diferentes partes de la aplicación ocupando un único espacio de memoria; Facade, para ocultar la complejidad de algunas clases y extraer únicamente las funcionalidades requeridas como la manipulación de imágenes o generación de archivos PDF; Strategy, para reutilizar la lógica en casos que requieren un tratamiento similar, como la exportación de diferentes tipos de credenciales de acceso a los *routers* o para eliminar diferentes tipos de registros desde un mismo módulo.

3.3.1 Arquitectura del sistema

La arquitectura del sistema adoptada es un enfoque cliente-servidor, donde los clientes pueden ser cualquier navegador web moderno y el servidor contiene el código fuente de la aplicación. Internamente, la aplicación tiene

© RITI. CC BY-NC

una estructura monolítica, en la que tanto el *backend* como el *frontend* y servicios de API se encuentran en un solo proyecto desplegado en el servidor.

Se implementaron módulos dentro de la aplicación dedicados a las diferentes funciones requeridas del sistema, incluyendo el soporte de una API REST que les permite a las máquinas expendedoras comunicarse con el sistema. En la Figura 2 se muestra un diagrama que especifica los principales componentes de la arquitectura de la aplicación y como se relacionan entre ellos.

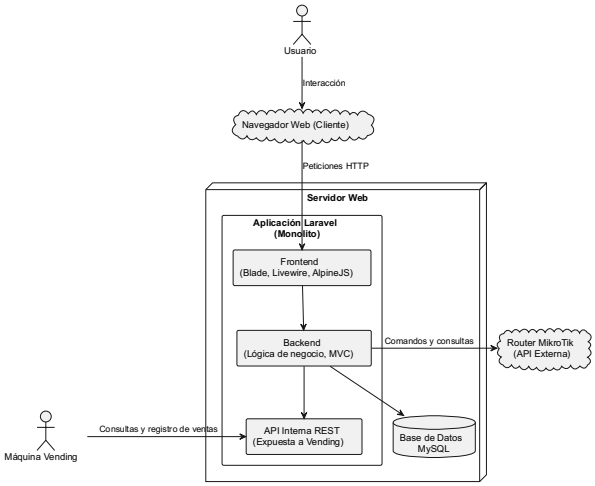


Figura 2. Arquitectura Cliente-Servidor Monolítica con APIs externa e interna.

3.4. Reuniones y revisión del sprint

La metodología requiere de diferentes tipos de reuniones que involucran al *Product Owner/Scrum Master* para planificar y evaluar el avance sobre el proyecto de desarrollo. Se realizaron reuniones conocidas como *Daily Scrum* llevadas a cabo a partir del inicio del proyecto tres veces por semana (lunes, miércoles y viernes) de aproximadamente 15 minutos de duración para revisar los avances, obstáculos y tareas a realizar.

También, fue necesario programar las reuniones conocidas como *Sprint Planning* llevadas a cabo al inicio de cada sprint con la finalidad de seleccionar las historias de usuario y tareas para abordar en dicho periodo con base en el *Product Backlog* planeado y las correcciones de sprint anteriores. Al finalizar cada sprint, se realizó una reunión de revisión en la cual se presentaron los avances alcanzados al *Product Owner/Scrum Master* para evaluar el cumplimiento de las funcionalidades, verificar la calidad de las entregas y proponiendo ajustes.

3.5. Pruebas y Verificación

Para asegurar la calidad del sistema realizaron pruebas funcionales y de aceptación manuales en conjunto con el *Product Owner/Scrum Master* para evaluar que se cumplan las funcionalidades de cada historia de usuario. Además, de garantizar la correcta integración de las máquinas expendedoras y de los *routers* MikroTik con la aplicación desarrollada.

Además de validar el cumplimiento de las funciones, se propuso a validar el rendimiento de la aplicación para corregir y obtener su mayor desempeño de ejecución. Al ser una aplicación realizada con el Framework de Laravel, se seleccionó *Clockwork* como la herramienta ideal para detectar cuellos de botella, uso excesivo de memoria, exceso de consultas a la base de datos, entre otras. La herramienta es fácil de usar, ya que solo requiere una extensión de navegador web y la instalación de su librería PHP en el proyecto para mostrar datos como el tiempo de procesamiento, uso de memoria, número de consultas, uso de caché, errores, excepciones, etc.

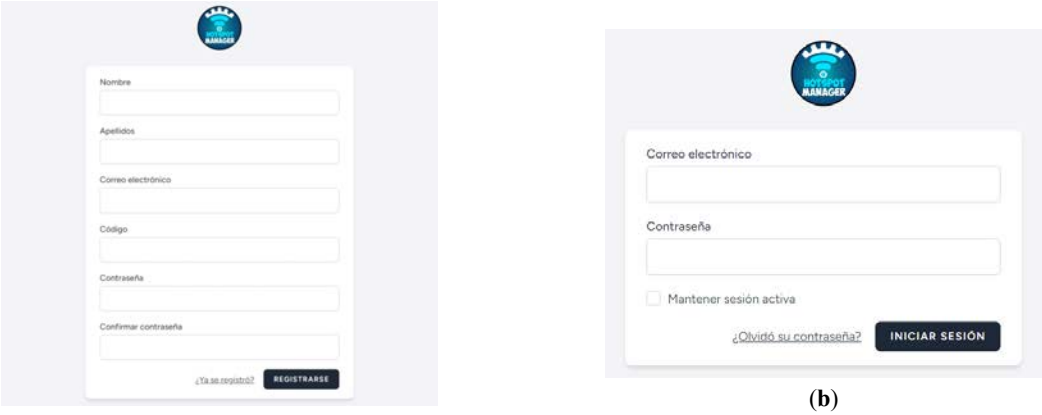
4. Resultados

En esta sección se presentan los resultados obtenidos durante el desarrollo de la aplicación web. Estos se encuentran organizados según los *sprint* planificados utilizando la metodología Scrum.

© RITI. CC BY-NC

4.1. Aplicación web desarrollada

***Sprint* 1: Fundamentos del sistema y gestión de usuarios:** El primer *sprint* estableció las bases del sistema, enfocándose en la gestión de usuarios y la autenticación. A cada usuario se le asigna un código único, utilizado posteriormente para la generación de credenciales *hotspot*. Por su parte, la interfaz de inicio de sesión permite a los usuarios autenticarse mediante el ingreso de su correo electrónico y contraseña. Se incluyó un manejo básico de errores para accesos incorrectos por parte del usuario. Véase la Figura 3.



(a) (b)

Figura 3. Interfaz de registro y autenticación de usuarios: (a) Formulario de registro de nuevo usuario con validaciones; (b) Formulario de inicio de sesión con campos para correo electrónico y contraseña.

También se desarrolló el "Escritorio" o panel principal, que muestra al usuario un resumen visual del sistema mediante gráficos y métricas clave. Esta interfaz proporciona una visión rápida del estado operativo. Véase la Figura 4.

Figura 4. Interfaz "Escritorio" de la aplicación.

***Sprint* 2: Gestión de clientes y administración del sistema:** En este *sprint* se desarrollaron funcionalidades para gestionar clientes y administrar roles y permisos que tiene los usuarios administradores dentro de la aplicación. Los usuarios con los permisos adecuados pueden ver, crear, modificar y eliminar roles dentro de la aplicación. Al crear un nuevo rol el usuario puede seleccionar los permisos que desea. Véase la Figura 5.

© RITI. CC BY-NC

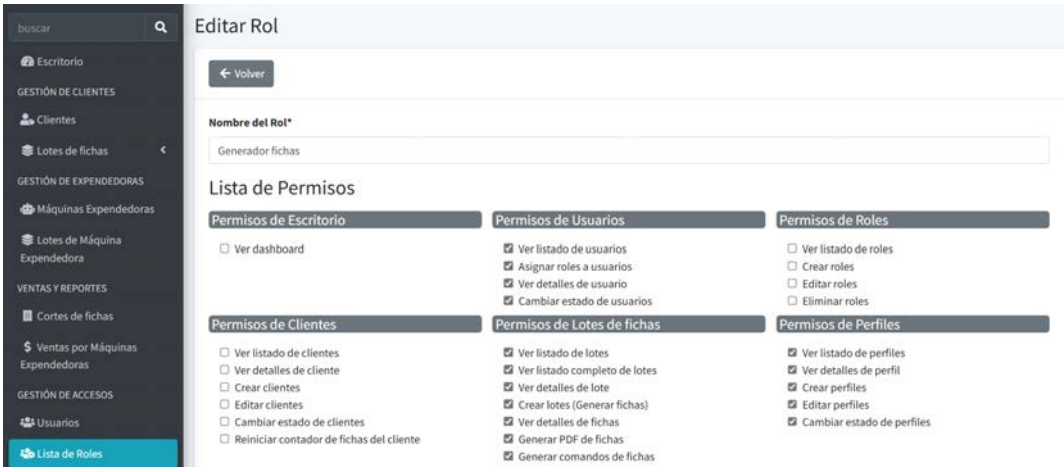


Figura 5. Interfaz de gestión de roles.

Los usuarios con los permisos adecuados pueden registrar y modificar clientes, capturando información clave como nombre, dirección y datos de contacto. El código único en este caso permite crear automáticamente los usuarios para las credenciales *hotspot* generadas por la aplicación. Véase la Figura 6. Los clientes registrados pueden filtrarse y ordenarse a conveniencia del usuario. Las búsquedas permiten usar campos como el nombre del cliente, sus apellidos o su código único de identificación.

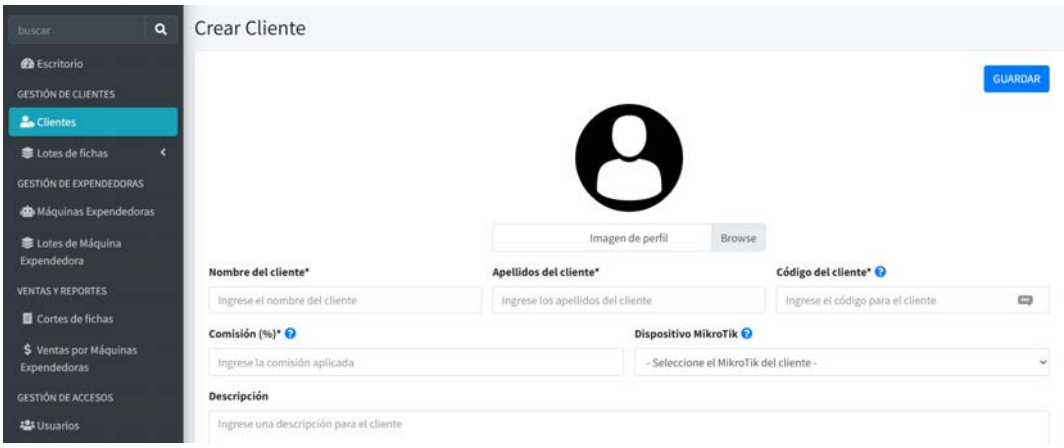


Figura 6. Formulario de creación de cliente.

***Sprint* 3: Gestión de fichas *hotspot* y cortes:** El enfoque de este sprint fue la creación y administración de fichas *hotspot*, además de la generación de cortes. Se desarrolló un sistema que permite generar lotes de fichas, basado en perfiles de fichas registrados previamente. La Figura 7 muestra la interfaz que permite gestionar estos perfiles de fichas.

Figura 7. Interfaz de gestión de perfiles de fichas.

Una vez creados los perfiles de fichas que el usuario necesita, el siguiente paso es seleccionar un cliente y elegir la opción para generarle fichas *hotspot*. En la Figura 8 se muestra la interfaz que permite crear un nuevo lote de fichas ingresando la cantidad de fichas deseada.

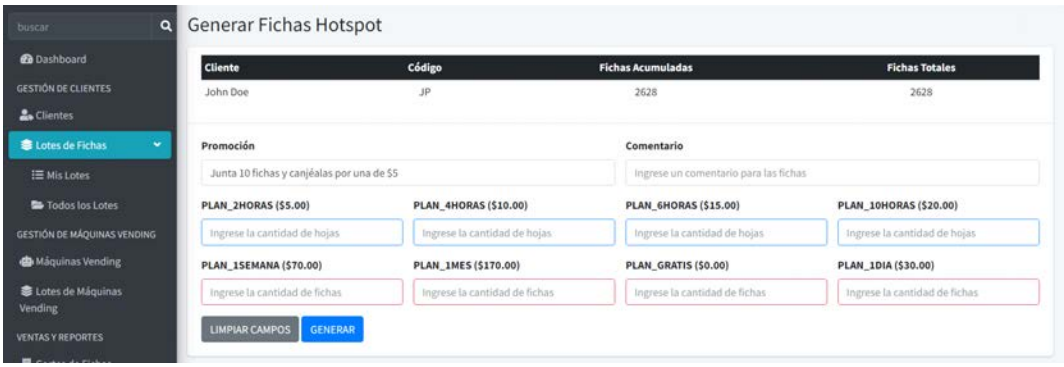


Figura 8. Interfaz de generación lotes de fichas.

Cuando un lote ha sido generado, el usuario puede ver detalles sobre las fichas *hotspot* que se acaban de crear. También puede generar un archivo PDF a partir del lote, generar y exportar los comandos para *routers* MikroTik, o generar un corte de fichas a partir del lote. Véase la Figura 9. Los lotes generados pueden ser filtrados y ordenados para facilitar al usuario su localización. Los usuarios con los permisos necesarios pueden ver no solo los lotes que han generado, sino también los lotes de los demás usuarios de la plataforma.

Figura 9. Interfaz de detalles de lote de fichas.

Los cortes permiten visualizar un resumen de las fichas y realizar cálculos automatizados, mejorando el control administrativo. A partir de los lotes de fichas generados, se puede acceder a la interfaz mostrada en la Figura 10 para crear un nuevo corte de fichas. En esta interfaz el usuario puede ingresar la cantidad de fichas que han sobrado, es decir, que no han sido vendidas, y también la cantidad de fichas que se han registrado con errores o incidencias y que no pueden considerarse para la venta.

***Sprint* 4: Integración con *routers* MikroTik y máquinas *vending*:** Este sprint incluyó la integración técnica con *routers* MikroTik y máquinas vending. Se implementó la funcionalidad para exportar las fichas generadas en cada lote directamente al dispositivo registrado y vinculado a un cliente previamente.

Al editar o crear un cliente, se tiene la opción de vincular un dispositivo para que, al generar los comandos de un lote que pertenece al cliente, en automático sea posible enviar las fichas al dispositivo elegido. Si el cliente no tiene un *router* asociado al momento de intentar exportar los comandos, entonces el usuario puede seleccionar alguno de aquellos registrados previamente en el sistema.

En la Figura 11 se muestra un intento de conexión al dispositivo para exportar las fichas y que sean registradas por el *router*. El registro de cada ficha le permite al cliente acceder a Internet cuando ingrese estas credenciales en el portal de autenticación del *hotspot*.

© RITI. CC BY-NC

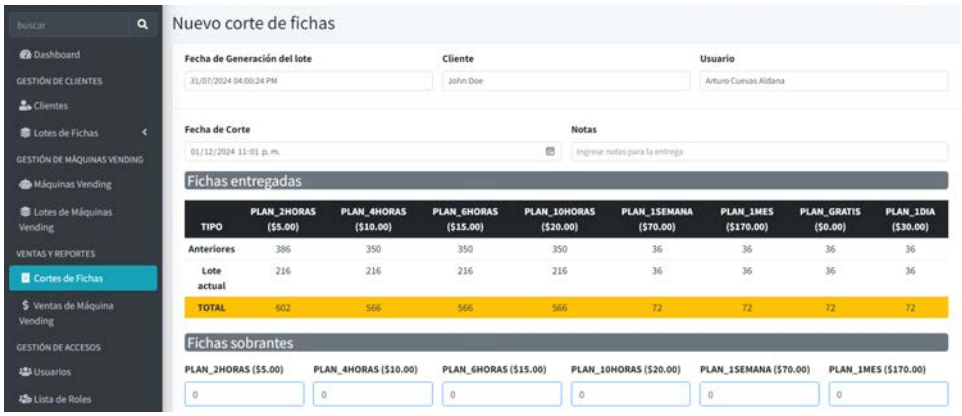


Figura 10. Formulario de creación de nuevo corte de fichas.

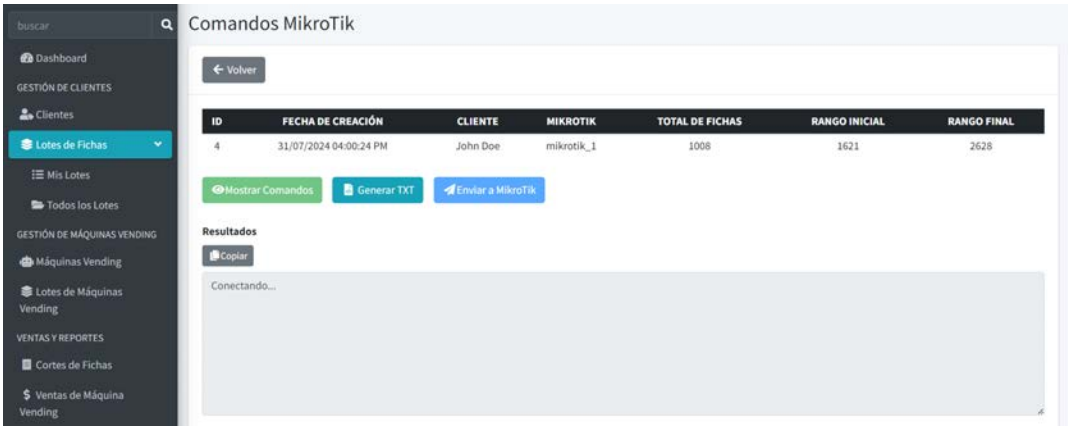


Figura 11. Interfaz de envío de comandos a MikroTik.

La interfaz anterior además permite crear comandos configurables para descargar en formato TXT o copiar en el portapapeles e implementarlos en el dispositivo de forma directa.

La aplicación permite administrar las máquinas vending que tienen acceso a la API publicada por la aplicación web. Al registrar una nueva máquina, se genera una llave con la cual, al agregarla a la configuración interna de la máquina, será capaz de autenticarse en esta aplicación web y obtener los datos que requiera sobre disponibilidad de fichas y realizar las ventas requeridas. Esto se puede ver en la Figura 12.

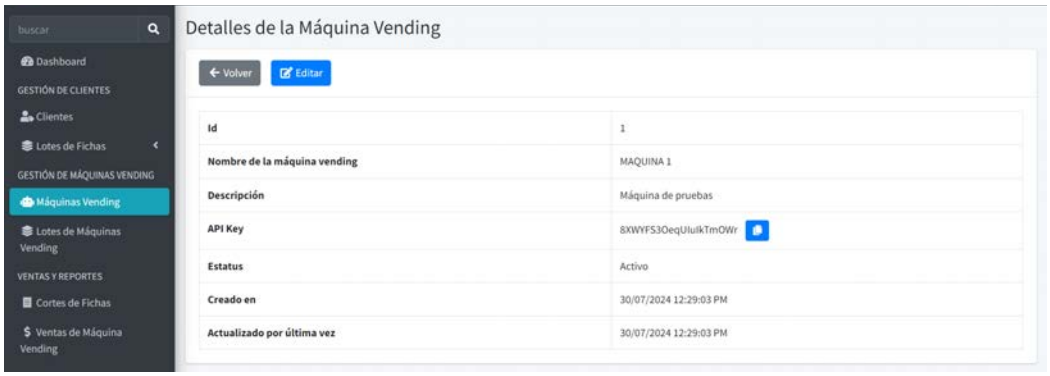


Figura 12. Interfaz de detalles de máquina *vending.*

Se permite generar lotes de fichas *hotspot* específicos para el uso de máquinas vending. Una vez que el lote es creado, cuenta con acciones como la generación y exportación de los comandos al *router* MikroTik y el resumen con los datos principales de creación del lote. Véase la Figura 13.

© RITI. CC BY-NC

Figura 13. Interfaz de detalles de lote de fichas para máquina *vending*

El sistema permite ver un resumen sobre las ventas realizadas por las máquinas vending. La interfaz además permite aplicar filtros para ver detalles específicos dentro de ciertos parámetros de fechas, de perfiles o de la máquina involucrada en cada venta. Véase la Figura 14.

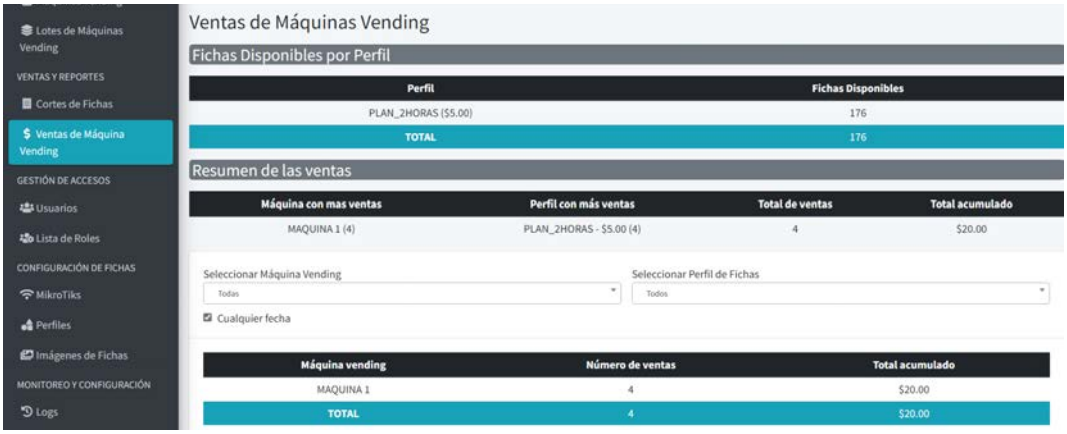


Figura 14. Interfaz de resumen de ventas de máquina *vending.*

***Sprint* 5: Exportación y reportes:** Este sprint se centró en la generación y exportación de datos a través de reportes en formatos PDF y JPG según lo requerido. Además, se implementó un sistema de notificaciones para que el usuario pueda administrar avisos importantes de la aplicación. Se añadieron opciones para exportar lotes de fichas en formatos accesibles para impresión o archivo. En la Figura 15, se muestra un lote de fichas convertido a PDF.

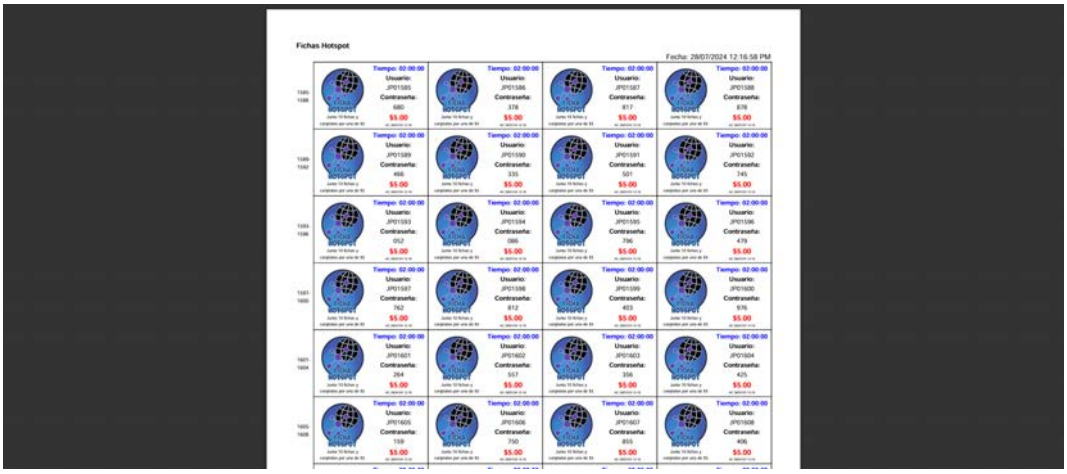


Figura 15. Lote de fichas en formato PDF

En la Figura 16, se muestra un corte de fichas en formato PDF. También es posible realizar la exportación en formato JPG conservando el mismo estilo del PDF generado.

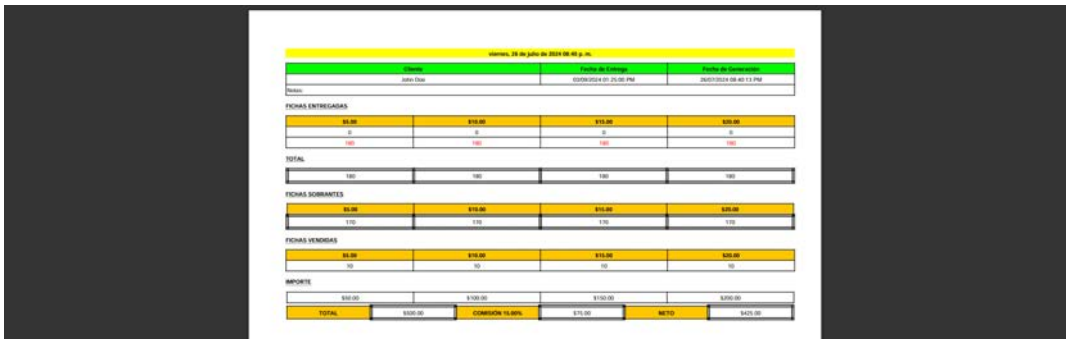


Figura 16. Corte de fichas en formato PDF.

El sistema permite ver las notificaciones que el usuario ha recibido dentro de la aplicación. El usuario puede acceder a ellas a través del desplegable en la barra superior ubicado a la izquierda que se puede ver en la Figura 17.

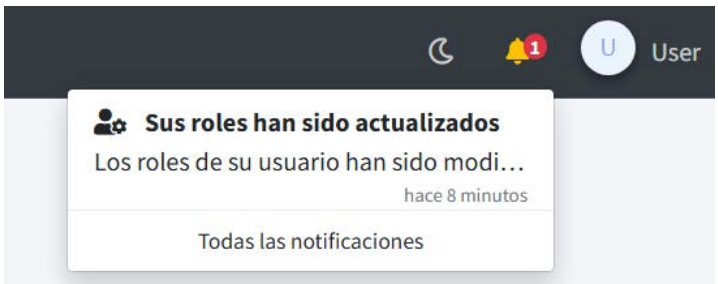


Figura 17. Icono y desplegable de notificaciones dentro de la aplicación.

La interfaz "Notificaciones" muestra todos los avisos recibidos y permite al usuario realizar acciones como marcar las notificaciones como vistas, seguir el enlace de detalles si posee uno y eliminar aquellas que carezcan de importancia. Esta interfaz se puede ver en la Figura 18.

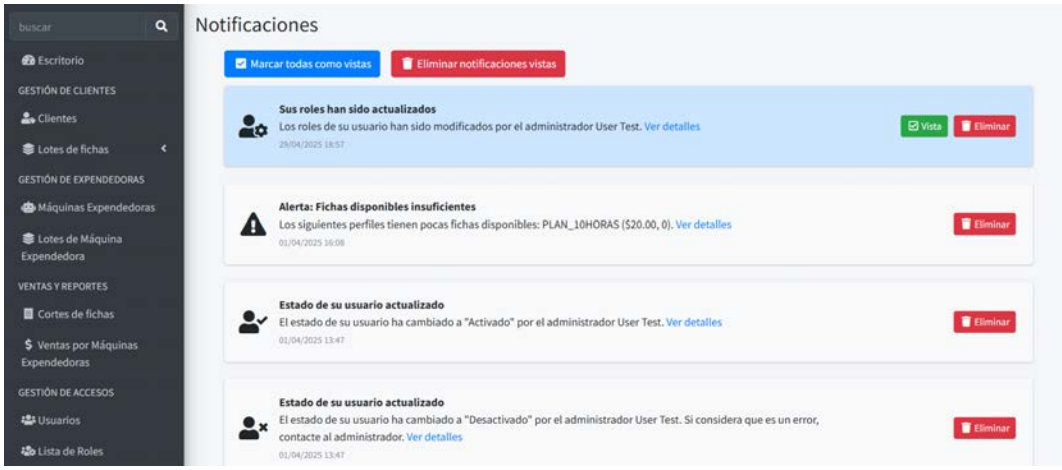


Figura 18. Panel de administración de notificaciones de la aplicación.

***Sprint* 6: Seguridad, auditoría y mantenimiento de datos:** El objetivo de este último sprint fue asegurar un registro sobre las operaciones importantes realizadas dentro de la aplicación, así como mantener la base de datos lo más limpia posible asegurando la eliminación de registros que se consideren desactualizados o sin importancia según el usuario.

El sistema permite visualizar, así como filtrar y ordenar los registros (log) de operaciones importantes realizadas por los usuarios del sistema. Cada registro log permite ver el responsable de la acción, el registro afectado y si hay un cambio entre los valores anteriores y los nuevos valores registrados en el sistema. Véase la Figura 19.

© RITI. CC BY-NC

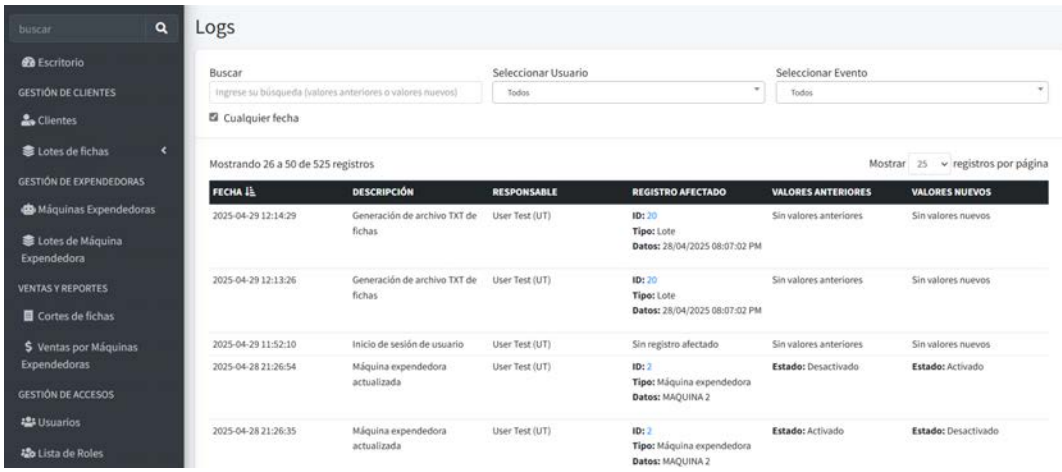


Figura 19. Interfaz de administración de logs del sistema.

Se implementó un módulo con acceso restringido que permite eliminar diferentes tipos de registros dentro de la aplicación. Además, permite al usuario seleccionar filtros específicos según el tipo de registro a eliminar para obtener un mayor control sobre los resultados obtenidos. Véase la Figura 20.

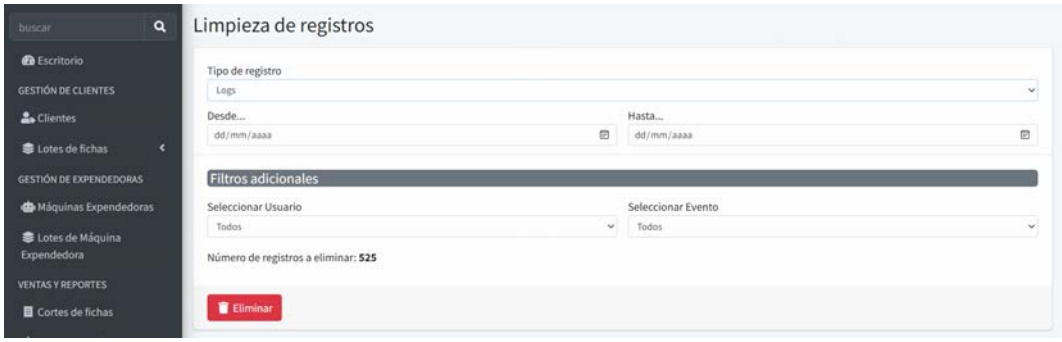


Figura 20. Módulo de eliminación de registros.

4.2. Pruebas funcionales

Como parte del proceso de verificación y validación de la aplicación, se definieron pruebas funcionales basadas en las historias de usuario planteadas durante la planificación del proyecto. A continuación, se presenta una tabla que relaciona cada historia de usuario con sus respectivos casos de prueba, describiendo el comportamiento esperado del sistema ante cada acción evaluada.

Todas las pruebas fueron ejecutadas en un entorno controlado de pruebas, validando que el sistema cumpliera con los requerimientos funcionales establecidos. El resultado global de la fase de pruebas fue satisfactorio, ya que todas las pruebas fueron superadas exitosamente, garantizando la estabilidad y funcionalidad del sistema desarrollado. Pueden observarse las pruebas realizadas en la Tabla 3.

Tabla 3. Pruebas funcionales aplicadas a la aplicación.

ID de Prueba	Historia de Usuario	Caso de Prueba	Resultado Esperado
P-01	HU-01 Registro de Usuario	Registrar un nuevo usuario con datos válidos	El sistema crea un nuevo usuario y muestra un mensaje de éxito.
P-02	HU-02 Iniciar Sesión	Iniciar sesión con credenciales correctas	El usuario accede correctamente al sistema.
P-03	HU-02 Iniciar Sesión	Iniciar sesión con credenciales incorrectas	El sistema muestra un mensaje de error de autenticación.
P-04	HU-03 Modificar Datos Personales	Editar y guardar los datos personales	El sistema actualiza los datos del usuario correctamente.

© RITI. CC BY-NC

P-05	HU-04 Registrar Nuevos Clientes	Registrar un cliente con datos válidos	El cliente se guarda y aparece en la lista de clientes.
P-06	HU-05 Modificar Clientes	Actualizar la información de un cliente existente	Los cambios se reflejan correctamente en el sistema.
P-07	HU-06 Consultar y Buscar Clientes	Buscar clientes por nombre o referencia	El sistema muestra los resultados de la búsqueda de manera correcta.
P-08	HU-07 Gestionar roles y permisos	Asignar roles y permisos a un usuario	El usuario obtiene los permisos y restricciones configurados.
P-09	HU-08 Generar Lotes de Fichas *Hotspot*	Crear un lote de fichas con los parámetros establecidos	Se genera el lote correctamente con la cantidad y configuración indicada.
P-10	HU-09 Administrar Lotes de Fichas *Hotspot*	Ver la lista de lotes de fichas generadas, consultar detalles individuales de cada lote, y visualizar los datos individuales de cada ficha.	Se muestra correctamente la lista de lotes generados, se puede consultar la información de un lote y se puede visualizar cada ficha asociada.
P-11	HU-10 Generar Cortes de Fichas *Hotspot*	Crear un corte de fichas *hotspot*	El sistema genera el corte con los datos y cantidades de fichas correspondientes.
P-12	HU-11 Exportar Fichas *Hotspot* a PDF	Exportar un lote de fichas a PDF	Se genera un archivo PDF descargable con las fichas.
P-13	HU-12 Exportar Cortes de Fichas a PDF o JPG	Exportar un corte de fichas a PDF o imagen	El archivo se genera en el formato seleccionado por el usuario.
P-14	HU-13 Generar Comandos para *Routers* MikroTik	Generar comando para envío de fichas	El sistema genera el comando en el formato adecuado.
P-15	HU-14 Enviar Fichas *Hotspot* a MikroTik	Enviar lotes de fichas al *router* MikroTik	El *router* recibe correctamente las fichas.
P-16	HU-15 Registrar y Gestionar Máquinas Vending	Registrar una nueva máquina y editar datos	Las máquinas vending se almacenan y actualizan correctamente.
P-17	HU-16 Generar Fichas *Hotspot* para Máquinas Vending	Crear fichas específicas para vending	El sistema genera fichas compatibles para vending.
P-18	HU-17 Gestionar Ventas de Máquinas Vending	Consultar y registrar ventas realizadas	Las ventas se registran y consultan de forma correcta.
P-19	HU-18 Alertas y notificaciones	Recibir notificaciones del sistema	Las alertas aparecen en tiempo real o por correo, según configuración.
P-20	HU-19 Gestión de configuraciones del sistema	Modificar parámetros del sistema	Los cambios en la configuración se guardan y aplican exitosamente.
P-21	HU-20 Registro y gestión de logs	Visualizar logs de acciones del sistema	Los registros se muestran con orden y detalle.
P-22	HU-21 Eliminación segura de datos del sistema	Eliminar datos respetando la integridad	El sistema elimina los datos seleccionados de forma controlada.
P-23	HU-22 Gestión de clientes residenciales	Administrar clientes residenciales (altas, bajas, pagos)	El sistema refleja correctamente los movimientos de clientes residenciales.
P-24	HU-23 Importación y exportación de datos del sistema	Importar o exportar información del sistema	La importación/exportación se realiza sin pérdida de datos.

Fuente: Elaboración propia.

Con base en los resultados obtenidos, se concluye que el sistema cumple adecuadamente con los objetivos funcionales propuestos, ofreciendo una operación estable, segura y alineada a las necesidades planteadas en el análisis de requerimientos.

© RITI. CC BY-NC

4.3. Pruebas de rendimiento

Para analizar el desempeño de la aplicación durante su ejecución, se utilizó la herramienta *Clockwork*, la cual permite inspeccionar tiempos de respuesta, número de consultas a la base de datos y uso de memoria en cada solicitud HTTP realizada. El análisis se llevó a cabo en un entorno local utilizando Docker, con los recursos adecuados para simular un uso realista. Se seleccionaron las funcionalidades más representativas de la carga del sistema, registrando los resultados promedio observados en la Tabla 4.

Tabla 4. Métricas obtenidas en pruebas de rendimiento.

Funcionalidad	Tiempo de Respuesta	No. de Consultas SQL	Memoria Usada
Inicio / *Dashboard*	578 ms	17	31 MB
Gestión de Clientes *Hotspot*	1329 ms	8	31 MB
Gestión de Clientes Residenciales	2525 ms	9	32 MB
Generación de Lotes de Fichas	998 ms	22	30 MB
Administración de Lotes	495 ms	20	30 MB
Exportar Fichas a PDF	26 ms por ficha aprox.	21	0.43 MB por ficha aprox.
Exportar Cortes a PDF/JPG	588 ms	15	34 MB
Envío de Fichas a *Routers*	55 ms por ficha aprox.	15	29 MB
Gestión de Máquinas Vending	1137 ms	8	33 MB
Gestión de Logs	588 ms	12	29 MB
Gestión de Configuración	955 ms	10	32 MB

Fuente: Elaboración propia.

Todas las funcionalidades críticas de la aplicación completaron sus procesos con tiempos aceptables y sin errores críticos de rendimiento. El uso de memoria se mantuvo dentro de los márgenes esperados según las pruebas locales. Las consultas a base de datos fueron optimizadas para ejecutarse en tiempos mínimos.

Los resultados indican que los tiempos de respuesta y el uso de recursos fueron aceptables para un entorno de producción de tamaño pequeño a mediano. No se identificaron cuellos de botella críticos en las funcionalidades analizadas.

4.4. Pruebas de eficiencia

Para complementar los resultados obtenidos en las pruebas funcionales y de rendimiento, se realizó una estimación comparativa entre el procedimiento manual tradicional y el sistema desarrollado. Esta comparación tiene como objetivo ilustrar la mejora operativa alcanzada en términos de tiempo, eficiencia y reducción de errores humanos.

La Tabla 5 resume los tiempos aproximados requeridos para realizar distintas tareas administrativas de la red hotspot mediante el enfoque manual previo, frente a los tiempos obtenidos al ejecutar las mismas funciones a través del sistema web desarrollado.

Tabla 5. Comparativa estimada entre procedimiento manual y sistema desarrollado.

Actividad	Procedimiento manual	Sistema desarrollado	Mejora estimada
Registro de 50 clientes	90 minutos	20 minutos	-77%
Generación y exportación de 100 fichas hotspot	60 minutos	2 minutos	-96%
Generación de corte de fichas y cálculo de comisión	30 minutos	3 minutos	-90%
Registro de ventas desde vending	No disponible	Tiempo real	Automatización 100%
Gestión de logs y eliminación segura de datos	Manual y disperso	Centralizado	Mejora cualitativa significativa

Fuente: Elaboración propia.

© RITI. CC BY-NC

5. Conclusiones

El desarrollo de la aplicación web propuesta permitió resolver de forma efectiva los principales desafíos identificados en la administración de redes *hotspot*, especialmente en entornos donde se requiere integrar funcionalidades de gestión de clientes, cortes, fichas de acceso y automatización con *routers* MikroTik y máquinas expendedoras.

La aplicación de la metodología Scrum fue clave para organizar el trabajo en *sprints* enfocados, lo que facilitó la entrega progresiva de funcionalidades y una validación constante del producto. La segmentación del proyecto en historias de usuario priorizadas permitió mantener una visión clara del avance y responder de forma flexible a necesidades emergentes durante el desarrollo.

Las pruebas funcionales demostraron que el sistema cumple con los requisitos definidos, y las pruebas de rendimiento realizadas con *Clockwork* confirmaron tiempos de respuesta aceptables, uso optimizado de recursos y ausencia de cuellos de botella críticos. En particular, funcionalidades complejas como la generación y exportación de fichas *hotspot*, integración con dispositivos MikroTik y automatización de ventas desde máquinas vending se comportaron con eficiencia y estabilidad.

Además, mediante una comparación estimada con el procedimiento manual anterior, se identificó una mejora significativa en los tiempos operativos. Por ejemplo, tareas como registrar clientes, generar fichas o calcular cortes, que antes tomaban entre 30 y 90 minutos, ahora se ejecutan en menos de 5 minutos, reduciendo entre un 70 % y 96 % el tiempo total y eliminando la posibilidad de errores de cálculo humano.

En conjunto, estos resultados confirman que el sistema desarrollado no solo cumple con los objetivos técnicos propuestos, sino que además representa una herramienta viable para la optimización operativa de redes *hotspot*. Como líneas futuras de mejora, se identifican oportunidades de investigación orientadas a la incorporación de métodos de pago electrónicos, una mayor automatización de configuraciones avanzadas en *routers* MikroTik (como reglas de firewall y monitoreo en tiempo real), y el fortalecimiento de las capacidades estadísticas de la plataforma para soportar análisis de patrones de uso y consumo. Estas mejoras abrirían la posibilidad de escalar la solución, incrementar su adopción por parte de proveedores de servicios de Internet y mantener su relevancia a medida que evoluciona la infraestructura tecnológica.

6. Agradecimientos

Los autores manifiestan su profundo reconocimiento al Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango y a la Maestría en Tecnologías de la Información, cuyo valioso apoyo resultó esencial para la ejecución de la presente investigación. Asimismo, expresan su sincero agradecimiento a la Revista de Investigación en Tecnologías de la Información (RITI) por su invaluable contribución al publicar este artículo.

7. Referencias

[1] Fernández, R. (2024). *El uso de Internet a nivel mundial—Datos estadísticos*. Statista. https://es.statista.com/temas/9795/el-uso-de-internet-en-el-mundo/

[2] Statista. (2025). *Países con mayor número de usuarios de Internet en América Latina 2025*. Statista. https://es.statista.com/estadisticas/1073677/usuarios-internet-pais-america-latina/

[3] Luna Santos, R., Santos Quiroz, R. A., Carrera Carrasco, E., González Domínguez, I., Santaella Benavente, M. G. (2021). Creación e implementación de una aplicación web para la administración de citas para un autolavado. *Revista de Investigación en Tecnologías de la Información (RITI)*, *9* (19), 97–108. https://doi.org/10.36825/RITI.09.19.008

[4] Sutherland, J. (2016). *SCRUM - El Arte de hacer el Doble de Trabajo en la Mitad del Tiempo* (1ra. Ed.). Océano de México, S.A. de C.V.

[5] Arroyo Gómez, I. M. (2020). *¿Qué es un hotspot y cómo funciona?* TeamVOX. https://teamvox.com/que-es-un-hotspot-y-como-funciona/

[6] MikroTik. (2019). *MikroTik Routers and Wireless*. https://mikrotik.com/

[7] Castillo Estrada, C. M., Cancino Villatoro, K., Benavides García, V., De La Cruz Vázquez, A. (2022). Diseño de un Sistema web para el control de Curriculum Vitae Electrónico de personal docente basado en una arquitectura orientada a servicios (API REST). *Revista de Investigación en Tecnologías de la Información (RITI)*, *10* (20), 28–42. https://doi.org/10.36825/RITI.10.20.003

© RITI. CC BY-NC

[8] MikroTik. (2020). *Winbox—RouterOS*. https://help.mikrotik.com/docs/display/ROS/Winbox
[9] Purwanto, F. H., Amin, M. (2022). Design and implementation of hotspot network login authentication using QR code based on Mikrotik. *Jurnal Komputer, Informasi Dan Teknologi (JKOMITEK)*, *2* (1), 229–238. https://doi.org/10.53697/jkomitek.v2i1.835
[10] Amarudin, Yuliansyah, A. (2018). Analisis penerapan MikroTik Router sebagai user manager untuk menciptakan internet sehat menggunakan simulasi virtual machine. *Jurnal TAM (Technology Acceptance Model)*, *9* (1), 62–66. https://jurnal.stmikpringsewu.ac.id/index.php/JurnalTam/article/view/598/532
[11] Nainggolan, C., Putra, S. D. (2018). Penggunaan teknologi router MikroTik dalam menunjang jaringan hotspot dan voucher hotspot pada warnet BNET. *Journal of Information System, Informatics and Computing (JISICOM)*, *2* (1), 57–67. https://journal.stmikjayakarta.ac.id/index.php/jisicom/article/view/21
[12] Hendrawan, E., Saputra, A. A. (2021). Desain jaringan RT/RW Net hotspot sistem dengan Mikrotik Routerboard sebagai manajemen billing: Implementasi sistem voucher quota di lingkungan Pringsewu Selatan. *Jurnal Teknol. Komput. Dan Sist. Informasi (JTKSI)*, *4* (2), 69–74. https://doi.org/10.56327/jtksi.v4i2.997
[13] Suryanto, Permadi, F. A. (2019). Optimalisasi internet hotspot menggunakan user manajemen pada Pusat Pengembangan SDM Asuransi Indonesia. *Jurnal Infortech*, *1* (2), 59–67. https://doi.org/10.31294/infortech.v1i2.7083
[14] Aguilar Alvarado, J., Quezada-Sarmiento, R., García-Galarza, K. (2018). Aplicación Java para el control de RB Mikrotik en empresas proveedoras de servicio de Internet. *Revista Ciencia UNEMI*, *11* (26), 161–169. https://doi.org/10.29076/issn.2528-7737vol11iss26.2018pp161-169p
[15] Hamid, H. A., Al Farizi, M. (2022). Peluang jaringan wireless pribadi untuk hotspot umum dengan sistem voucher. *Jurnal Industri Kreatif Dan Informatika Series (JIKIS)*, *2* (1), 1–10.
[16] Rodríguez Ávila, G., Martínez Castillo, C., Toscano Palomar, L., Callejas Melgoza, O. E., & Castro Contreras, R. (2024). Diseño e implementación de un Hotspot con Mikrotik y Access Point Ubiquiti: Gestión eficiente de redes inalámbricas. *REVISTA IPSUMTEC*, *7* (2), 20–27. https://doi.org/10.61117/ipsumtec.v7i2.304
[17] Laravel. (2022). *Laravel Documentation*. Laravel 9.x - The PHP Framework For Web Artisans. https://laravel.com/docs/9.x
[18] Porzio, C. (2023). *Livewire*. Laravel Livewire. https://livewire.laravel.com
[19] Bootstrap Team. (2013). *Bootstrap*. https://getbootstrap.com/
[20] Erickson, J. (2024). *MySQL: qué es y cómo se usa.* https://www.oracle.com/mx/mysql/what-is-mysql/
[21] Amundsen, M. (2023). *RESTful Web API Patterns and Practices Cookbook: Connecting and Orchestrating Microservices and Distributed Data* (1st. Edi.). O'Reilly Media, Incorporated.