



## Aplicación para administrar clientes, cortes y generar fichas hotspot desarrollada con tecnología .NET

### Application to manage clients, outages and generate hotspot cards developed with .NET technology

**Carmen Jeannette Sampayo Rodríguez**

Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango, Puebla, México

carmen.sr@huauchinango.tecnm.mx

ORCID: 0000-0001-8844-6055

**Arturo Cuevas Aldana**

Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango, Puebla, México

d17390015@huauchinango.tecnm.mx

ORCID: 0000-0002-9219-3406

**Jacinto Torres Jiménez**

Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango, Puebla, México

jacinto.tj@huauchinango.tecnm.mx

ORCID: 0000-0002-8006-6397

doi: <https://doi.org/10.36825/RITI.10.20.001>

Recibido: Noviembre 15, 2021

Aceptado: Enero 15, 2022

**Resumen:** En este artículo se muestra el desarrollo de una aplicación de escritorio hecha a la medida de tipo *Windows Forms* que permite realizar la gestión de usuarios con sus permisos de acceso, clientes, generación de fichas de puntos de acceso (*hotspot*) y la administración de cortes de los clientes para un proveedor de servicio de internet inalámbrico (WISP). Se siguió el marco de trabajo de SCRUM, con un ciclo de vida en cascada, para obtener los requerimientos se realizaron entrevistas, en el diseño se utilizó el Lenguaje de modelado unificado (UML), se empleó una arquitectura monolítica con un diseño en capas, para el desarrollo de la aplicación se utilizó la tecnología .NET con el lenguaje de programación C# y con la ayuda del entorno de desarrollo integrado *Visual Studio* en su versión *Community 2019* y el Sistema Gestor de Base de Datos *SQL Server 2019* en versión *Express* y se utilizó la librería de software *iText 7 Community* para la integración de funcionalidad PDF. Con el sistema desarrollado se automatizó el proceso de generación de fichas, se disminuyó en un 94.96% el tiempo que tarda el proveedor de servicio en generar la fichas y el archivo de texto que contienen usuario y contraseña, se garantizó que las claves para acceder al servicio de internet sean únicas, permitió administrar los usuarios y realizar los cortes de los clientes en los diferentes puntos de acceso, siendo el sistema rápido y eficiente.

**Palabras clave:** *Aplicación .NET, Hotspot, WISP.*

**Abstract:** This article shows the development of a custom-made Windows Forms desktop application that allows

the management of users with their access permissions, clients, hotspot generation and client outage management for a wireless internet service provider (WISP). The SCRUM framework was followed, with a waterfall life cycle, interviews were conducted to obtain the requirements, the Unified Modeling Language (UML) was used in the design, a monolithic architecture with a layered design was used, for the development of the application the .NET technology was used with the C# programming language and with the help of the integrated development environment Visual Studio in its Community 2019 version and the SQL Server 2019 Database Management System in Express version and the iText 7 Community software library was used for the integration of PDF functionality. With the system developed, the token generation process was automated, the time taken by the service provider to generate the tokens and the text file containing the user name and password was reduced by 94.96%, it was guaranteed that the passwords to access the Internet service are unique, it allowed the administration of users and the cutting of clients at the different access points, and the system was fast and efficient.

**Keywords:** *Application .NET, Hotspot, WISP.*

## 1. Introducción

Las Tecnologías de la Información y Comunicación (TIC) actualmente cuentan con una integración cada vez mayor en diversos ámbitos de la sociedad humana, principalmente en la solución de problemas que beneficien y den solución a los contextos rurales.

Proveer de servicio de internet, ya sea en un negocio, escuela o en alguna comunidad en donde el acceso a internet no es por el servicio alámbrico sino por servicio inalámbrico, puede representar un problema mayor cuando la administración de los usuarios que pueden conectarse a esta red es muy grande y se vuelve difícil bloquear ciertos usuarios o permitir su conexión solo por unas horas. Colocar una contraseña a una red y darla a conocer a los usuarios que se busca dar acceso a la red, puede ocasionar ciertos problemas como la reducción en el ancho de banda y un bajo rendimiento de la red en general. Estos problemas pueden percibirse como un mal servicio y como una mala experiencia de usuario. Para evitar estos problemas es que son utilizadas las fichas *hotspot*. Una ficha *hotspot* es el medio impreso o digital que indica principalmente el nombre y la contraseña para permitir la conexión a internet de un cliente a algún punto de acceso de algún proveedor de servicios que se encuentre en el área. Se incluyen además datos como la duración, el precio y la fecha de generación. Las fichas *hotspot* pueden crearse de forma manual, utilizando software de aplicación general, pero la administración de las fichas generadas, así como de los clientes a los que pueden ser generadas, puede ser complicado y llevar mucho tiempo.

A pesar de que en internet existe software comercial que puede realizar estas acciones, resultan muy costosos e incluyen funcionalidades que pueden ser no requeridas o necesarias y que pueden ralentizar el sistema. En este proyecto se propuso la realización de un sistema de generación de fichas *hotspot* de forma automática y en un menor tiempo, con una administración eficiente, creando los usuarios y las contraseñas ideales para permitir a los usuarios conectarse a la red sin tantos problemas. Permitiendo la administración y control de fichas de puntos de acceso a internet (*hotspot*), mediante el desarrollo de una aplicación generadora de fichas para la solución de problemas que presentan los proveedores servicios de internet inalámbrico a la hora de generar y proveer acceso al servicio de internet beneficiando de alguna manera a las comunidades rurales en las que se emplea la distribución de fichas para acceder a internet desde los diferentes puntos de acceso.

El proveedor de servicios de internet inalámbrico, caso de estudio, realiza la generación de fichas para acceder a sus puntos de acceso usando una herramienta de cálculo y le lleva aproximadamente una hora generar las fichas para un cliente que realiza la venta de fichas individuales, esto debido a que debe tener cuidado de que las fichas que genera no se repitan las cuentas de usuarios y contraseñas. Además de presentar problemas de seguridad o pérdida de información, la carga de datos puede ser demasiada para manejarse en formato de archivos, pudiendo crear inconsistencias en los datos o incluso, ocasionar un error que produzca que el fichero quede inutilizable. El proceso de la realización de fichas se realiza de forma casi manual, con un aprovechamiento de tiempo no óptimo y con la posibilidad de incurrir en errores de secuencia y, por ende, la pérdida de tiempo en correcciones.

Por lo que se requiere un sistema para la generación automática de fichas *hotspot* en formato PDF, el cual servirá para:

- Mantener un registro y control de administradores, debido a lo difícil de llevar el control de permisos de

los usuarios y monitorear las acciones que estos realizan hacia los clientes.

- Mantener un registro y control de clientes, es necesario saber los datos del cliente al cual se le están realizando las fichas *hotspot*.
- Mantener un registro de las fichas que se han generado, se requiere de un registro de fichas en el cual mantenga los datos de quien fue el responsable de la generación, el cliente al que pertenece y si han sido entregadas al cliente o no.
- Generar en un documento PDF de formato estándar las fichas *hotspot* con los datos solicitados y generar a la vez un archivo de texto con los datos de conexión necesarios para posteriormente dar acceso al cliente a la red introduciéndolos al equipo de Mikrotik.

En este artículo se describe el marco de trabajo empleado como metodología y arquitectura del sistema y las actividades realizadas en cada etapa de desarrollo: en la primera etapa se muestra la lista de requerimientos funcionales y no funcionales y diagrama de casos de uso, el diagrama de despliegue y la arquitectura monolítica del sistema y diseño en capas, el diagrama lógico del sistema y el diagrama relacional, en la segunda etapa se mencionan las herramientas utilizadas para el desarrollo del sistema, en la tercera etapa el caso de prueba realizado, en la última etapa se muestran los resultados obtenidos y las interfaces del desarrollo de la aplicación “Sistema Generador de Fichas *Hotspot*”; finalmente, se presentan las conclusiones, agradecimiento y referencias utilizadas para la elaboración de este artículo.

## 2. Estado del arte

### 2.1. Scrum

Un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Es liviano, fácil de entender, difícil de llegar a dominar. Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo de modo que podamos mejorar. El marco de trabajo Scrum consiste en los Equipos Scrum y sus roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos [1].

### 2.2. Lenguaje de Programación C#

Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO [2].

### 2.3. Librería iText7

Es una biblioteca *Open Source* para crear y manipular archivos PDF, RTF, y HTML en Java. Fue escrita por Bruno Lowagie, Paulo Soares, y otros; está distribuida bajo la *Affero General Public License*. El mismo documento puede ser exportado en múltiples formatos, o múltiples instancias del mismo formato. Los datos pueden ser escritos a un fichero o, por ejemplo, desde un *servlet* a un navegador web. Más recientemente, ha sido extendida a una biblioteca PDF de propósito general, capaz de rellenar formularios, mover páginas de un PDF [3].

### 2.4. Servidor de Base de Datos SQL Server 2019 Express

Es un sistema de administración de datos gratuito, eficaz y confiable que ofrece un almacén de datos completo y confiable para sitios web ligeros y aplicaciones de escritorio [4].

### 2.5. Microsoft SQL Server

Es un sistema de gestión de base de datos relacional (RDBMS) producido por *Microsoft*. Su principal lenguaje de consulta es *Transact-SQL*, una aplicación de las normas ANSI / ISO estándar *Structured Query Language* (SQL) utilizado por ambas *Microsoft* y *Sybase*. Entre sus características están el soporte de transacciones, escalabilidad,

estabilidad y seguridad, soporta procedimientos almacenados. Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente, permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red solo acceden a la información. Además, permite administrar información de otros servidores de datos. Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos, pero orientado a proyectos más pequeños, que en su versión 2005 pasa a ser el *SQL Express Edition*, que se distribuye en forma gratuita. *Microsoft SQL Server* constituye la alternativa de *Microsoft* a otros potentes sistemas gestores de bases de datos como son *Oracle*, *Sybase ASE*, *PostgreSQL* o *MySQL* [5].

## 2.6. Visual Studio Community 2019

Un entorno de desarrollo integrado (IDE) es un programa con numerosas características que respalda muchos aspectos del desarrollo de software. El IDE de *Visual Studio* es un panel de inicio creativo que se puede usar para editar, depurar y compilar código y, después, publicar una aplicación. Aparte del editor y el depurador estándar que proporcionan la mayoría de IDE, *Visual Studio* incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas más características para facilitar el proceso de desarrollo de software [6].

Existe implementación de software como el que se muestra en la tesis [7] que permite administrar y automatizar un proceso de negocios y también optimizar la red del proveedor de servicios, para mejorar e incrementar los actuales servicios, así como de incrementar su capacidad actual para permitir un mayor acceso a Internet y planes de mayor capacidad, automatizar procesos del negocio por medio de la integración de un sistema externo comunicado con los dispositivos de Red.

Desarrollo de una aplicación java para el control de RB Mikrotik en empresas proveedoras de servicio de internet // java, que contiene ubicación GPS, corte y reconexión, compilado en las plataformas Windows y Linux, con una base de datos *Postgres*. Por consiguiente, se obtiene una mejora considerable en la administración de sus clientes y optimización de procesos empresariales vinculados con los servicios de internet [8].

Algunos de los problemas que presenta el promovedor de servicios de internet inalámbrico (WISP), que proporcionó los requerimientos del sistema, es el tiempo que tarda en realizar la gestión de fichas que generan para el sistema de prepago, debido a que tienen que generarlas previamente, por punto de acceso (*hotspot*), por tiempo (hora, día, semana, etcétera) y denominaciones distintas (5, 10, 15, etcétera), y generalmente utilizan herramientas de cálculo que complica la aleatoriedad de nombre de usuario y contraseña. La generación de fichas lleva a cabo para cada uno de sus clientes o socios, quienes son los encargados de realizar la venta individual de las fichas a los usuarios finales y tiene que llevar a cabo la administración de fichas generadas para cada estación de cada cliente y realizar los cortes de la venta de las fichas generadas.

A pesar de que existen en línea aplicaciones y software comercial para realizar la tarea de generar fichas *hotspot* resulta muy costoso y complejos debido a que incluyen funcionalidades que no requiere el usuario o no usa en su totalidad, a continuación, se muestran la Tabla 1 con una comparativa con características, ventajas y desventajas de software similar al desarrollado.

**Tabla 1.** Comparativa de software similar.

	Características	Ventajas	Desventajas
<b>Winbox [9]</b>	Software Oficial de <i>MikroTik</i> para acceder por Interfaz gráfica del <i>Router Board</i> .	Acceso directo al ingreso de fichas en el sistema de la red.	Difícil de usar. No puede generar las fichas por si solo. No lleva un control de clientes.
<b>EasyTickets Free [10]</b>	Aplicación Creada por "Internet por fichas.com", compañía dedicada a distribuir dispositivos de venta de fichas <i>hotspot</i> .	Creación de perfiles y fichas <i>hotspot</i> . Diseño de fichas agradable.	No lleva un control de los clientes a los que se generan fichas.
<b>VoucherHotspot [11]</b>	Es Un Sistema Generador de Fichas Para <i>Hotspot</i> de <i>MikroTik</i> .	Fácil de usar. Lleva un registro de las fichas generadas, de los perfiles y de usuarios. Permite diseñar el reporte de	Tiene un costo de licencia de \$800.00 MXN.

		impresión con la cantidad de fichas deseada por hoja, crear las fichas personalizadas, y elegir los campos a imprimir.	
<b>Wisphub [12]</b>	Sistema de Administración <i>Wisp</i> en la nube.	Cero instalación. Cortes automáticos. Aviso de pago en pantalla. Historial de tráfico. Facturación automática. Facturación electrónica. Mensajería-SMS/ <i>Whatsapp</i> . Control del <i>Router</i> desde el software.	Licencia gratuita al contar con menos de 30 clientes. Al superar la tarifa gratuita, se tiene que comprar una licencia de 15 a 70 dólares mensuales dependiendo el plan contratado.
<b>CYBER ALONSO [13]</b>	Blog dedicado a ofrecer información sobre diversas tecnologías de red y computación.	Explicación de como generar fichas <i>hotspot</i> mediante Excel. Proporcionamiento de las plantillas prefabricadas para realizarlo con mayor facilidad	El software Excel de Microsoft requiere de una licencia. Es un software no dedicado y a la medida que contiene herramientas no utilizadas.

Fuente: Elaboración propia.

Para dar solución a la problemática planteada se tomaron en cuenta algunas Razones para elegir el enfoque de las aplicaciones de escritorio en el desarrollo: Las aplicaciones de escritorio interactúan mejor con el equipo del usuario, el rendimiento para cálculos complejos es mucho mayor que el de las aplicaciones web, la ejecución de lógica personalizada en el lado cliente es posible, pero mucho más difícil, con una aplicación web, el uso de *multithreading* es más fácil y eficaz, a curva de aprendizaje para diseñar interfaces de usuario (UI) no es pronunciada. Además, en el caso de *WinForms*, el Diseñador de *Windows Forms* ofrece una experiencia de arrastrar y colocar muy intuitiva, empezar a programar y probar los algoritmos es fácil, ya que no es necesario configurar una infraestructura de servidor ni preocuparse por los problemas de conectividad, los *firewalls* ni la compatibilidad del explorador, la depuración es eficaz en comparación con la depuración web. Dado que la tecnología ya lleva tiempo entre nosotros, hay muchos expertos y una base de conocimiento disponible para desarrollar aplicaciones de escritorio [14].

Es por eso que se pretende resolver la problemática con el desarrollo de una aplicación de escritorio hecha a la medida que permita automatizar el proceso de negocio para realizar la generación de fichas de cada *hotspot*, administrar a los clientes y realizar los cortes de manera rápida y eficiente.

### 3. Materiales y métodos

Se aplicó el marco de trabajo ágil de SCRUM. En un primer momento, se define el *Product Backlog*. Este, podría definirse como una lista de las funcionalidades que el cliente quiere que tenga el producto, ordenadas de mayor a menor importancia. Una vez definido el *Product Backlog*, se comenzó a desarrollar la lista de tareas de la iteración o *Sprint Backlog*, que consistió en un índice con ciertos puntos del *Product Backlog* que se contemplaron en la primera parte del desarrollo. El equipo de trabajo definió su duración.

A lo largo del *Sprint Backlog* también se realizaron reuniones con el cliente y el equipo de trabajo. Estas reuniones son conocidas como *Sprint Planning Meeting* y sirvieron para determinar cómo enfocar el proyecto, las etapas y los plazos. Durante todo el periodo del *Sprint* se realizaron reuniones diarias conocidas como *Daily Scrum* o *Stand-up Meeting*. Este último concepto se traduce como «reuniones de pie», pues se recomienda que se realicen de pie y que no duren más de cinco o diez minutos. En estas reuniones el equipo de trabajo planteo y respondió tres preguntas: ¿Qué hice ayer?, ¿qué voy a hacer hoy?, ¿qué ayuda necesito?

Con el *Sprint* finalizado, se dio comienzo a la fase conocida como *Sprint Review*, que consistió en una revisión de todos los avances del proyecto hasta la fecha. En este punto el equipo de trabajo tuvo un producto tangible que puedo ser presentado al cliente. Por último, se revisaron los objetivos cumplidos y se recapituló qué se ha hecho bien y qué no y se apuntó para evitar cometer los mismos errores. A este último punto se le conoce como *Sprint Retrospective*. Aplicando las fases del desarrollo de los productos entregables: Análisis y diseño,

implementación y pruebas en cada *sprint*.

- **Análisis:** Se trabajó con el levantamiento de requerimientos del sistema a desarrollar a través de entrevistas realizadas al cliente, se generó para la obtención de requisitos funcionales del sistema y diseño el diagrama de casos de uso que se muestra en la Figura 1, junto con su especificación de casos de uso. Una vez realizado el análisis de la información proporcionada se enlistó en la Tabla 2 las funciones básicas del sistema. En la Tabla 3 se muestran los requerimientos no funcionales del sistema de administración y generador de fichas *hotspot*.
- **Diseño:** Existen muchas definiciones de la arquitectura de software. En este trabajo se asume la siguiente definición: “la arquitectura de software es una descripción de los subsistemas y componentes de un sistema de software y las relaciones entre ellos” [15] y [16].

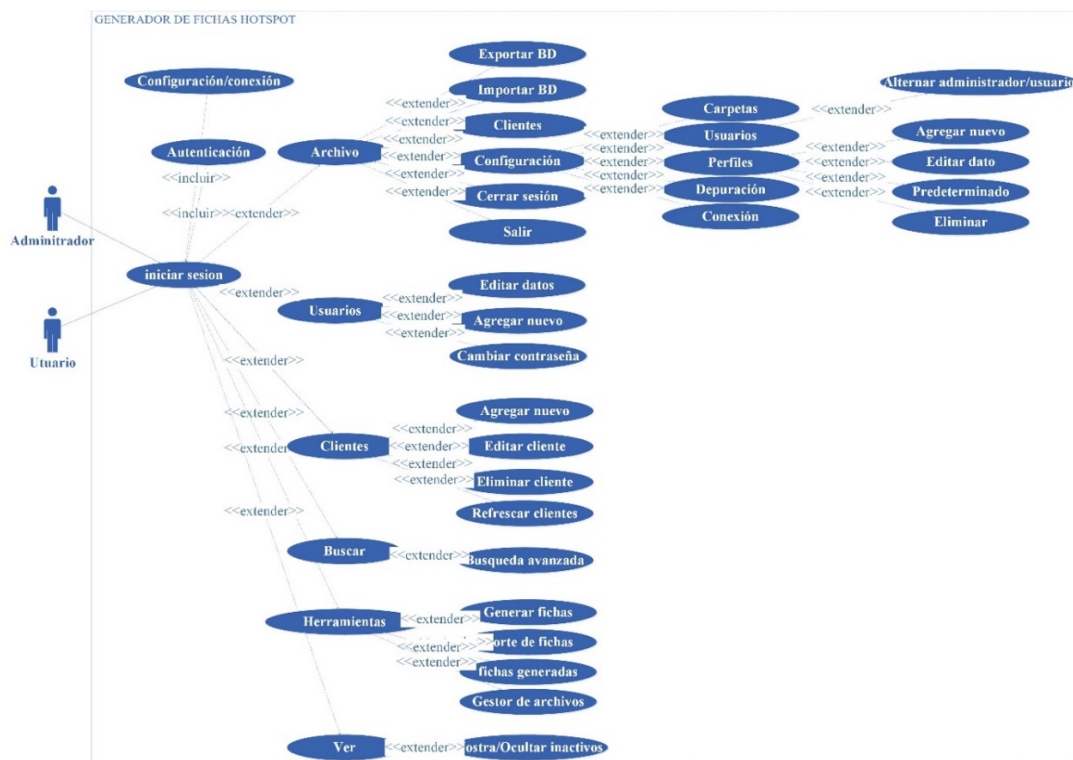


Figura 1. Diagrama de casos de uso del Sistema Generador de Fichas *Hotspot*.

Tabla 2. Las tablas de requerimientos funcionales del sistema.

Código	Nombre	Descripción
RF-001	Insertar, modificar y eliminar Clientes en la base de datos	Los administradores deben ser capaces de insertar, eliminar y modificar Clientes.
RF-002	Insertar, modificar y eliminar administradores del sistema en la base de datos	Los administradores deben ser capaces de insertar, eliminar y modificar Administradores del sistema.
RF-003	Insertar, modificar y eliminar perfiles para fichas en la base de datos	Los administradores deben ser capaces de insertar, eliminar y modificar perfiles de fichas del sistema.
RF-004	Insertar y eliminar fichas generadas en la base de datos	Los administradores deben ser capaces de insertar y eliminar fichas generadas del sistema.
RF-005	Insertar y eliminar archivos en la base de datos	Los administradores deben ser capaces de insertar y eliminar archivos generados del sistema.

RF-006	Alternar estado de actividad en clientes y administradores	Los administradores deben ser capaces de alternar entre “activo” e “inactivo” el estado de los clientes y administradores en la base de datos.
RF-007	Alternar niveles de permiso para administradores entre “Administrador” y “Usuario”	Los administradores deben ser capaces de alternar entre “Administrador” y “Usuario” el nivel de acceso de los administradores en la base de datos.
RF-008	Consultar los administradores registrados	Los administradores deben ser capaces de consultar los administradores desde la base de datos.
RF-009	Consultar los Clientes registrados	Los administradores deben ser capaces de consultar los clientes desde la base de datos.
RF-010	Consultar los perfiles de ficha registrados	Los administradores deben ser capaces de consultar los perfiles de ficha desde la base de datos.
RF-011	Consultar las fichas registradas	Los administradores deben ser capaces de consultar las fichas generadas desde la base de datos.
RF-012	Consultar los archivos registrados	Los administradores deben ser capaces de consultar los archivos desde la base de datos.
RF-013	Generar fichas con el usuario y contraseña creado para un cliente elegido por el administrador	Los administradores deben ser capaces de generar automáticamente fichas con un perfil de ficha y para un cliente seleccionados previamente.
RF-014	Crear un archivo PDF y TXT con las fichas generadas	Se debe generar automáticamente al generar fichas, un archivo PDF de forma gráfica y atractiva, y un archivo de texto con comandos para introducir al router las fichas generadas.
RF-015	Permitir el acceso ( <i>login</i> ) de los administradores al sistema	Se debe permitir a los administradores ingresar con un usuario y contraseña al sistema para poder utilizarlo.
RF-016	Permitir la Exportación de la base de datos	Se debe permitir exportar la base de datos completa, y también mediante archivos CSV con los datos de clientes y administradores para utilizarla según convenga al administrador
RF-017	Permitir la Importación de la base de datos	Se debe permitir importar la base de datos completa y también con un archivo CSV para la inserción de clientes y administradores por lotes.
RF-018	Permitir el filtrado de datos mostrados	Se debe permitir que los administradores filtren los datos mostrados mediante su estado de actividad registrado.

Fuente: Elaboración propia.

**Tabla 3.** Las tablas de requerimientos no funcionales del sistema.

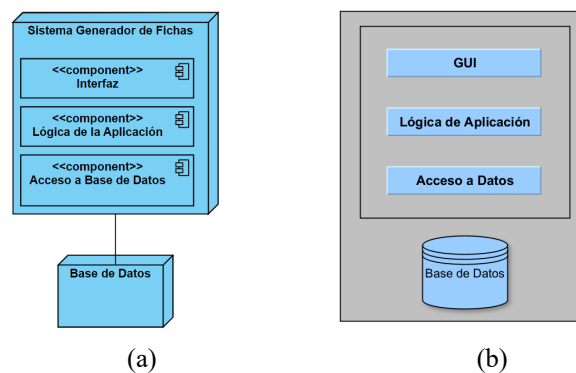
<b>Código</b>	<b>Nombre</b>	<b>Descripción</b>
RNF-001	Crear una interfaz visualmente atractiva	Diseñar una interfaz fácil de usar y que sea atractiva para los usuarios.
RNF-002	Crear un logotipo para el software	Diseñar un icono utilizado para el software y la documentación.
RNF-003	Crear una imagen alterna para las fichas <i>hotspot</i>	Diseñar una imagen para añadir a las fichas generadas en caso de no haber imágenes para cargar.
RNF-004	Diseñar los formatos de las plantillas para los archivos PDF	Diseñar las plantillas para mostrar las fichas generadas y los formatos de corte.
RNF-005	Permitir la modificación de valores de carpetas	Permitir la modificación de las carpetas de salida y la carpeta de imágenes para las fichas.
RNF-006	Permitir la modificación de los valores de conexión	Permitir la modificación del servidor de base de datos, y del usuario y contraseña de acceso

RNF-007	Permitir el filtrado de los datos mostrados en tablas	Permitir el filtro de registro por los atributos de los datos mostrados en pantalla
---------	---	---

Fuente: Elaboración propia.

El estilo arquitectónico empleado fue monolítico, ya que el sistema monolítico implementa absolutamente toda la funcionalidad necesaria para funcionar, y es autosuficiente para los requerimientos establecidos por el cliente y ya que se desarrolló para una pequeña empresa esta aplicación resulta apropiada. Una aplicación monolítica típica usa un diseño de capas independiente para la interfaz de usuario, la lógica de la aplicación y el acceso a los datos [17][18].

En la Figura 2 se presenta la arquitectura monolítica representada en un diagrama de despliegue, con un diseño de capas empleada para la implementación de la aplicación de escritorio de tipo *Windows Forms* del sistema de administración y generador de fichas *hotspot*.



**Figura 2.** (a) Diagrama de Despliegue del Sistema, (b) diseño de capas del Administrador y Generador de Fichas *Hotspot*.

Se realizó el diseño del software utilizando herramientas *CASE UML*. Una vez realizada la fase de análisis, se procedió al diseño del sistema, entre lo que se encuentra, el diseño de la lógica, el diseño de la base de datos que se muestran en la Figura 3 y la Figura 4 respectivamente.

- **Desarrollo:** Para el desarrollo de la una aplicación hecha a la medida de tipo *Windows Forms* se utilizó *.Net framework* con *C#* como base porque además de ser el lenguaje de programación del que se tiene mayor dominio se considera que: se tiene herramientas que hacen más fácil el desarrollo, se llega a un resultado con menos líneas de código que con otros lenguajes de programación, la interfaz que permite genera esta más ligada a *Windows*, que es el Sistema Operativo en el que va a ejecutar y fue uno de los requerimientos, las interfaces que se generan son estéticas y agradables. Se utilizó el entorno de desarrollo integrado *Visual Studio* en su versión *Community 2019*. El Sistema Gestor de Base de Datos elegido para la administración y manejo de los datos fue *SQL Server 2019* en versión *Express*. También se consideró la librería de software *iText 7 Community*, permitiendo la integración de funcionalidad PDF en el sistema de software.
- **Pruebas:** Se realizaron las pruebas necesarias de funcionalidad del sistema, se comenzó con una fase de pruebas, verificando el correcto funcionamiento de cada opción y herramienta ofrecida por el sistema, comprobando si el sistema cumple con las funciones básicas que fueron solicitadas y necesarias para resolver la problemática. Todas las observaciones y posibles errores en la lógica se corrigieron mejorando el rendimiento del software.



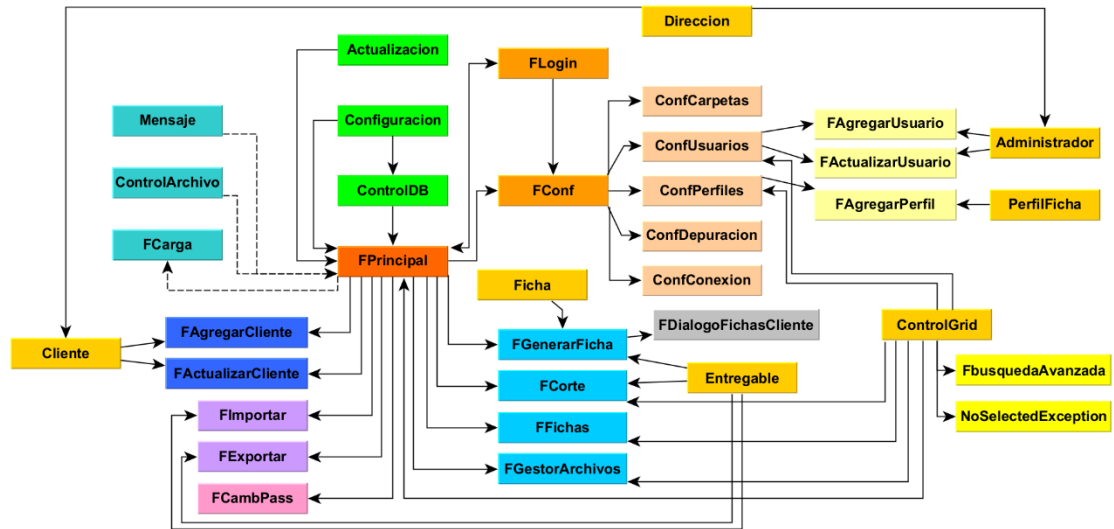


Figura 3. Diseño de la lógica del sistema.

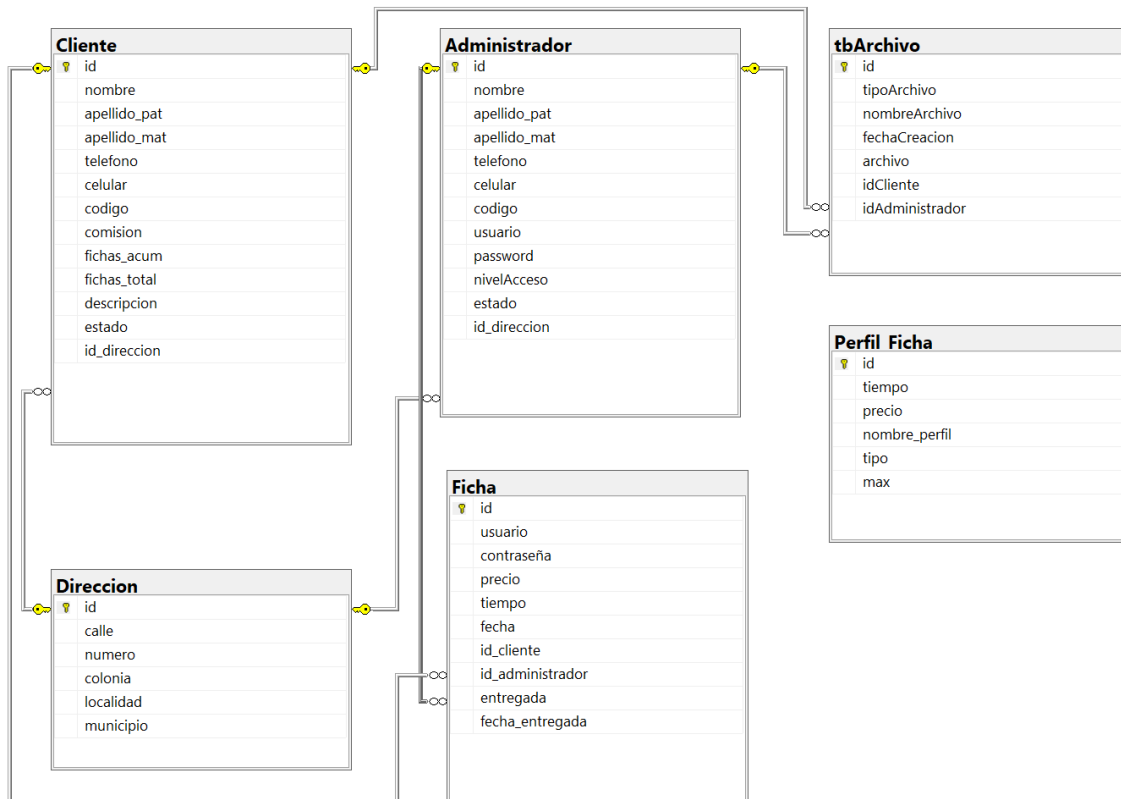


Figura 4. Diagrama relacional.

En la Tabla 4 se muestra una plantilla con los casos de prueba realizados. Donde: **Código:** Indica en número de identificación del caso de prueba, **Módulo a probar:** Indica al módulo que se le ejecuta el caso de prueba. **Datos de entrada:** Datos ingresados por el usuario del sistema. **Prerrequisito:** Lo que se debe cumplir para poder llevar a cabo el caso de prueba. **Resultado esperado:** Es lo que se espera que ocurra al finalizar el caso de prueba (correcto o incorrecto). **Resultado obtenido:** Es lo que se obtiene al finalizar el caso de prueba (Ok: Resultado Correcto, Error: Resultado Incorrecto).

Tabla 4. Casos de prueba del sistema.

<b>Código</b>	<b>Módulo a probar</b>	<b>Datos de entrada</b>	<b>Prerrequisito</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
CP-001	<i>Login</i>	Usuario: abc Contraseña: abc		Usuario o contraseña no válidos	Ok
CP-002	<i>Login</i>	Usuario: admin Contraseña: admin	Usuario ingresado en el sistema	Ingreso al sistema como “Administrador (ROOT)”	Ok
CP-003	<i>Login</i>	Usuario: user Contraseña: test	Usuario ingresado en el sistema	Ingreso al sistema como “Usuario”	Ok
CP-004	Registro Usuario	Datos de usuario	Llenar todos los campos de forma correcta	Usuario registrado	Ok
CP-005	Registro Usuario	Datos de Usuario	Faltan datos en algún campo obligatorio	No se puede registrar sin llenar los campos obligatorios	Ok
CP-006	Registro Usuario	Datos de Usuario	Usuario y código ya registrados en el sistema	No se puede registrar con el mismo usuario o código	Ok
CP-007	Actualizar Usuario	Seleccionar un Usuario	Llenar o conservar todos los campos de forma correcta	Datos del usuario actualizados	Ok
CP-008	Eliminar Usuario	Seleccionar un Usuario	Contar con nivel de permiso “Administrador (ROOT)”	Usuario y sus datos relacionados como fichas y archivos generados eliminados	OK
CP-009	Registro Cliente	Datos de Cliente	Llenar todos los campos de forma correcta	Cliente registrado	Ok
CP-010	Registro Cliente	Datos de Cliente	Faltan datos en algún campo obligatorio	No se puede registrar sin llenar los campos obligatorios	Ok
CP-011	Registro Cliente	Datos de Cliente	Código ya registrado en el sistema	No se puede registrar con el mismo código	Ok
CP-012	Actualizar Cliente	Seleccionar un Cliente	Llenar o conservar todos los campos de forma correcta	Datos del cliente actualizados	Ok
CP-013	Eliminar Cliente	Seleccionar un Cliente	Contar con nivel de permiso “Administrador (ROOT)”	Cliente y sus datos relacionados como fichas y archivos generados eliminados	Ok
CP-014	Registro de Perfil Ficha	Datos del Perfil	Llenar todos los campos de forma correcta	Perfil registrado	Ok
CP-015	Registro de Perfil Ficha	Datos del Perfil	Faltan datos en algún campo obligatorio	No se puede registrar sin llenar los campos obligatorios	Ok
CP-016	Actualizar Perfil Ficha	Seleccionar un perfil de ficha	Llenar o conservar todos los campos de forma correcta	Datos del perfil de ficha actualizados	Ok

CP-017	Eliminar Perfil Ficha	Seleccionar un perfil de ficha	Contar con nivel de permiso "Administrador (ROOT)"	Perfil de ficha eliminado	Ok
CP-018	Consultar Clientes			Mostrar todos los clientes registrados en pantalla	Ok
CP-019	Consultar Usuarios			Mostrar todos los usuarios registrados en pantalla	Ok
CP-020	Consultar Perfiles			Mostrar todos los perfiles registrados en pantalla	Ok
CP-021	Generar Fichas <i>Hotspot</i>	Seleccionar un cliente y Número de fichas a generar de cada perfil	Cliente Activo en el sistema	Generación de fichas y del archivo PDF y TXT en pantalla	Ok
CP-022	Generar Fichas <i>Hotspot</i>	Seleccionar un cliente	Falta seleccionar la cantidad de fichas por perfil	No se puede proceder a la generación de fichas	Ok
CP-023	Generar Formato de Corte	Seleccionar un cliente y una fecha en la que se generó fichas	Calcular los datos e ingresar las fichas sobrantes	Generación del PDF de corte de fichas	Ok
CP-024	Consultar Fichas Generadas	Introducir una fecha, cliente y usuario		Mostrar las fichas en pantalla	Ok
CP-025	Eliminar Fichas Generadas	Seleccionar un grupo de fichas	Contar con nivel de permiso "Administrador (ROOT)"	Eliminación del grupo de fichas y de los archivos generados	Ok
CP-026	Consultar Archivos Generados	Introducir una fecha, cliente y usuario		Mostrar los archivos en pantalla	Ok
CP-027	Abrir Archivo Generado	Seleccionar un archivo		Mostrar el archivo en pantalla	Ok
CP-028	Eliminar Archivo Generado	Seleccionar un archivo	Contar con nivel de permiso "Administrador (ROOT)"	Eliminación del archivo y de archivos asociados o grupo de fichas	Ok
CP-029	Cambiar Valores de Conexión	Datos de la conexión	Datos correctos para la conexión	Conexión establecida correctamente	Ok
CP-030	Cambiar Valores de Conexión	Datos de la conexión	Datos erróneos de conexión	No es posible establecer la conexión	Ok

Fuente: Elaboración propia.

Como lo plantea [19] buscamos en nuestra implementación cumplir con los “los principios de seguridad y a la vez permita que el acceso a la información se realice de forma rápida y fácil”. El desarrollo del sistema, al igual que otros desarrollos, constituye un medio de trabajo para llevar un mejor registro de la información solicitada que persigue como objetivo favorecer la labor de los especialistas, reduciendo el tiempo de obtener los datos de una manera rápida [20].

Como buenas prácticas para el desarrollo de la aplicación: se emplearon clases dedicadas previendo el mantenimiento, todo está modulado, cada módulo hace su propia función y no depende de los demás, para la escalabilidad, cuenta con una clase para el acceso a datos, archivo de configuración que permite almacenar configuraciones nuevas, cuenta con una clase para actualizar que verifica la versión del sistema, se emplearon expresiones regulares en el módulo de perfiles, se redujeron los cuellos de botella en la programación para la optimización de software y uso de la memoria, se empleó la validación de entrada de datos en campos vacíos, se agregaron en el diseño objetos de selección para eliminar el error humano, se agregaron comentarios dentro del código a los métodos empleados, se crearon manuales de usuario y técnico.

#### 4. Resultados

La aplicación de escritorio que se implementó es de tipo *Windows Forms (WinForms)* y cuenta con los siguientes procesos de entrada y salida:

- Entrada: Ingresar en el software (acceso o *login*), Ingresar datos para el registro de usuarios administradores, Ingresar datos para el registro de clientes, Ingresar datos para el registro de perfiles de ficha, Ingresar datos para la solicitud de generación de formatos (Generación de Fichas PDF y Comandos, Formato de Corte, Exportación de BD).
- Salida: Consulta de Usuarios registrados, Consulta de Clientes registrados, Consulta de Perfiles de Ficha registrados, Consulta de Fichas Generadas, Consulta de Archivos Generados, Generar Formatos (Generación de Fichas PDF y Comandos, Formato de Corte, Exportación de BD, Archivo de Contraseñas Aleatorias).

En la Figura 5 se muestra la interfaz de inicio de sesión del sistema Generador de Fichas *Hotspot*, mediante esta ventana es posible iniciar sesión con un usuario registrado previamente en el sistema (RF-015).



Figura 5. Inicio de sesión.

En la Figura 6 se muestra la opción del filtrado de los datos y la opción del menú Ver que permite seleccionar la información que se desea mostrar en la ventana principal del sistema. Así como también se pueden visualizar las opciones del menú principal del sistema con las opciones de: Archivo, Usuarios, Clientes, Buscar, Herramientas y Ver.

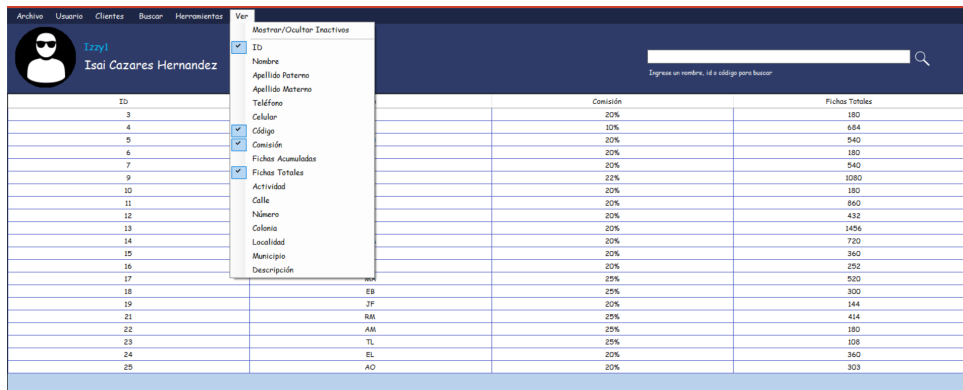
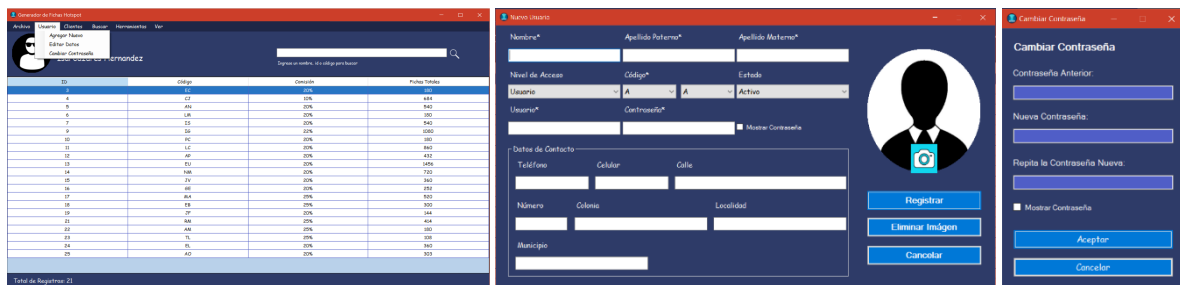


Figura 6. Ventana principal.

La Figura 7 muestra la opción del menú con las opciones del menú principal Usuario que permite Agregar Nuevo, Editar Datos y Cambiar Contraseña.



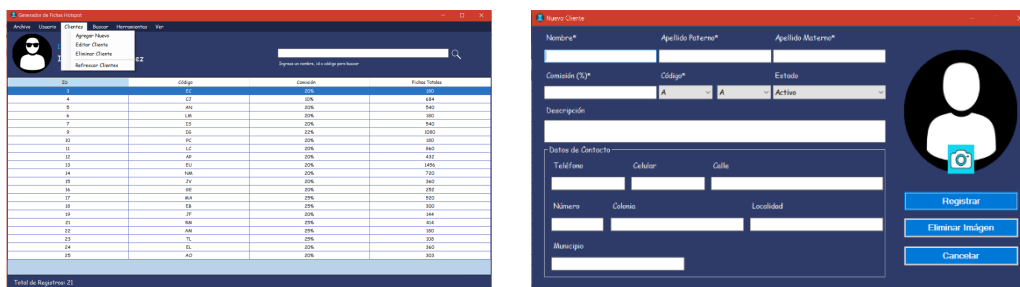
(a)

(b)

(c)

Figura 7. (a) Opción del menú principal Usuario; (b) Opción del menú principal de Usuario / Nuevo Usuario; (c) Opción del menú principal Usuario / Cambiar contraseña.

Las opciones del menú principal Clientes que permite agregar nuevo cliente, editar cliente, eliminar cliente y refrescar cliente, según se tenga autorización por el nivel de permiso asignado a los usuarios registrados se muestran en la Figura 8.



(a)

(b)

Figura 8. (a) Opción del menú principal Clientes; (b) submenú de Clientes / Nuevo Cliente.

La Figura 9 muestra las opciones del menú principal Archivo que permite tiene como submenús: Exportar Base de Datos, Importar Base de Datos, Configuración, Cerrar Sesión y Salir.

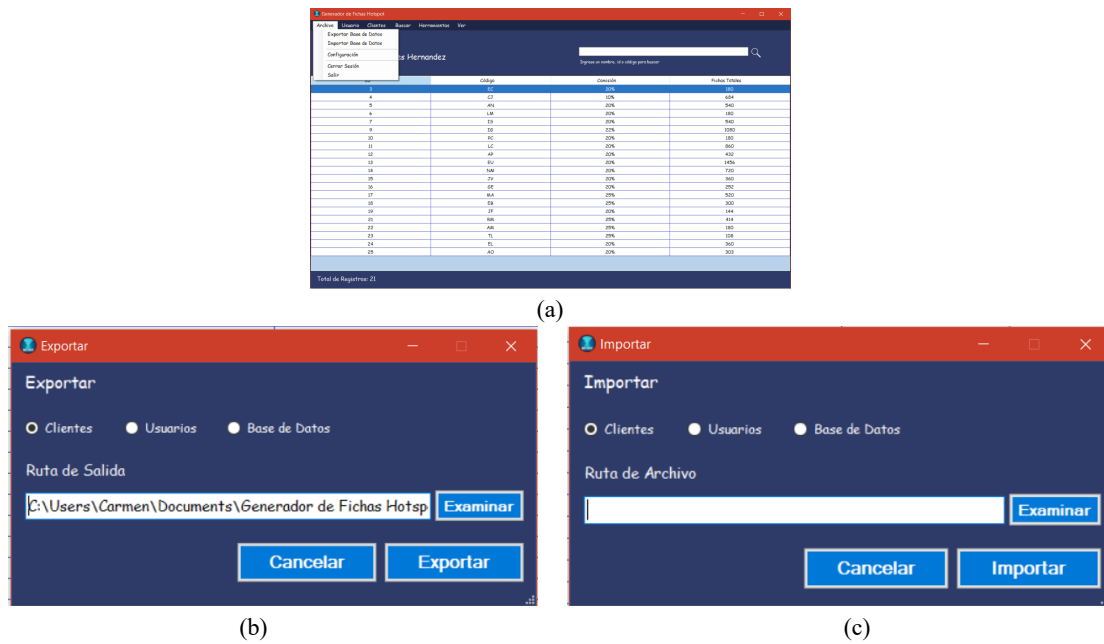


Figura 9. (a) Opción del menú principal Archivo; (b) Submenú de Archivo / Exportar; (c) Submenú del Archivo / Importar.

En la Figura 10 se muestran las opciones del menú principal Archivo del submenú Configuración desde aquí se puede modificar las carpetas de datos, a los usuarios administradores del sistema, los perfiles de ficha y los valores de conexión al servidor de base de datos. Además, dependiendo el nivel de permiso, permite depurar la base de datos para eliminar registros no deseados.



Figura 10. (a) Opción del menú principal Archivo/Configuración/Carpetas; (b) Opción del menú principal Archivo/Configuración/Usuarios; (c) Opción del menú principal Archivo/Configuración/Perfiles; (d) Opción del menú principal Archivo/Configuración/Depuración; (e) Opción del menú principal Archivo/Configuración/Conexión.

Las opciones del menú principal Herramientas, en el que se encuentra el módulo generador de fichas, el módulo generador de reportes de corte de fichas, el módulo administrador de fichas y el módulo administrador de archivos se pueden observar en la Figura 11.



Figura 11. (a) Opción del menú principal Herramientas; (b) Opción del submenú Herramientas/Generar Fichas; (c) Opción del submenú Herramientas/Corte de Fichas; (d) Opción del submenú Herramientas/Fichas Generadas; (e) Opción del submenú Herramientas/Gestor de Archivos.

Finalmente, en la Figura 12 se muestran los diferentes archivos generados por el sistema como son el pdf de fichas generadas, el archivo de texto con los comandos necesarios para cargarlos en el Mikrotik, y el archivo en pdf de los cortes de fichas generadas.

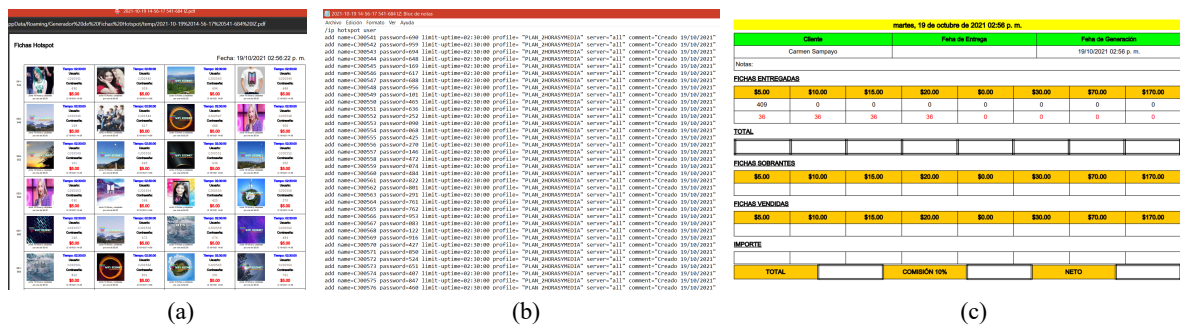


Figura 12. (a) Archivo generación de fichas en pdf; (b) Archivo comando de fichas en archivo de texto; (h) Archivo generado para corte de fichas.

### 5. Conclusiones

Durante la realización de este proyecto, se logró ampliar de forma significativa los conocimientos técnicos sobre las tecnologías y herramientas que se emplearon, sobre todo, al aplicar estos conocimientos en un software para un caso de la vida real. La utilización de herramientas de software libre hace que esta herramienta sea de fácil acceso y además fácil de mantener y escalar para poder optimizar, mejorar o añadir nuevos procesos.

Con respecto a la aplicación del sistema, las pruebas realizadas indican que cumple con las funciones necesarias para crear y administrar las cuentas de los usuarios, las de los clientes, y sobre todo la generación de fichas en muy poco tiempo y aminorando al máximo el error humano. Es importante mencionar que el sistema puede continuar creciendo y ofreciendo cada vez más funciones, permitiendo realizar un trabajo más efectivo y que no genere errores de ningún tipo, ofreciéndose un sistema efectivo para facilitar la conexión de la red en diferentes sectores, ya sean públicos o privados.

Los beneficios obtenidos con el uso del software desarrollado son: Se eliminó el pago de la licencia de office que se utilizaba para la hoja de cálculo. Se eliminó el trabajo laborioso de generar varias hojas con fichas en las que no se repitiera la contraseña y los usuarios. Se eliminó la duplicidad de fichas con igual clave y contraseña. Se estandarizó el nombre de usuario para cada cliente. De la forma en la que lo realizaban solo podían generar 5 hojas en formato pdf con 36 fichas cada una y después tenían que repetir el proceso y generarlo en otro archivo pdf esto para cada perfil: colocar un nombre de usuario que se inventaba en el momento y generar las contraseñas y verificar que no se repitan; con el uso del sistema no importa cuantas hojas generen y de que perfil se generen los genera en un solo archivo. La disminución en el tiempo de generación de las fichas, con sus usuarios y contraseñas, para los distintos perfiles y clientes.

El tiempo promedio empleado para generar fichas por hojas de 36 fichas por perfil utilizando una hoja de cálculo es de 420 segundos, esto de acuerdo con datos proporcionados por la empresa que proporcionó los requerimientos del sistema. Utilizando el sistema desarrollado, teniendo en cuenta que ya se autenticó el usuario y ya están realizadas las configuraciones de perfiles y el cliente ya está registrado, el tiempo promedio para generar una hoja con 36 fichas utilizando el sistema desarrollado baja a 21.15 segundos, de acuerdo con la Tabla 5. Se disminuyó el tiempo de generación de fichas en un 94.96%.

Tabla 5. Tiempo empleado para generar las fichas por hoja de 36 fichas.

Perfiles	Tiempo empleado para generar las fichas por hoja de 36 fichas						
	\$5	\$10	\$15	\$20	Día	Mes	Gratis
<b>Con el Software desarrollado</b>	26.32 seg	19.00 seg	16.22 seg	16.46 seg	24.52 seg	23.43 seg	22.20 seg

Fuente: Elaboración propia.

El siguiente paso para poder emplear el sistema, sería la generación de un autómatas que pueda recibir de forma directa las fichas generadas para así implementar de forma directa la conexión de los clientes automáticamente.

## 6. Agradecimientos

Se agradece al Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango por el apoyo brindado para llevar a cabo el proyecto y a la empresa que proporcionó los requerimientos del sistema.

## 7. Referencias

- [1] Schwaber, K., Sutherland, J. (2016). *La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*. Recuperado de: <https://scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf#zoom=100>
- [2] EcuRed. (2021). *Lenguaje de Programación C Sharp*. Recuperado de: [https://www.ecured.cu/Lenguaje\\_de\\_Programaci%C3%B3n\\_C\\_Sharp](https://www.ecured.cu/Lenguaje_de_Programaci%C3%B3n_C_Sharp)
- [3] Academic. (2021) *Los diccionarios y las enciclopedias sobre el Académico. IText*. Recuperado de: <https://es-academic.com/dic.nsf/eswiki/591978>
- [4] Microsoft. (2021). *Microsoft Download Center*. Recuperado de: <https://www.microsoft.com/es-ar/download/details.aspx?id=101064>
- [5] Santamaría, J., Hernández, J. (2021). *Microsoft SQL Server*. Recuperado de: <https://iessanvicente.com/colaboraciones/sqlserver.pdf>
- [6] Lee, T. G. (2021). *Información general sobre Visual Studio*. Recuperado de: <https://docs.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- [7] Siguenza Suscal, D. M., Jiménez Pesantez, J. P. (2014). *Estudio e implementación de la nueva arquitectura física y lógica de la red de datos, servicios utilizando RouterOS y tecnologías Open Source de bajo costo, integradas a un sistema de administración Web para control de abonados y gestión de planes de Internet para el proveedor de Servicios de Internet Inalámbricos Sigsignet* (Tesis de Grado). Universidad Politécnica Salesiana, Cuenca. Recuperado de: <http://dspace.ups.edu.ec/handle/123456789/6964>
- [8] Aguilar Alvarado, J. V., Quezada-Sarmiento, R., García-Galarza, K. (2018). *Aplicación Java para el control*



- de RB Mikrotik en empresas proveedoras de servicio de Internet//Java application for Mikrotik RB control in companies providing Internet service. *Ciencia Unemi*, 11 (26), 161-169. doi: <https://doi.org/10.29076/issn.2528-7737vol11iss26.2018pp161-169p>
- [9] SIA Mikrotikls. (2021). *MikroTik*. Recuperado de: <https://mikrotik.com/>
- [10]Cyber Alonso. (2021). *Internet por Fichas*. Recuperado de: <https://www.internetporfichas.com/index.html>
- [11]Tolentino, J. J., López, F. (2018). *VoucherHotspot*. Recuperado de: <https://www.voucherhotspot.net/index.html>
- [12]WispHub. (2021). *Generador de Fichas Hotspot Mikrotik*. Recuperado de: <https://wisphub.net/documentacion/generador-de-fichas-mikrotik-hotspot-27/>
- [13]Cyber Alonso. (2021). *Crear e Imprimir Fichas para Hotspot Mikrotik*. Recuperado de: <http://cyber-alonso.blogspot.com/>
- [14]Gavrysh, O. (2021). *Aplicaciones de escritorio modernas, ¿por qué?* Recuperado de: <https://docs.microsoft.com/es-es/dotnet/architecture/modernize-desktop/why-modern-applications>
- [15]Buschmann, F., Meunier, R., Rohnert, H., Sornmerlad, P., Stal, M. (1996). *Pattern-Oriented Software Architecture: A System of Patterns*. USA: John Wiley & Sons Ltd.
- [16]Medel Viltres, Y., Castro Dieguez, F., Figueredo León, A., Leyva Polo, A., Almaguel Guerra, A. (2021). Sistema informático para la distribución de uniforme escolar. Caso de estudio: provincia de Granma, Cuba. *Revista de Investigación en Tecnologías de la Información (RITI)*, 9 (19), 1-15. doi: <https://doi.org/10.36825/RITI.09.19.001>
- [17]Price, E., Buckgit, A., Peterson, N., Stanford, D., Boeglin, A., Albemi, V., Wasson, V. (2022). Migración de una aplicación monolítica a microservicios mediante el diseño controlado por dominios. Recuperado de: <https://docs.microsoft.com/es-es/azure/architecture/microservices/migrate-monolith>
- [18]Blancarte, O. (2021). *Arquitectura de Monolítico. Estilo arquitectónico*. Recuperado de: <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/monolitico>
- [19]Pérez Céspedes, A., Fonseca Mastrapa, O. (2018). Sistema informático para la gestión de la comercialización en la unidad empresarial de Base Frutas Selectas Las Tunas. *Revista de Investigación en Tecnologías de la Información (RITI)*, 6 (12), 94-99. Recuperado de: <https://www.riti.es/ojs2018/inicio/index.php/riti/article/view/126>
- [20]Batista Díaz, C., Lujo Aliaga, Z. (2017). Sistema informático para el control de los colaboradores en la dirección provincial de educación de Las Tunas. *Revista de Investigación en Tecnologías de la Información (RITI)*, 5 (9), 86-90. Recuperado de: <https://www.riti.es/ojs2018/inicio/index.php/riti/article/view/50>