UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERÍA EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIONES UNIFICADAS BASADO EN SOFTWARE LIBRE QUE INTEGRA CAPACIDADES DE VOZ SOBRE IP, MENSAJERÍA INSTANTÁNEA, FAX, MAIL, PBX Y UN SUBSISTEMA DE REGISTRO DE ASISTENCIA Y DESBLOQUEO DE CERRADURA POR HUELLA DACTILAR PARA LA EMPRESA VUELOFERTAS CIA. LTDA.

AUTOR: KLEVER JAVIER CHULDE CHULDE

TUTOR: Ing. Jorge Ismael Mera Gutiérrez, MSc.

AÑO: 2016

INFORME FINAL DE RESULTADOS DEL PIC

CARRERA:	**INGENIERÍA EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES**
AUTOR:	**CHULDE CHULDE KLEVER JAVIER**
TEMA DEL TT:	**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIONES UNIFICADAS BASADO EN SOFTWARE LIBRE QUE INTEGRE CAPACIDADES DE VOZ SOBRE IP, MENSAJERÍA INSTANTÁNEA, FAX, MAIL, PBX Y UN SUBSISTEMA DE REGISTRO DE ASISTENCIA Y DESBLOQUEO DE CERRADURA POR HUELLA DACTILAR PARA LA EMPRESA VUELOFERTAS CIA. LTDA.**
ARTICULACIÓN CON LA LÍNEA DE INVESTIGACIÓN INSTITUCIONAL:	**TECNOLOGÍA APLICADA A LA PRODUCCIÓN Y A LA SOCIEDAD**
SUBLÍNEA DE INVESTIGACIÓN INSTITUCIONAL:	**REDES TELEFÓNICAS BAJO TECNOLOGÍA IP**
ARTICULACIÓN CON EL PROYECTO DE INVESTIGACIÓN INSTITUCIONAL DEL ÁREA	**LA COMUNICACIÓN APLICADA AL BENEFICIO SOCIAL**
FECHA DE PRESENTACIÓN DEL INFORME FINAL:	**AGOSTO DE 2016**

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El presente proyecto tiene como objetivo principal la realización del diseño y la correspondiente implementación de un sistema de comunicaciones unificadas y también de un sistema de control de acceso. El sistema de comunicación que aglutina los servicios de PBX, Fax, correo electrónico, casillero de voz y mensajería instantánea establece una comunicación más productiva lo cual permite una adecuada y oportuna atención a los clientes a través de los diferentes canales y otorga opciones de comunicación interna entre los empleados de la empresa. El proyecto se complementa con un sistema de acceso que brinda seguridad a las instalaciones de la empresa el cual impide el ingreso de personas no autorizadas mediante la validación de huellas dactilares, además se mantiene un registro de la hora de entrada y salida de cada trabajador, al cual se puede acceder a través de un navegador con la dirección IP.

Las comunicaciones se implementan con el sistema de código abierto Elastix albergadas en un servidor dedicado y se dispone de interfaces FXO compatibles con líneas telefónicas analógicas. El sistema de control de acceso está diseñado en torno al microcontrolador Arduino el cual se encarga de gestionar la validación de las huellas digitales mediante dos lectores biométricos y además graba los registros en una memoria flash instalada en el módulo Ethernet.

Este informe tiene la siguiente estructura: en la Sección 1 se tiene la introducción, los antecedentes y objetivos del proyecto; en la Sección 2 se tiene la fundamentación teórica, el diagnóstico del problema y la fundamentación metodológica; en la Sección 3 se presenta el diseño y desarrollo de la implementación; En la sección 4 se encuentran el análisis de resultados.

DESCRIPTORES:

Comunicaciones unificadas

Central telefónica IP

Control de acceso con Arduino

Cerradura controlada con Arduino

ABSTRACT

The Project has a main objective that is the realization of the design and the corresponding implementation of a unified communication system and an access control system. The communication system that brings PBX capabilities, Fax service, voice mailbox, and instant messaging establishes a more productive communication allowing adequate and timely attention to customers through different channels and provides internal communication options among employees of the company. The project is complemented by an access system that provides security to company premises preventing unauthorized entry by validating fingerprint people, plus a record of the time of entry and exit of each worker is maintained, which you can be accessed through a browser by typing an IP address.

Communications are implemented with Elastix open source system hosted on a dedicated server and is available FXO interfaces compatible with analog phone lines. The access control system is designed around the Arduino microcontroller which manages the validation of two biometric fingerprints by scanners and also records the information in a flash memory installed in the Ethernet module.

This report is structured as follows: Section 1 is the introduction, the background and objectives of the project; Section 2 is the theoretical foundation, problem diagnosis and methodological foundation; in Section 3 the design and process of the implementation, in Section 4 the results achieved are presented.

DESCRIPTORS:

Unified communications

Ip telephone switchboard

Access control with Arduino

Controlled Lock with Arduino

1. INTRODUCCIÓN

1.1. Antecedentes

VUELOFERTAS C.L. es una agencia de viajes nacional e internacional que se dedica a la comercialización de paquetes turísticos y soluciones en todo lo relacionado a viajes y entretenimiento, sus operaciones iniciaron en el año 2007, las instalaciones se encuentran ubicadas en Quito en la Av. 12 de Octubre N24-529 y Cordero. Edificio Pallares. Pb. Oficina #5.

Este tipo de empresas utiliza muchas herramientas tecnológicas relacionadas con la ofimática y que por desconocimiento o falta de recursos para realizar inversiones no son debidamente actualizadas o usadas de una forma más eficiente con ello se refiere a la disminución de costos en la operación, tales como: menor consumo de insumos de papelería , tinta de impresión , factura de energía eléctrica, factura de líneas telefónicas, etc; además la ausencia de un sistema telefónico PBX dificulta la comunicación con los clientes, principalmente cuando son ellos quienes llaman a las líneas de la empresa y se escucha el tono de ocupado y que por esa circunstancia se pierden posibles negocios. Otra situación a tomar en cuenta es el control y registro de acceso a las oficinas de la empresa, que actualmente carece de un sistema que realice esta labor. A nivel general el propósito de este proyecto es que la empresa Vuelofertas C.L. mediante la implementación del mismo brinde una mejor atención al cliente y maximice el rendimiento de los recursos disponibles y que además las instalaciones tengan un sistema de seguridad en el acceso mediante un dispositivo biométrico.

1.2. Problema investigado

En la actualidad es imperativo mantener los canales de comunicación en forma óptima, especialmente si el giro de negocio se relaciona directamente con las ventas y la atención al cliente.

En este contexto la empresa VUELOFERTAS C. L. mantiene los servicios de correo, fax, mensajería instantánea, y telefonía de forma dispersa en diferentes aplicaciones, además actualmente el canal telefónico se volvió insuficiente para el volumen creciente de llamadas que se reciben, con lo que de acuerdo a la premisa inicial se hace necesario el diseño y la implementación de un sistema de comunicaciones unificadas cubra la función de central telefónica y que además tenga la capacidad de agrupar todos los servicios de comunicación que maneja la empresa. El sistema proveerá una solución en la que converjan el servicio telefónico tradicional con la

flexibilidad y escalabilidad de las comunicaciones basadas en internet como son: mensajería instantánea, fax, mail, pbx y casillero de voz.

Otra de las aristas que es necesario cubrir es el problema de la seguridad, es decir el control de acceso a las instalaciones debido a intrusiones no autorizadas y además el registro del personal en una base de datos que sirva tanto para validar el acceso como también mantener estadísticas de asistencia de los trabajadores.

1.3. Objetivo general

Implementar un sistema de comunicaciones unificadas en el que converja la telefonía tradicional con los servicios IP de mensajería instantánea, fax, mail y un subsistema complemento que permita el registro de asistencia y el desbloqueo de una cerradura eléctrica mediante huellas dactilares para la empresa Vuelofertas C. L.

1.4. Objetivos específicos

- Estudiar los requerimientos de la empresa para la implantación del sistema de comunicaciones unificadas y el subsistema de registro de asistencia con control de cerradura.
- Definir los medios a utilizar de software y hardware disponibles en el mercado que cumplan con los requerimientos de la implementación.
- Diseñar la solución con los elementos de hardware y software elegidos.
- Implementar el sistema de comunicaciones unificadas en la empresa Vuelofertas C.L.
- Implementar el subsistema de registro de asistencia y control de cerradura en las instalaciones de la empresa Vuelofertas C.L.
- Realizar pruebas de funcionamiento y corregir posible errores.

2. FUNDAMENTACIÓN TEÓRICA Y METODOLÓGICA

2.1. Redes de computadoras

Una red de computadoras son los dispositivos que se encuentran conectados a través de diferentes medios como son: cables de cobre, microondas, fibra óptica, señales luminosas etc. Que permiten la comunicación entre estos para compartir información digital. En el caso específico de este proyecto la mayor cantidad de información a transmitir proviene de la voz digitalizada que es parte del servicio de telefonía.

Para mantener un orden y estandarización en la forma de comunicarse las aplicaciones entre los diferentes equipos dentro de la red, se definió el modelo OSI conformado por siete capas y que cada una de ellas tengan funciones específicas, con el desarrollo en las técnicas de comunicación se instauró el modelo TCP/IP también conformado en capas y que toma como base el modelo OSI. (Jordi Griera, Barceló, Cerdà, & Peig, 2008)

2.1.1. Modelo TCP/IP

Es un conjunto de pautas en los protocolos de comunicación que permiten la transmisión de la información entre diferentes sistemas, este modelo a diferencia de OSI cuenta con cuatro capas que mantienen una jerarquía y funciones determinadas, estas capas son: (Philippe Atelin, 2007)

Capa de acceso de red (Capa 1)

Es la capa que maneja el enlace físico entre los dispositivos de red, es el caso de los módems, las tarjetas Ethernet etc., prácticamente permite la interacción entre los elementos de hardware y los medios de transmisión. Las funciones de la capa de acceso son la asignación de direcciones IP a las direcciones físicas, encapsulamiento de paquetes IP en tramas. (Dordoigne & Atelin, 2006)

Capa de internet (Capa 2)

Las funciones de esta capa son la selección de la mejor ruta para encaminar los paquetes de datos a través de la red, definir un esquema de direccionamiento y la transferencia de los datos entre la capa de acceso a la red y la de internet. (Dordoigne & Atelin, 2006)

Capa de transporte (Capa 3)

En esta capa los datos son segmentados para enviarse a través de la nube (internet), y brindar de esta manera una conexión lógica entre el host transmisor y el receptor, es decir entre los puntos finales de una red. (Dordoigne & Atelin, 2006)

Capa de aplicación (Capa 4)

Las funciones de esta capa son el manejo de la presentación, codificación y control de dialogo, que en el modelo OSI se lo representa en 3 niveles capa 5 (sesión), capa 6 (presentación), capa 7 (aplicación).

Estos son los conceptos principales que se debe tener en cuenta para implantar una red funcional y que el objetivo de comunicar y transportar la información se cumpla satisfactoriamente. Adicionalmente es necesario conocer la definición de los tipos de redes involucradas en este proyecto. (Dordoigne & Atelin, 2006)

2.2. Tipos de redes según su área de acción

Red de área personal (PAN)

Es una red conformada por los dispositivos periféricos que se comunican con una computadora, estos pueden ser un teclado, un mouse, auriculares, teléfonos inteligentes, que a menudo se vinculan inalámbricamente mediante Bluetooth. En la figura 2.1 se muestra los dispositivos. (www.informatica-hoy.com.ar, 2015)

Figura 2.1 Red de área personal
Fuente: http://www.informatica-hoy.com.ar

Red de área local (LAN)

Es una red que físicamente se limita a un área usualmente pequeña como es una oficina, una casa, o un edificio. Si se tiene un gran número de dispositivos se acostumbra a dividirlo en segmentos lógicos denominados Workgroups, en la Figura 2.2 se tiene un ejemplo de red local. (Andrew S., 2003)

Figura 2.2 Red de área local

Fuente: http://www.ieaisa.es

Red de área extendida (WAN)

Son agrupaciones de computadoras o conjuntos de redes de área local que se comunican con otros grupos que están difundidos en una localidad geográfica mucho más grande, para este caso se habla de países o continentes, ver Figura 2.3 . (Andrew S., 2003)

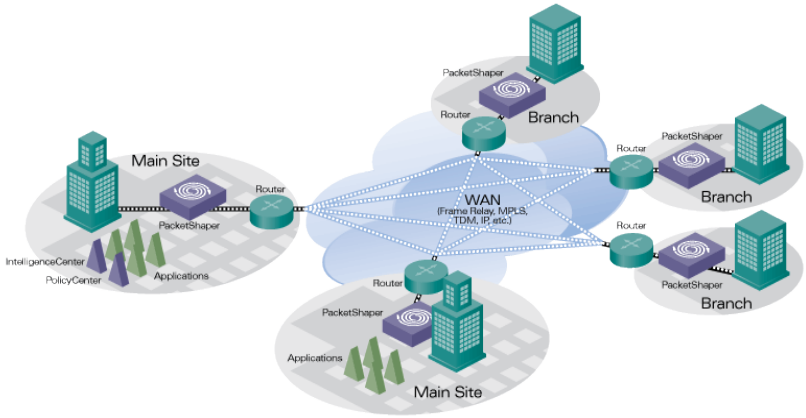


Figura 2.3 Red de área extendida (WAN)

Fuente: https://bto.bluecoat.com/

2.2.1. VOIP

La palabra VoIP viene del inglés Voice Over Internet Protocol, que quiere decir “voz sobre un protocolo de internet”, lo cual en esencia significa que la voz es digitalizada y codificada para transportarse a través de la red de datos e internet, esto puede realizarse en tiempo real en varias aplicaciones como televigilancia o teleasistencia. (Carballar, 2008)

2.2.2. Arquitectura de una red para Voz IP

Uno de los beneficios de la telefonía IP, es permitir la configuración de las redes con una arquitectura centralizada o distribuida, Lo que deja a las empresas en la capacidad de construir una red con administración simplificada.

Arquitectura centralizada.

En forma general se caracteriza por utilizar los protocolos MGCP y MEGACO, que fueron diseñados para su uso con un equipo central llamado Controlador de Pasarela de medios (Media Gateway Controller) , que se encarga de la lógica de conmutación y control de llamadas. En esta configuración los dispositivos finales de usuario mantienen funciones y características limitadas.

Arquitectura Distribuida

Está asociada con los protocolos H.323 y SIP, estos protocolos hacen posible que la inteligencia de la red se reparta entre los dispositivos de control de llamadas y los terminales en sí. Con ello se refiere a establecer llamadas, enrutamiento, aprovisionamiento, o cualquier tipo de manejo de llamadas.

2.3. Sistemas telefónicos IP-PBX de distribución libre

2.3.1. Trixbox

Es un sistema operativo basado en CENTOS cuya función de central telefónica se encuentra optimizada y además sus raíces son tomadas de la conocida PBX por software Asterisk. Actualmente la distribución de la versión profesional se encuentra a cargo de la empresa Fonailty que al ser ediciones comerciales incluyen varios módulos que complementan la solución básica de PBX. (Fonality Inc., 2016)

2.3.2. Elastix

Es un sistema de código abierto también basado en Centos cuya característica es reunir lo mejor de varios sistemas para obtener un solo software de comunicaciones unificadas. Esta distribución se compone de aplicaciones específicas como Asterisk,

Hylafax, Openfire y Postfix con los que gestiona PBX, FAX, mensajería instantánea y correo electrónico. (PaloSanto Solutions, 2016)

2.3.3. Asterisk

Es un sistema de distribución libre cuyo principal propósito es la función de central PBX con capacidad de establecer comunicaciones entre los terminales adheridos directamente o realizar conexiones hacia un proveedor de VoIP o hacia la red telefónica normal. (Digium Inc., 2016)

2.4. Equipos y aplicaciones compatibles con telefonía IP

La implementación del servicio de telefonía implica que los dispositivos y las aplicaciones que se utilizarán deben tener las capacidades para establecer la comunicación dentro la red de computadores, por ello se hace necesario conocer los elementos de hardware y software que conforman el sistema.

2.4.1. Teléfonos IP

En esencia son dispositivos que permiten transmitir paquetes de voz y/o video a través de una red de datos y que tienen la forma de un teléfono convencional pero con más características de funcionamiento y administración mediante un servidor, en la Figura 2.4 se muestra un teléfono con capacidad de video llamada.

Figura 2.4 Teléfono IP
Fuente: Grandstream.com

2.4.2. Softphones

Es una aplicación de software (programa) que se instala en un ordenador y con el manejo de protocolos adecuados en un entorno de voz sobre IP tiene las capacidades de un teléfono IP, en la figura 2.5 se ilustra la aplicación 3CX.

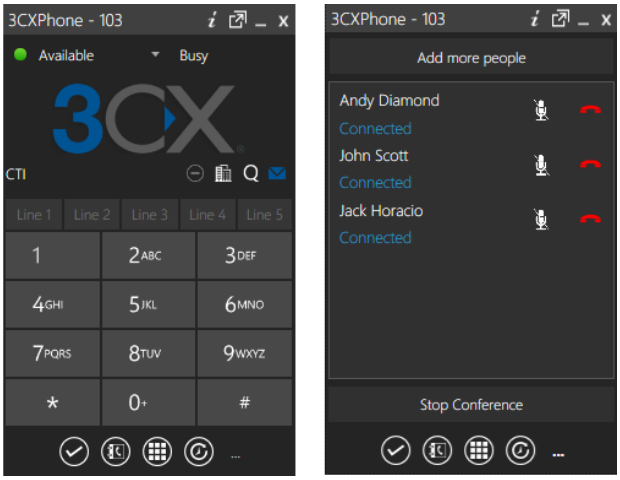


Figura 2.5 Softphone 3CX

Fuente: www.3cx.com

2.4.3. Adaptador para teléfonos analógicos

Es un elemento que permite conectar un teléfono analógico normal a un entorno de voz sobre IP, en la Figura 2.6 se ve el dispositivo que se antepone al teléfono tradicional para conectarlo al entorno IP.

Figura 2.6 Adaptador VoIP para teléfonos analógicos

Fuente: www.cisco.com

2.5. Sensores

Sensor es un dispositivo que está en capacidad de detectar acciones o estímulos externos y responder debido a estos, las alteraciones físicas o químicas detectadas tienen como consecuencia, a través del sensor, una respuesta eléctrica, que con el adecuado acondicionamiento la señal permitirá utilizarse en los sistemas de control automático. (Cientificas, 1987)

Lector de huellas dactilares

Las huellas digitales en los humanos son como una tarjeta de identificación integrada, son accesibles y los diseños son virtualmente únicos, por lo tanto se hizo un uso muy diverso en la validación de credenciales para permitir el acceso a diferentes objetos, lugares, aparatos etc. etc.

Un lector de huella digital lleva a cabo dos tareas:

1. Obtener una imagen de la huella digital, y
2. Comparar el patrón de esa imagen con los patrones de huellas almacenadas en la base de datos.

Para obtener la imagen de la huella existen dos métodos que son los más comunes, mediante lectores ópticos y lectores capacitivos.

2.5.1. Lectores Ópticos

Funcionan con un dispositivo CCD (Charged Coupled Device) como el que contienen las cámaras fotográficas digitales, cada diodo sensor graba un pixel, así en conjunto la luz y las zonas oscuras forman imagen de la huella leída, esto se ilustra en la figura 2.7.

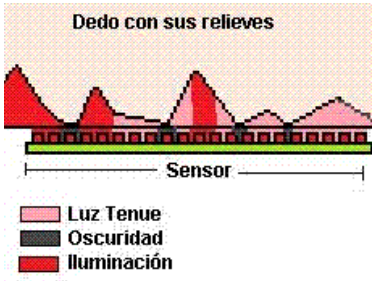


Figura 2.7 Funcionamiento de lector óptico
Fuente: http://www.informaticamoderna.com

2.5.2. Lectores Capacitivos

Los lectores capacitivos de huella digital generan una imagen de las crestas y valles que conforman una huella digital, mediante la corriente eléctrica que utilizan los capacitores. La Figura 2.8 muestra un ejemplo de sensor capacitivo. El sensor está hecho de uno o más chips que contienen un arreglo de pequeñas celdas. Cada celda incluye dos placas conductoras, cubiertas con una capa aislante.

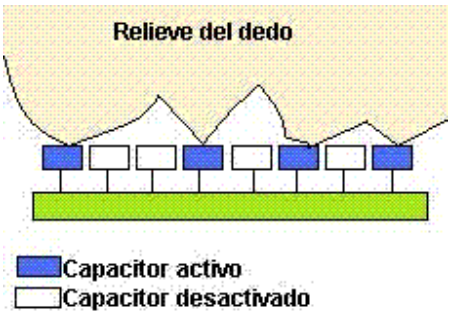


Figura 2.8 Funcionamiento de lector capacitivo
Fuente: http://www.informaticamoderna.com

2.6. Microcontroladores basados en hardware libre

El Microcontrolador

Es un circuito integrado en cuyo interior tiene una CPU (unidad de procesamiento), memorias RAM y ROM, puertos de entrada y salida y periféricos, obsérvese Figura 2.9. Todas estas unidades se encuentran interconectadas dentro del microcontrolador. El propósito fundamental de los microcontroladores es leer y ejecutar los programas que el usuario escribe, es por esto que la programación es una actividad básica e indispensable cuando se diseñan circuitos y sistemas que los incluyan. (Bonilla, 2005)

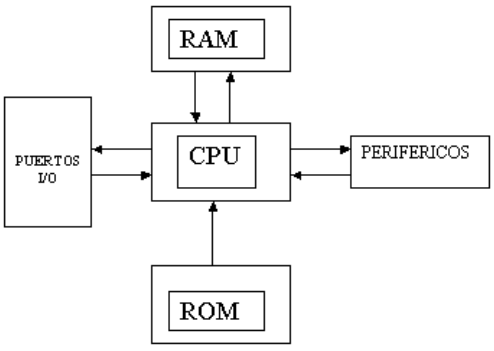


Figura 2.9 Esquema de un microcontrolador
Fuente: http://www.electronicaestudio.com

Microcontrolador Arduino

Es una plataforma enfocada en la realización de proyectos prototipo y basada en el uso fácil tanto de hardware como software. Existe una variedad de módulos compatibles con las placas principales que contienen el microcontrolador y que se las puede interconectar de acuerdo al uso que se vaya a dar al proyecto. El lenguaje de programación está basado en WIRING, que es una estructura de programación dedicada a microcontroladores.

2.7. Control de acceso

2.7.1. Cerraduras eléctricas y electrónicas

La cerradura es un dispositivo que se coloca en puertas, cajones, escotillas, etc., para trabar la apertura de estas y con ello proteger el contenido de su interior o en el caso de este proyecto mantener el control de acceso de personal a las instalaciones de la empresa. Existen varios tipos de cerraduras, pero los que competen al proyecto son exclusivamente cerraduras eléctricas y electrónicas. Las cerraduras eléctricas tienen un mecanismo electromagnético, al estar energizadas simplemente funciona como un potente magneto y no permite separar las placas metálicas ancladas a la puerta, para desbloquear la puerta es necesario cortar la corriente eléctrica que alimenta ese electroimán, en este caso la interrupción de energía se inicia al presionar un pulsador o por otro dispositivo autónomo que mediante una validación de credenciales autoriza o no la apertura de la cerradura y por ende de la puerta. La figura 2.10 muestra la cerradura electromagnética anclada con sus herrajes.

Figura 2.10 Cerradura electromagnética
Fuente: https://bdscerrajeros.files.wordpress.com

Las cerraduras electrónicas tienen una estructura parecida a las eléctricas pero además de ello mantienen un sistema de validación incorporado en sus carcasas, es decir, que no necesitan dispositivos adicionales para la verificación de credenciales como pueden ser huellas dactilares, una clave numérica, una tarjeta inteligente etc. Ver Figura 2.11

Figura 2.11 Cerradura electrónica
Fuente: http://www.segurhogarsa.es

2.8. Diagnóstico del problema y descripción del proceso investigativo realizado.

Una vez revisados los conceptos y teoría pertinente al desarrollo del proyecto es necesario centrarse en el diagnóstico del problema, y en este caso se manejan dos:

El primer problema es que la empresa no tiene un sistema que aglutine los canales de comunicación que maneja para el desempeño de sus funciones y el segundo es la ausencia completa de sistemas de seguridad que permita controlar el acceso a las instalaciones de Vuelofertas. En tal sentido los objetivos trazados van encaminados a cambiar esa realidad, es decir, proveer a la empresa de un sistema de comunicaciones unificadas y también de un sistema de control de acceso a las oficinas mediante la validación de huellas dactilares.

La hipótesis planteada es “la implementación de los dos sistemas permitirá que se disminuya el número de clientes que no son atendidos, además de mejorar la comunicación interna y proveerá de seguridad a los trabajadores de la empresa”. Con lo que se tiene las variables independiente y dependiente.

Variable independiente: Sistema de comunicaciones unificadas y sistema de control de acceso.

Variable dependiente: Satisfacción en los clientes y bienestar de los empleados de la empresa.

2.9. Metodología de la investigación

Análisis y síntesis.- Este método se utilizó para encontrar información sobre los dispositivos que conforman los sistemas a implementarse.

Investigación descriptiva.- Se recopiló información sobre los sistemas y canales de comunicación que tiene la empresa y su utilización en el desempeño de sus funciones.

Método de modelación.- Con la fundamentación teórica obtenida se tiene la información y el criterio para generar diseños en la red de comunicaciones y diseño en la conformación del circuito en el sistema de control de acceso.

Método experimental.- Por la naturaleza de ser proyectos pilotos o prototipos siempre se encuentra errores y fallas que deberán solventarse mediante este método, es decir, prueba y error; hasta que su funcionamiento sea el esperado.

3. DISEÑO, INSTALACIÓN Y PRUEBAS

3.1. Situación actual de los servicios de comunicación

Actualmente la empresa de turismo VUELOFERTAS C.L mantiene en sus instalaciones infraestructura y servicios de comunicaciones que se detallan a continuación:

- Existen tres líneas telefónicas de tipo analógico provistas por los operadores CNT (dos líneas) y CABLEMODEM (una línea). El acceso a estas líneas se lo realiza mediante splitters (divisores de línea "T"l) y son utilizadas indistintamente por el trabajador que lo requiera, esto se observa en la figura 3.1.

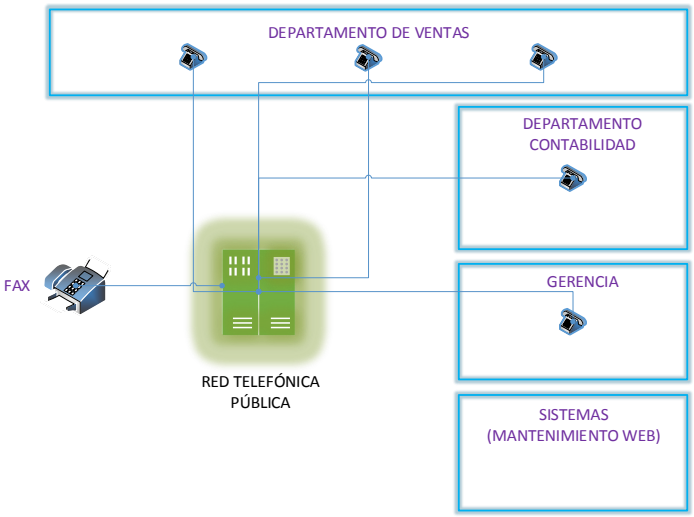


Figura 3.1 Red telefónica analógica

Fuente: El autor

- Servicio de internet de alta velocidad mediante cable coaxial y modem para la conexión mediante un router con acceso inalámbrico (WIFI). La figura 3.2 muestra la conexión a internet.

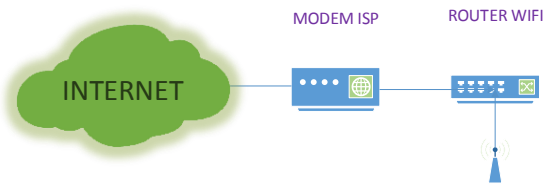


Figura 3.2 Servicio de Internet

Fuente: El autor

- Las computadoras se encuentran conectadas a un switch en una topología denominada estrella, como se muestra en la Figura 3.3.

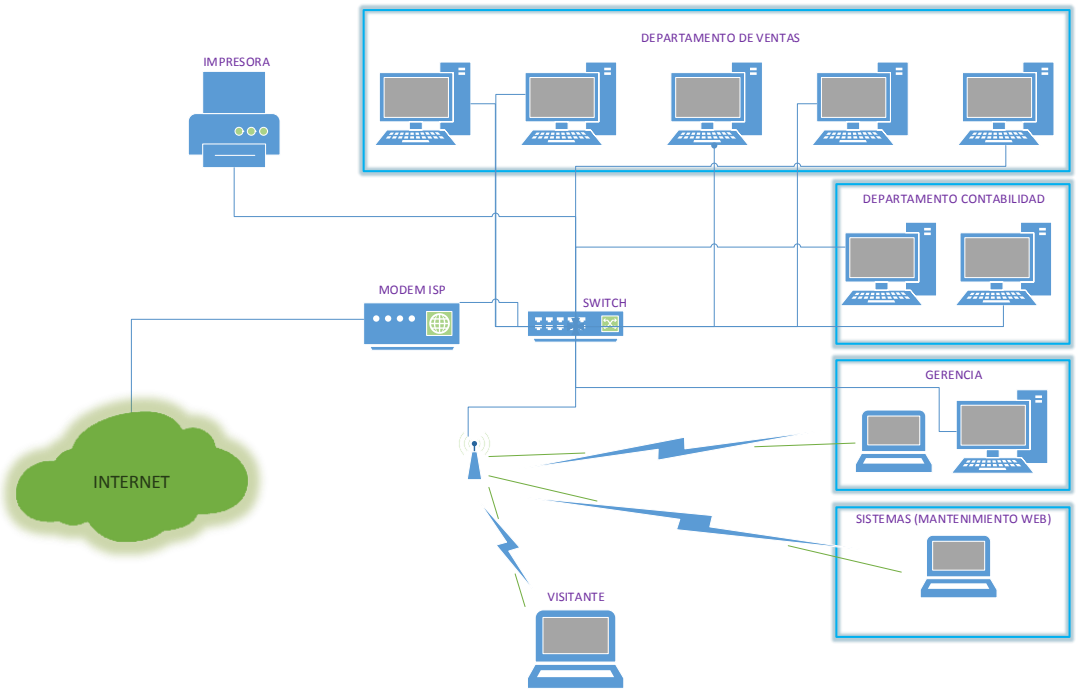


Figura 3.3 Red de computadoras en Vuelofertas

Fuente: El autor

- Tanto la página web como los correos se encuentran alojados en el proveedor de dominios y servicio de hosting GO DADDY.
- En el servicio de fax que aún se maneja para ciertos trabajos de la empresa se lo realiza mediante un teléfono-fax analógico (ver Figura 3.4) que se conecta a una de las líneas analógicas y los documentos se imprimen directamente en papel térmico.

Figura 3.4 Teléfono con capacidad de enviar Fax

Fuente: El autor

- En lo referente a sistemas de seguridad, ya sea activa o pasiva la empresa prácticamente no tiene ninguno, además tampoco posee dispositivos o sistemas de registro que permitan controlar la asistencia y acceso del personal a las oficinas de la compañía.

3.2. Comparación de los sistemas telefónicos IP-PBX de distribución libre.

Los sistemas de Telefonía IP de mayor difusión son los indicados en el marco teórico y con los que se realiza una comparativa para dilucidar la mejor opción en el desarrollo del proyecto. A continuación se tiene la tabla 3.1 en la que se recopila información de relevancia que será analizada.

Tabla 3.1 Sistemas de telefonía IP

Sistema	SERVICIOS	SOPORTE	COSTO
TRIXBOX	CENTRAL IP-PBX	Medio	Gratuito (versión básica)
	IVR		
	Buzón de voz		
ELASTIX	Telefonía IP	Alto	Gratuito
	Fax		
	Mensajería		
	PBX		
	IVR		
	Buzón de voz		
ASTERISK	Telefonía IP	Medio	Gratuito
	PBX		
	IVR		

Fuente: el autor

De acuerdo a las características mostradas en la tabla, la opción que se escoge es el sistema Elastix, que además cabe tomar en cuenta el gran soporte y ayuda que se puede encontrar a través de foros y blogs, también al ser de código abierto existen varios módulos complementos que son de gran utilidad como por ejemplo el servicio de VPN que de hecho se lo utilizará en este proyecto y que permite la autenticación de usuarios que se encuentran fuera de la red local o LAN.

3.3. Requisitos que el sistema de comunicaciones debe cumplir.

1. El sistema operativo debe provenir de una versión estable que tenga suficiente soporte técnico.

2. El sistema debe ser gratuito y basado en software libre.
3. Debe tener la capacidad de gestionar el servicio telefónico, realizar funciones de PBX, mensajería instantánea, mantener una contestadora automática, manejar correos electrónicos, máquina de fax y servicio de correo de voz.

3.4. Diseño del sistema de comunicaciones unificadas

Una vez conocidos los antecedentes técnicos y la situación de la empresa, es posible desarrollar una solución que se ajuste a sus requerimientos y necesidades, para lo cual se definió un sistema de comunicaciones que unifique todos los servicios que la empresa utiliza en su giro de negocio y un sistema de registro y control de acceso que sea fácil de manejar y brinde la seguridad necesaria a los trabajadores. Entonces como primera fase se iniciará con el diseño del sistema de comunicación.

3.5. Diagrama de propuesta de solución

Debido a las características que presenta cada uno de los sistemas de central IP-PBX analizados en punto 3.2, se optó por el sistema ELASTIX de distribución gratuita de la compañía PALO SANTO SOLUTIONS, y que es el sistema operativo que aglutina todas las funcionalidades que requiere la empresa Vuelofertas C.L. además de ofrecer diversos tipos de ayuda y soporte tanto en libros como en internet, la propuesta se ilustra en la imagen 3.5.

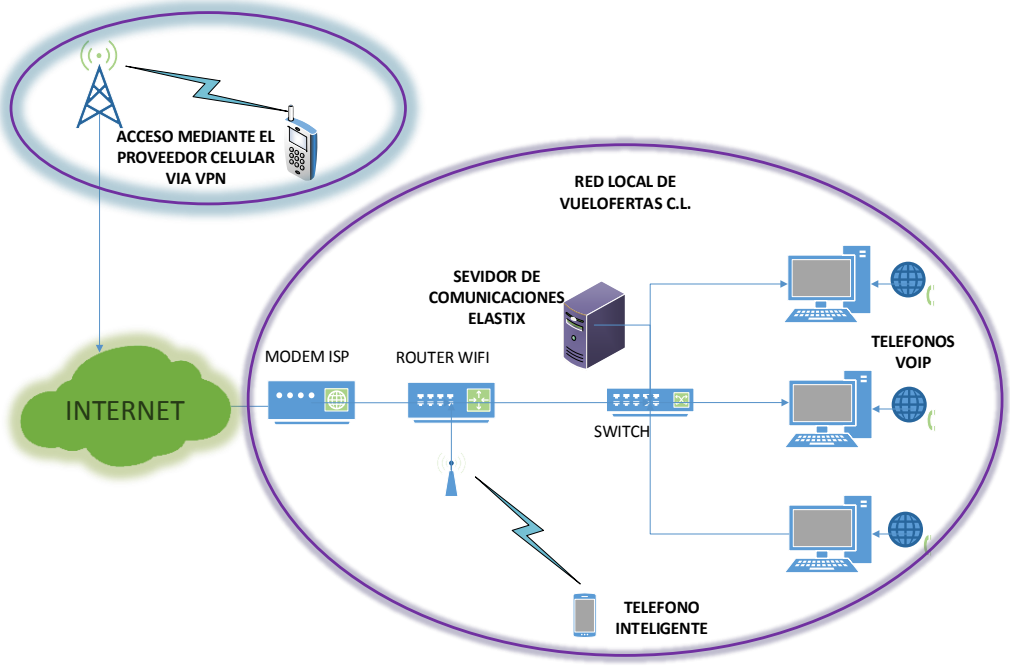


Figura 3.5 Propuesta del sistema de comunicación

Fuente: El autor

3.6. Implementación del sistema de comunicaciones.

Los requerimientos en la instalación del sistema Elastix no son muy estrictos, es decir se puede utilizar una computadora que ya se encuentre en desuso y utilizarla en la instalación que se necesita, lo mínimo es tener un procesador de la gama Pentium, 512 MB de memoria RAM, un disco duro de 40 GB y que tenga una tarjeta de red Ethernet. Este tipo de configuración es suficiente para manejar hasta veinte extensiones, lo cual es perfecto en este proyecto.

Para empresas que necesiten mayor número de extensiones es necesario implantar un servidor dedicado con mayores prestaciones de hardware.

En la tabla 3.2 se tabula el hardware utilizado y su montaje.

Tabla 3.2 Hardware del servidor

Descripción	Capacidad	Arquitectura
CPU + MBO	2,3 GHz	64 Bits
Memoria RAM	1 GB	
Disco duro	60 GB	
Lector óptico		

Fuente: El autor

Como se mencionó el hardware utilizado se ensambla de partes recicladas lo cual contribuye al bajo costo de inversión en el proyecto. A seguir se tiene las etapas para el ensamblaje e instalación del servidor

1. Montaje de mainboard y procesador. Figura 3.6 y Figura 3.7

Figura 3.6 Mainboard del servidor

Fuente: El autor

Figura 3.7 Procesador del servidor

Fuente: El autor

2. Ya instalado el módulo FXO para conectar las líneas telefónicas se emplaza el case que contiene todos los componentes del servidor, ver Figura 3.8

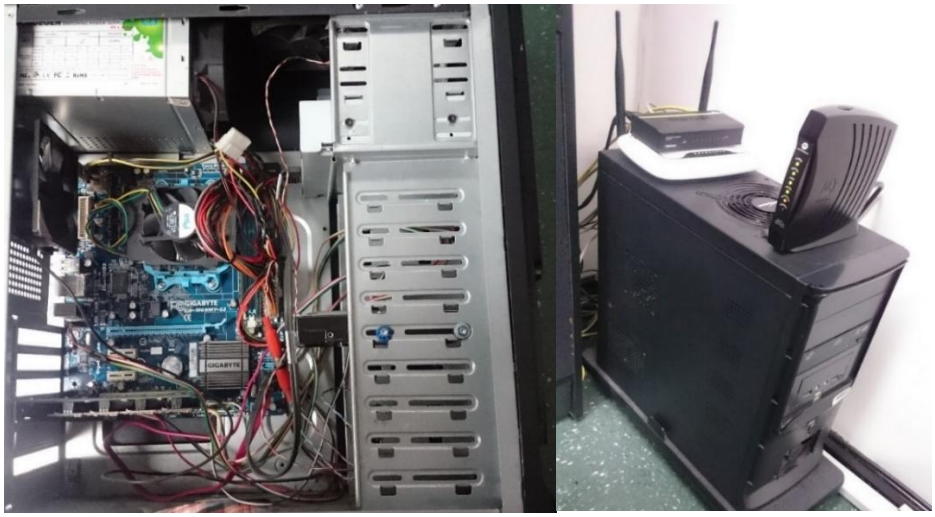


Figura 3.8 Ensamblaje de servidor Elastix

Fuente: El autor

3.7. Instalación y configuraciones del servidor de comunicaciones

Una vez ensamblada la computadora que hará las funciones de servidor se prosigue con la instalación de la versión estable adecuada al tipo de hardware que se tiene, ya sea 32 o 64 Bits. En este caso se instala la versión estable 2.5.0 de 64 bits que se la descarga gratuitamente de su página web oficial

http://www.elastix.com/downloads/, y con los pasos descritos en la guía de instalación se tiene el servidor listo para configurarse mediante el acceso web del que dispone. La imagen 3.9 muestra el proceso de instalación del sistema operativo.

Figura 3.9 Instalación de Elastix
Fuente: El autor

La configuración de red se la realizó con las siguientes consideraciones:

- El servicio de asignación de direcciones IP automáticas (DHCP) se encuentra deshabilitado, debido a que el router del proveedor de internet ya lo tiene activado.
- La dirección IP del servidor 192.168.0.10 se encuentra fuera del rango de asignación dinámico para evitar direccionamiento duplicado.

3.8. Servicio FAX

Con el avance de la técnica, especialmente en telecomunicaciones, el uso del fax como medio de comunicación disminuyó paulatinamente, a pesar de ello en este proyecto se realizará las configuraciones con el fin de asegurar la disponibilidad del servicio de fax que los clientes o proveedores requirieran. Así se tiene los siguientes pasos.

1. Se inicia con la creación de una extensión IAX para asociar la máquina de fax. A continuación se configura el servicio de fax con la consideración de los parámetros de código de país y código de área, ver Figura 3.10.

Figura 3.10 Configuración cuenta de fax

Fuente: El autor

2. En la casilla Fax Maestro se configura el correo electrónico donde se recibirá las notificaciones y faxes externos que están dirigidos a la empresa. En la Figura 3.11 se observa la cuenta de FAX habilitada para su funcionamiento en la extensión 112.

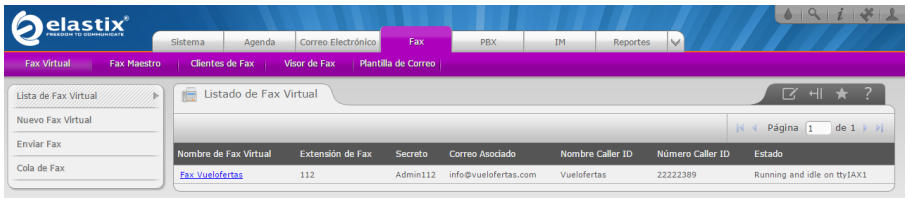


Figura 3.11 Cuenta de fax habilitada

Fuente: El autor

3.9. Servicio PBX

Antes de iniciar con la distribución de las extensiones se configura las rutas tanto para las llamadas entrantes como las salientes, es decir, se definen las reglas con las que el servidor canalizará todo el tráfico telefónico. De este modo según requerimientos de la empresa las reglas de marcado están definidas como sigue, ver Figura 3.12, si es necesario marcar a líneas externas se antepone el 9 y luego se digita el número al que se quiere llamar, como se indica en la Figura 3.13.

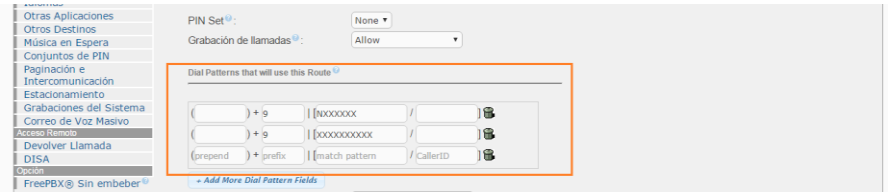


Figura 3.12 Reglas de marcado

Fuente: El autor

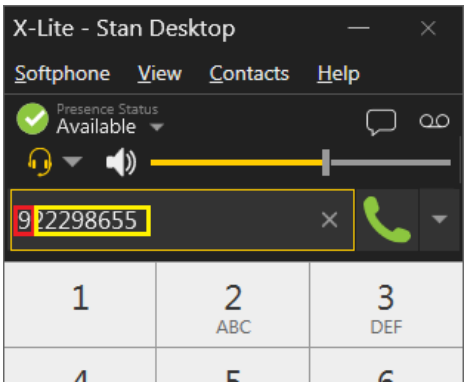


Figura 3.13 Marcado a la red pública

Fuente: El autor

Para estas configuraciones se hace necesaria cierta organización y distribuir de la mejor manera las extensiones con sus respectivos nombres como se muestra en la Tabla 3.3.

Tabla 3.3 Distribución y organización de extensiones

Descripción	Extensión	Acceso	Observación
Mafer	101	Fija y móvil	Extensión SIP
Stan	102	Fija y móvil	Extensión SIP
Germ	103	Fija y móvil	Extensión SIP
Contab	104	Fija y móvil	Extensión SIP
Ventas01	105	Fija y móvil	Extensión SIP
Ventas02	106	Fija y móvil	Extensión SIP
Ventas03	107	Fija y móvil	Extensión SIP
Ventas04	108	Fija y móvil	Extensión SIP
Ventas05	109	Fija y móvil	Extensión SIP
Admin	110	Fija y móvil	Extensión SIP
Gerencia01	150	Fija y móvil	Extensión directa externa
Gerencia02	160	Fija y móvil	Extensión directa externa
Gerencia03	170	Fija y móvil	Extensión directa externa

Fuente: El autor

Ya con la información organizada se procede a configurar cada extensión así:

1. Se selecciona el tipo de extensión, en este caso son SIP y luego se configura el número de la extensión junto con un alias que lo identifique dentro del grupo de extensiones, en la figura 3.14 se muestra los campos a completarse.

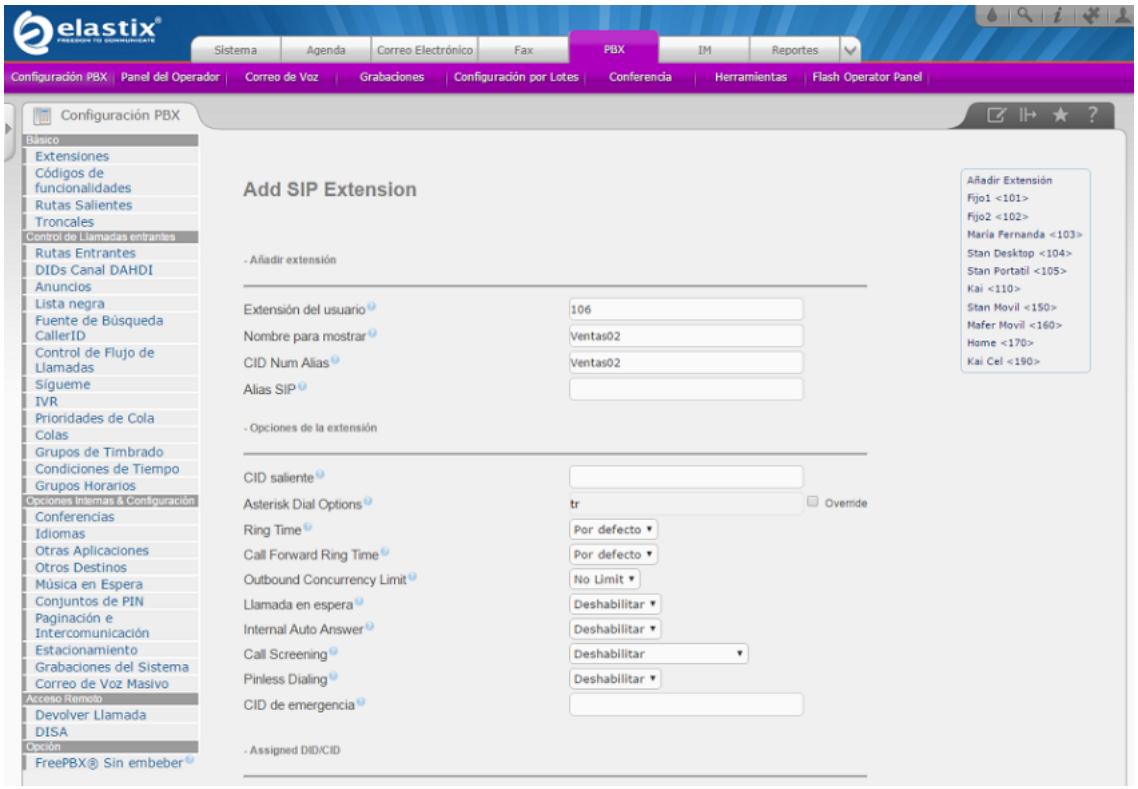


Figura 3.14 Configuración de extensión SIP

Fuente: El autor

2. Se quita la clave generada por defecto debido a su complejidad y se escribe el password que será único para que el servidor autentique esa extensión y permita la comunicación.

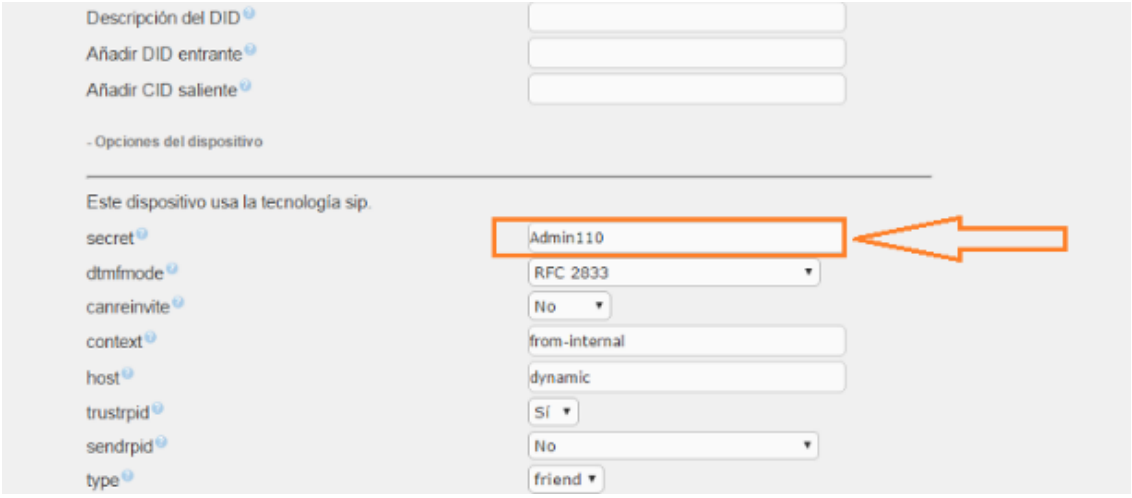


Figura 3.15 Configuración de la clave en la extensión

Fuente: El autor

3. Culminada la configuración de cada extensión se puede observar en la parte superior derecha todos los usuarios que tiene el grupo (ver Figura 3.16), para activar los servicios es necesario que el administrador tenga asociada una de las extensiones, en este caso se toma la extensión 110.

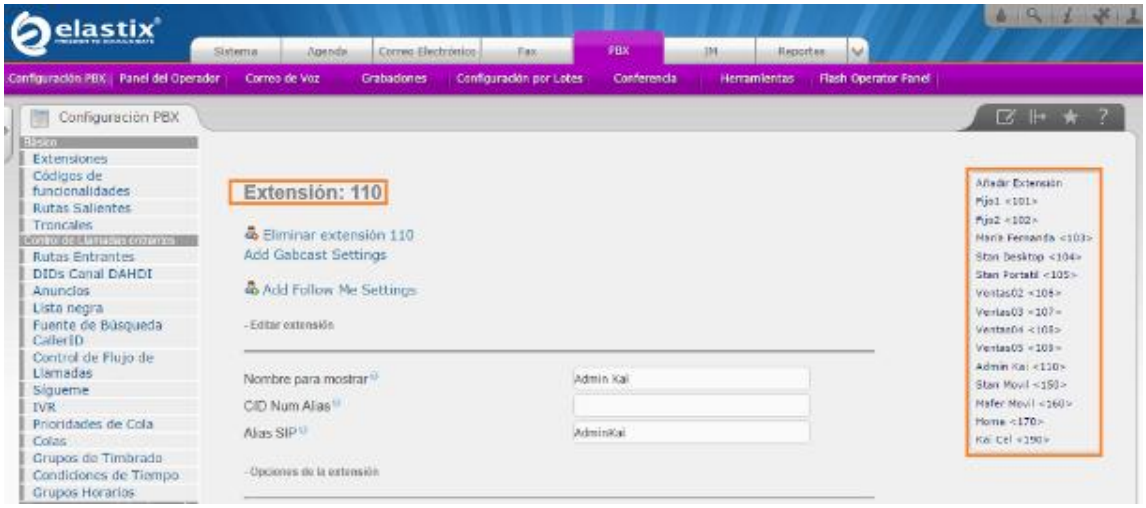


Figura 3.16 Extensiones SIP e IAX
Fuente: El autor

3.10. Servicio de telefonía IP

Este punto se refiere a la configuración de los teléfonos IP que manejará el equipo de trabajo. La preparación se desarrolla en pocos pasos así:

1. Para la conexión a la red local no hace falta instalar otro cable debido a que el teléfono viene equipado con un switch de 2 puertos, es decir que del mismo dispositivo es posible conectar la red a la estación del usuario.
2. Los teléfonos SIP disponen de acceso mediante un navegador, con los que es posible configurarlos desde una PC, o también desde el propio teléfono con los botones de navegación se configura la dirección IP y los parámetros necesarios, ver Figura 3.17

Figura 3.17 Configuración de IP en teléfono VoIP

Fuente: El autor

3. Para disponer del servicio telefónico en el equipo se configura la cuenta SIP asociada al empleado que utilizará esa extensión, en la Figura 3.18 se observa el apartado de la cuenta SIP realizada directamente en el teléfono.

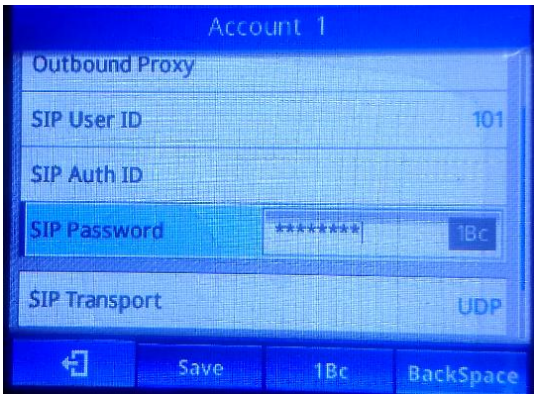


Figura 3.18 Configuración de cuenta SIP

Fuente: El autor

3.11. Servicio de correo electrónico

El servidor Elastix permite crear dominios y cuentas de correo para trabajar con la correspondencia electrónica lo que es posible visualizar directamente en la interface de Elastix o utilizar un gestor de correos como Thunderbird, Outlook, Zimbra, etc. A continuación se describe el proceso de configuración.

1. Se crea el dominio que se necesita, en este caso la empresa Vuelofertas maneja el dominio vuelofertas.com

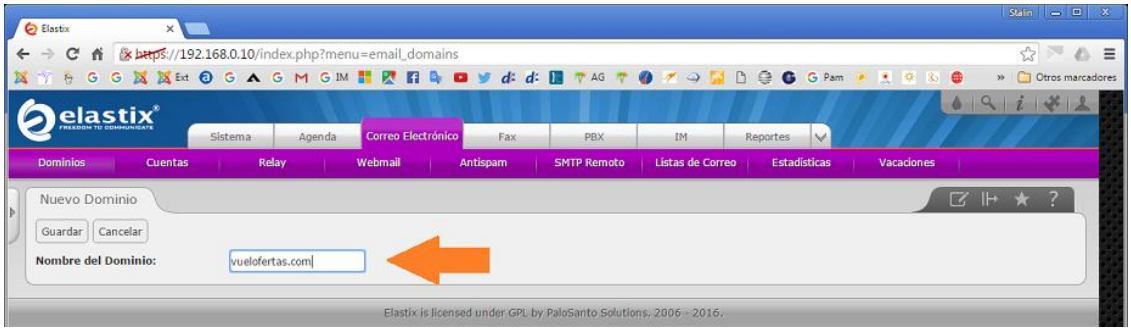


Figura 3.19 Configuración del dominio

Fuente: El autor

2. Se crea las cuentas asociadas a cada colaborador de la empresa con 100 MB de espacio para el buzón de correos, en la Figura 3.20 se muestra la cuenta info@vuelofertas.com que es utilizada como canal de atención general desde la página web www.vuelofertas.com.

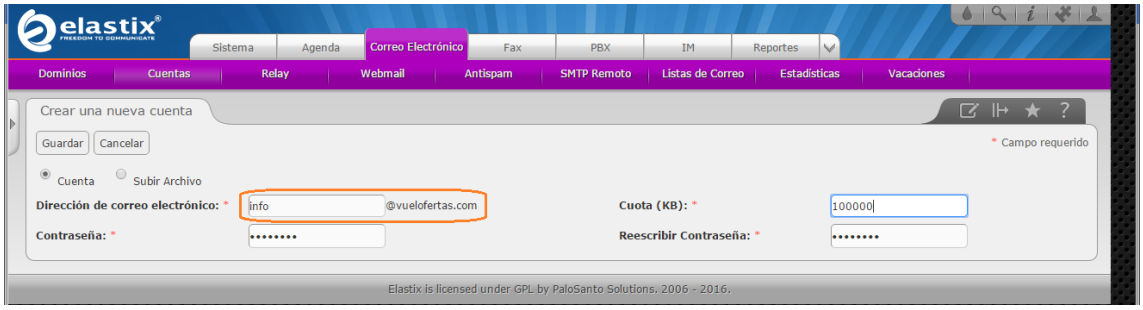


Figura 3.20 Creación cuenta de correo

Fuente: El autor

Una vez creado el dominio y sus cuentas, ya se tiene la capacidad de enviar y recibir correos electrónicos.

3.12. Mensajería instantánea

Para la activación de este servicio en primera instancia se instala la base de datos OPENFIRE dentro del servidor Elastix (Figura 3.21), luego de completar el requisito previo se activa el servicio desde la interface web de Elastix y se completa la instalación con los parámetros por defecto como se muestra en la Figura 3.22.

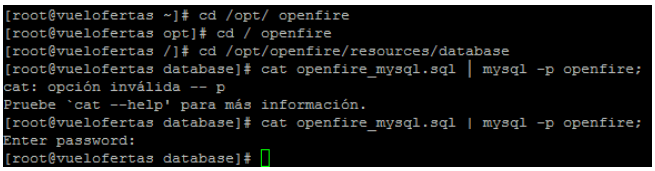


Figura 3.21 Instalación base OPENFIRE

Fuente: El autor

Figura 3.22 Configuración de OPENFIRE

Fuente: El autor

Con el servidor de mensajería listo se configura las diferentes cuentas de los trabajadores y se instala la aplicación cliente de mensajería SPARK de licencia gratuita y se inicia sesión con el usuario y clave asignados como se muestra en la figura 3.23.

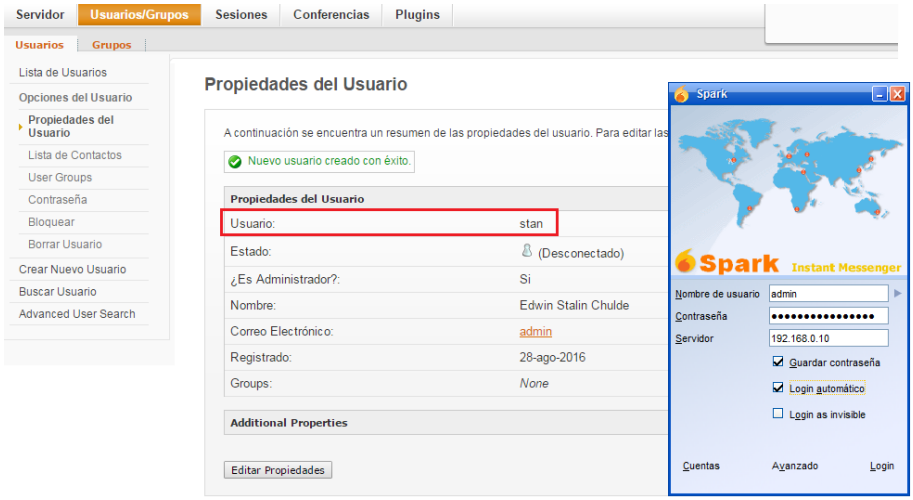


Figura 3.23 Cliente de mensajería instantánea

Fuente: El autor

3.13. Contestador automático IVR

La respuesta de voz interactiva conocida como IVR emite un mensaje de audio, el cual presenta un corto menú con el que un cliente, mediante la pulsación de un botón, puede canalizar su requerimiento a la extensión correcta, La pista grabada se la puede tratar con una aplicación de edición de audio (ver Imagen 3.24) para que tenga las características adecuadas de tonos, ruidos y además en formato compatible con Elastix. En este caso el mensaje que se presenta menciona “Gracias por llamar a su agencia de viajes Vuelofertas, si conoce el número de extensión digítelo ahora, caso contrario si desea comunicarse con una asesor de viajes marque 1, departamento contable marque 2 o marque 0 para ser atendido por uno de nuestros ejecutivos” . En la puesta en marcha de este servicio se tiene los siguientes pasos.

1. Se prepara el mensaje de audio que se emitirá al momento de marcar al PBX (ver figura 3.24), esto se lo realiza con un micrófono conectado directamente en un ordenador o desde una extensión conectada al servidor Elastix.

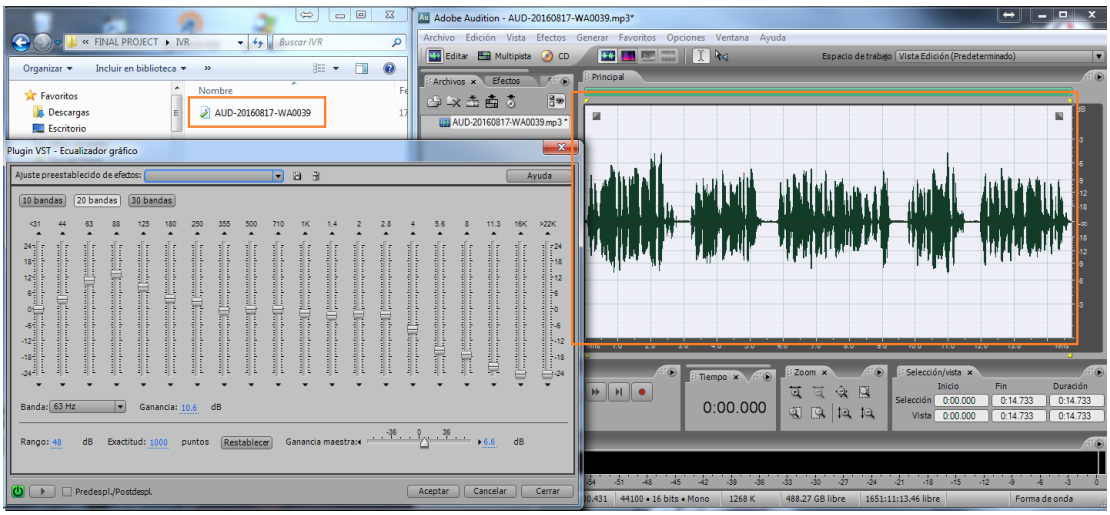


Figura 3.24 Edición pista de audio

Fuente: El autor

2. Se procede a subir la pista de audio al servidor en el apartado “Grabaciones del Sistema”, ver imagen 3.25.

Figura 3.25 Grabaciones del sistema

Fuente: El autor

3. Se crea el IVR con una descripción y se selecciona la pista de audio que se subió en el paso anterior, además se configura el destino de una llamada en caso de que se escoja una opción no válida, que ninguna extensión tome la llamada o si el cliente en la línea no presiona ningún botón, la figura 3.26 muestra los parámetros indicados.

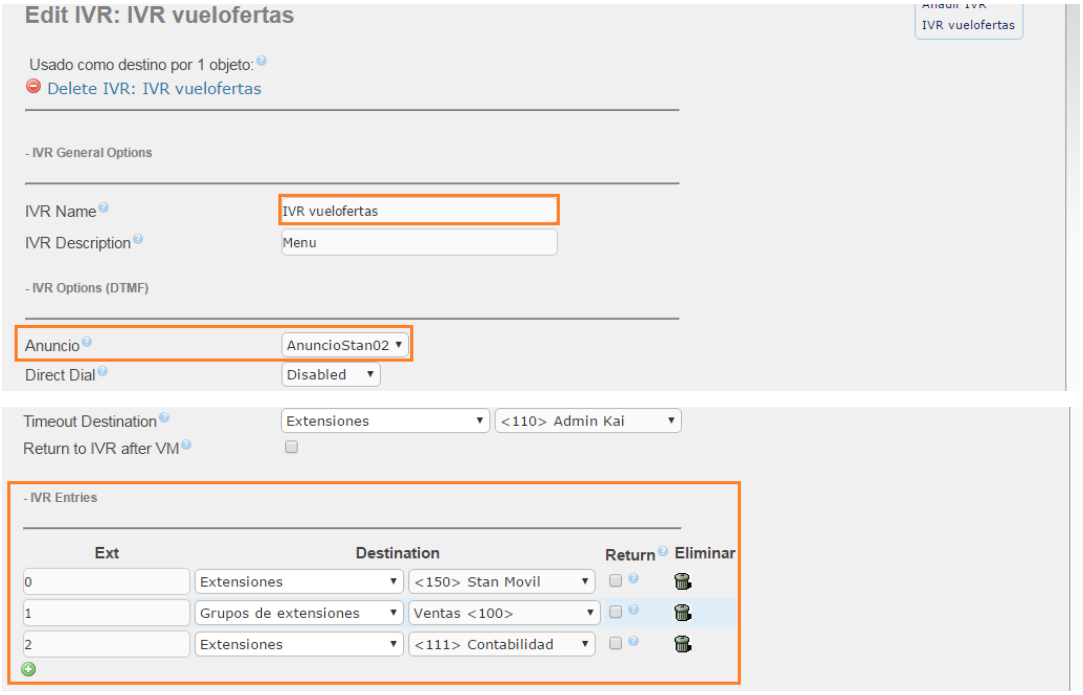


Figura 3.26 Configuraciones del IVR

Fuente: El autor

4. Es necesario crear una ruta que canalice todas las llamadas hacia el IVR que se configuró, esto se ve en la figura 3.27.

Figura 3.27 Ruta de llamadas entrantes hacia el IVR

Fuente: El autor

3.14. Configuración en las computadoras de la red local

En la estación de cada colaborador se instalará la aplicación de softphone llamada X-lite que le permitirá acceder al servicio telefónico. A continuación se describe los pasos para habilitar el servicio.

1. Se descarga el instalador de las páginas oficiales (http://www.counterpath.com/x-lite/), luego se procede con la instalación.
2. Una vez instalado se abre la aplicación y configura la cuenta SIP que corresponda a la extensión del empleado de la empresa, esto es, su número de extensión con su respectiva clave única.

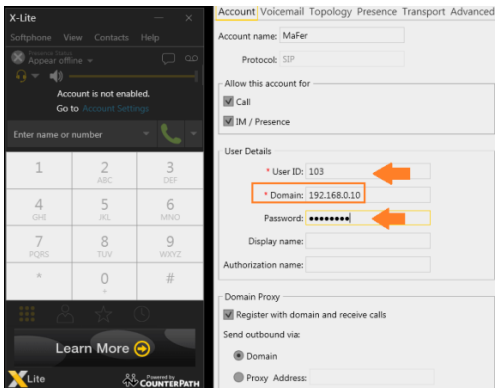


Figura 3.28 Configuración de cuenta SIP en aplicación cliente

Fuente: El autor

Con los parámetros completos se acepta las configuraciones y el servidor autenticará la sesión del usuario (ver imagen 3.29), mediante auriculares y el micrófono conectado al ordenador ya se tiene la capacidad de acceder a los servicios de telefonía proporcionados por Elastix.

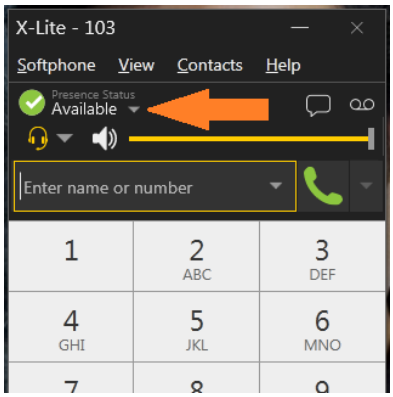


Figura 3.29 X-Lite con cuenta autenticada

Fuente: El autor

3.15. Diseño del sistema de registro y control de cerradura

Para establecer el diseño del sistema de control de acceso se acogen ciertos requisitos que la herramienta debe cumplir, se tiene en cuenta criterios de confiabilidad, escalabilidad e integración con otros sistemas.

3.16. Requisitos que el sistema de control de acceso debe cumplir

1. El sistema debe ser escalable, es decir, tener la capacidad de adicionar servicios relacionados de ser necesario.
2. El mantenimiento y/o reemplazo de partes no debe ser complicado.
3. Capacidad de conectarse a la red local y permitir el acceso desde un navegador.
4. Se debe registrar tanto la entrada como la salida de las instalaciones de la empresa.
5. En caso de falla del sistema debe existir una alternativa que permita la entrada o salida del personal en la oficina.
6. Si el suministro eléctrico es cortado el sistema por defecto deberá deshabilitarse y permitir el acceso y salida.

De acuerdo a los requisitos citados se escoge como base la plataforma Arduino cuya integración de forma modular lo hace efectivo para este desarrollo.

La sección principal del sistema es el microcontrolador Arduino, ya que es el dispositivo que autoriza o no el acceso de una entidad que ingresa sus credenciales mediante los lectores de huellas. Arduino se desarrolla de forma modular por lo cual en este proyecto se tiene un banco de relés y también un módulo Ethernet.

3.17. Diagrama de etapas de la propuesta de solución

La figura 3.30 muestra la estructura del sistema de control, el modulo principal es el microcontrolador y es a donde se conectan el Shield Ethernet que incluye la memoria flash, a través de cables se unen los dos lectores de huellas y el banco de relés a los cuales se conecta la cerradura para permitir su energización.

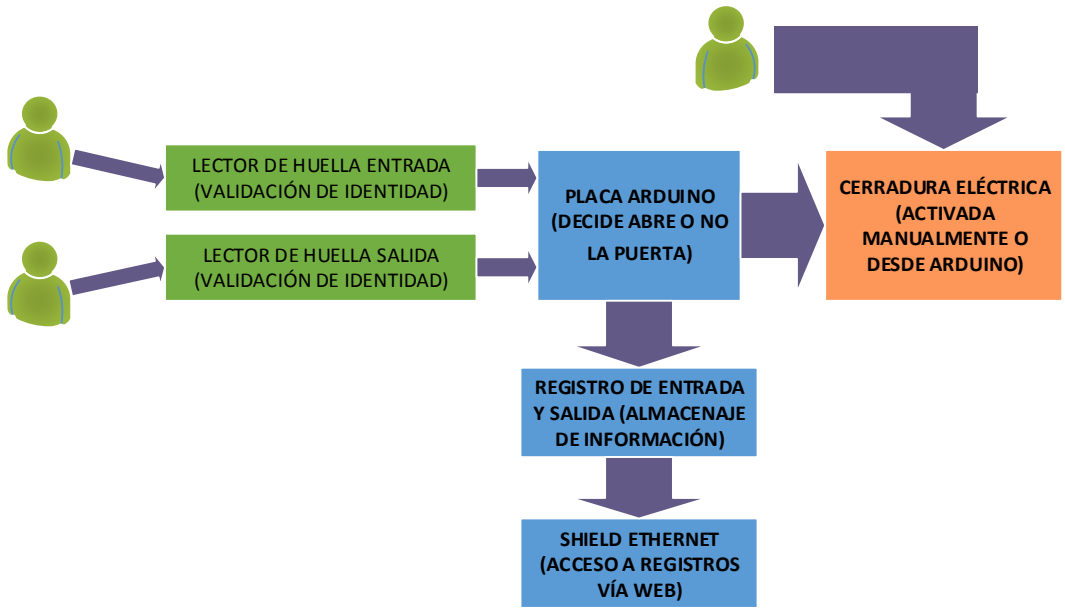


Figura 3.30 Esquema de sistema de control de acceso

Fuente: El autor

3.18. Registro y almacenamiento de información

La información de las huellas dactilares, es decir la base de datos con la cual el dispositivo biométrico valida la huella de una persona está almacenada en la memoria del mismo dispositivo, que en este caso permite guardar 162 identidades, es decir, que si un empleado registra su entrada, al escanear su huella el lector inicia la comunicación serial con el microcontrolador y envía la identidad de ese trabajador, la hora en la que el evento ocurre es tomada desde servidores en internet, esta información se la utiliza para realizar la tarea de desbloquear la puerta y enviarla al módulo que contiene la memoria(ver Figura 3.31).

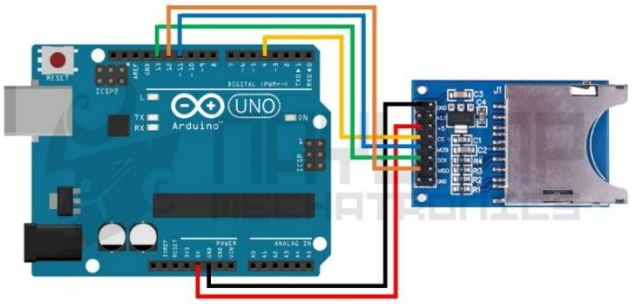


Figura 3.31 Módulo Ethernet Arduino

Fuente: El autor

La secuencia de programación dicta que en primera instancia se abra el archivo, luego escribe los datos al final de los registros previos y finalmente cierra el archivo. Para lectura de la información recopilada se hace uso del Shield Ethernet que permite configurar una dirección IP (192.168.0.177), con la que se ingresa mediante un navegador a la información que guarda la memoria SD. La placa Ethernet puede ser montada sobre la placa del microcontrolador ARDUINO, como muestra la Figura 3.32.

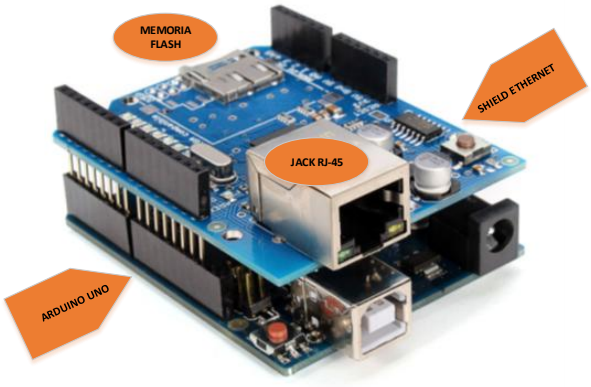


Figura 3.32 Módulo Ethernet Arduino

Fuente: el autor

3.19. Dispositivo lector de huellas

Una de las cualidades del hardware libre es que existe gran variedad de dispositivos compatibles que trabajan con la plataforma Arduino, en este caso se tiene el lector de huellas FPM10 que permite almacenar 162 identidades y la comunicación es en modo serial, con lo que es necesario realizar una programación dirigida hacia los pines específicos del microcontrolador, en la Figura 3.33 se muestra el dispositivo biométrico con los cables de conexión, rojo y negro se usa para la energía y verde con amarillo en la comunicación serial.

Figura 3.33 Lector de huellas

Fuente: El autor

3.20. Configuración y programación del microcontrolador ARDUINO

En esta etapa se toma como base el ejemplo que viene como parte de la librería “Adafruit_Fingerprint”, ver figura 3.34, necesaria en el reconocimiento y almacenaje de las huellas dactilares. En la figura 3.35 se tiene el diagrama de flujo el cual permite desarrollar el programa principal que ejecuta el microcontrolador para permitir el acceso del personal. El programa se encuentra en la sección de ANEXOS al final de este informe.

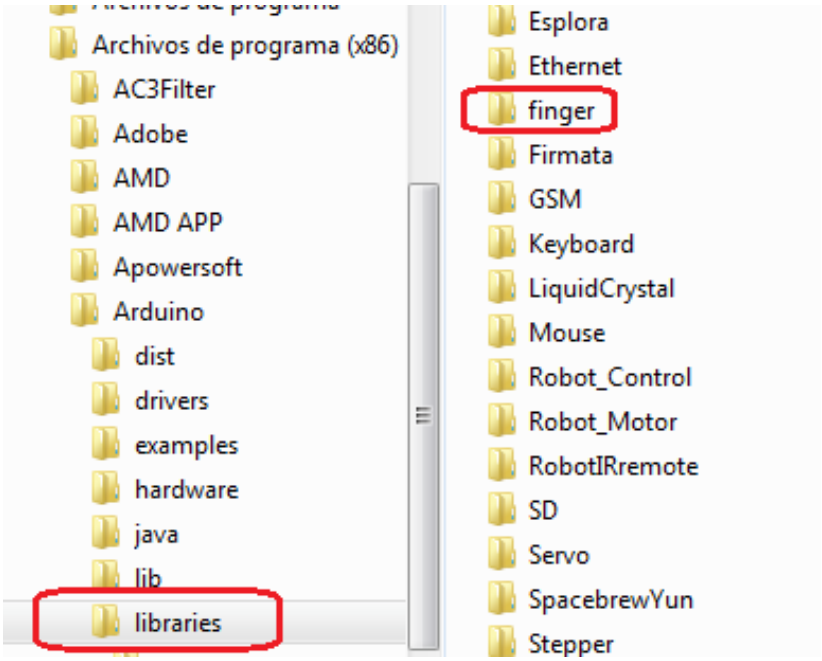


Figura 3.34 Librería para programación en Arduino

Fuente: El autor

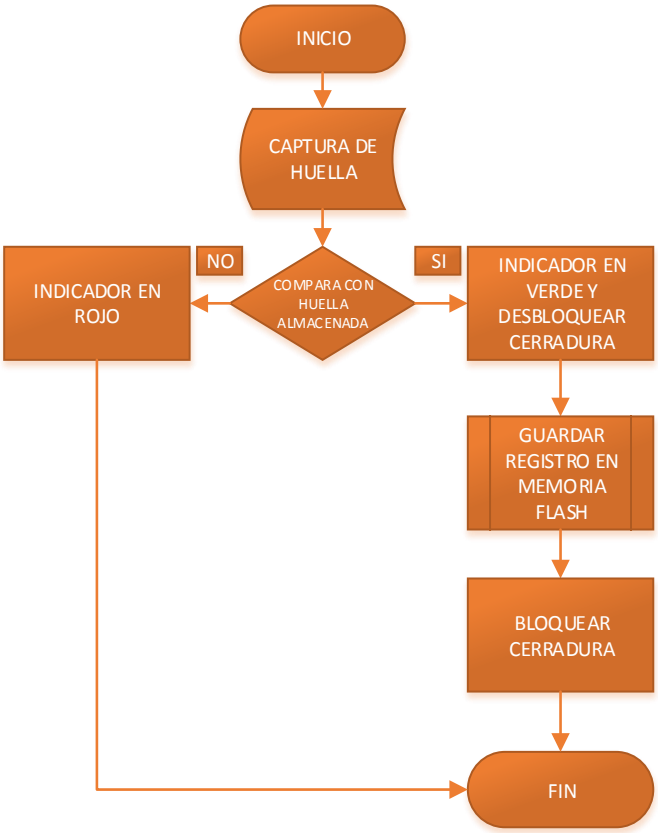


Figura 3.35 Diagrama de flujo

Fuente. El Autor

3.21. Interconexión de etapas

Las etapas que conforman el sistema de control de acceso son las siguientes.

Alimentación.- En esta instancia encuentra la fuente de energía que alimenta todo el sistema, esto es, las placas del microcontrolador, el Shield Ethernet con su memoria, la placa del banco de relés y también la cerradura electromagnética que necesita 12 voltios para su funcionamiento.

Microcontrolador.- Esta etapa contiene el cerebro del sistema cuya función principal es tomar la decisión de abrir o no la cerradura en respuesta a la información recibida desde los sensores, además de ello se debe encargar del registro y almacenamiento para su posterior visualización.

Captura de información.- Aquí se encuentran los dos dispositivos lectores de huellas que están conectado mediante cables de cobre y que son los encargados de censar las huellas y validarlas con las credenciales almacenadas en su memoria.

Banco de relés y cerradura.- Con el uso de electroimanes y optoacopladores, el banco de relés permite manejar dispositivos que funcionan con un voltaje y potencia mayores, es decir independizan el circuito de dispositivos digitales y el circuito de potencia en el

que se encuentra la cerradura. A continuación se tiene la Figura 3.36 que muestra las etapas involucradas.

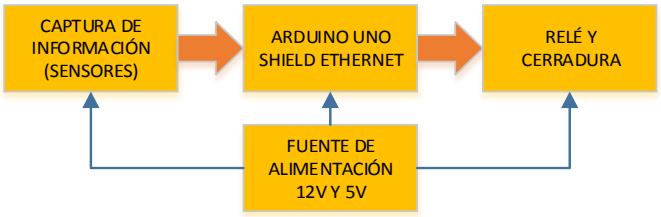


Figura 3.36 Etapas del sistema de control de acceso

Fuente: El autor

3.22. Implementación del Sistema de control de acceso

Se inicia la etapa de implementación con la conexión de los dispositivos que conforman el sistema en una mesa de pruebas, y se logra un entorno controlado y de fácil modificación en caso de requerirlo, a continuación se tiene el proceso de implementación.

1. Utilización de mesa de pruebas para conexiones preliminares, en la figura 3.37 se tiene la conexión básica del circuito.

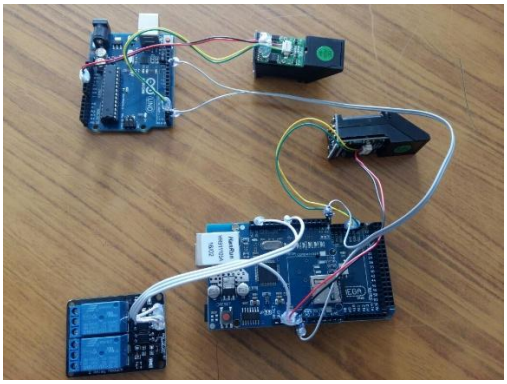


Figura 3.37 Mesa de pruebas

Fuente: El autor

2. Acondicionar el entorno y recopilación de herramientas.- Este paso consiste en obtener el software en sus versiones estables, descargar librerías, adecuar los cables y conectores, es decir, generar las condiciones propicias para la conformación del sistema de control de acceso.

3. Desarrollo de la programación. En base al diagrama de flujo (Figura 3.34) se desarrolla el programa y se prueba el funcionamiento a medida que avanza (Figura 3.38).

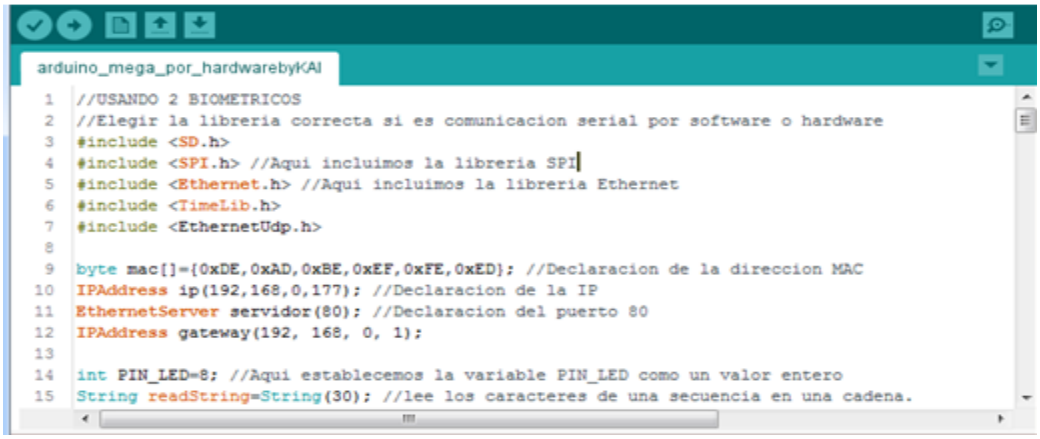


Figura 3.38 Proceso de programación

Fuente: El autor

4. Registro y almacenaje de huellas digitales.- Una vez concluida la etapa de programación se procede a capturar las huellas de cada empleado, es obligatorio que para cada persona se debe capturar cuatro huellas, es decir, de los dedos índice y pulgar de las dos manos, ver ejemplo Tabla 3.4. Además el procedimiento se lo realiza en los dos biométricos debido a que cada dispositivo tiene su propia memoria, la figura 3.39 muestra el proceso de lectura de huellas.

Tabla 3.4 Registro de huellas

Ubicación Lector 01	USUARIO	HUELLA	Ubicación Lector 02
0	KLEVER	Indice derecho	0
1	KLEVER	pulgar derecho	1
2	STALIN CHULDE	Indice derecho	2
3	STALIN CHULDE	pulgar derecho	3
4	STALIN CHULDE	indice izquierdo	4
5	STALIN CHULDE	pulgar izquierdo	5
6	FERNANDA TANDAZO	Indice derecho	6
7	FERNANDA TANDAZO	pulgar derecho	7
8	FERNANDA TANDAZO	indice izquierdo	8
9	FERNANDA TANDAZO	pulgar izquierdo	9

Fuente: El autor

Figura 3.39 Captura de huellas
Fuente: El autor

5. Cableado eléctrico y de datos.- Se realiza el tendido de cable tanto para alimentación de voltaje como de transmisión de datos (ver figura 3.40), en este caso se usa cable UTP CAT 5e, se debe tener muy en cuenta que la cerradura trabaja a 12 Voltios y el voltaje del microcontrolador es de 5 a 10 Voltios.

Figura 3.40 Cableado
Fuente: El autor

6. Adecuación de la fuente de voltaje dentro del servidor de comunicaciones.-Para dotar a los componentes de energía se optó por instalar otra fuente independiente dentro del servidor de comunicaciones, sin que ello altere el normal funcionamiento, la figura 3.41 muestra la salida de voltaje.

Figura 3.41 Fuentes de voltaje

Fuente: El autor

7. Instalación de la cerradura electromagnética.- Con los anclajes y tornillos de sujeción se procede a instalar la cerradura en la posición más adecuada para obtener el máximo contacto entre las placas de acople, ver figura 3.42.

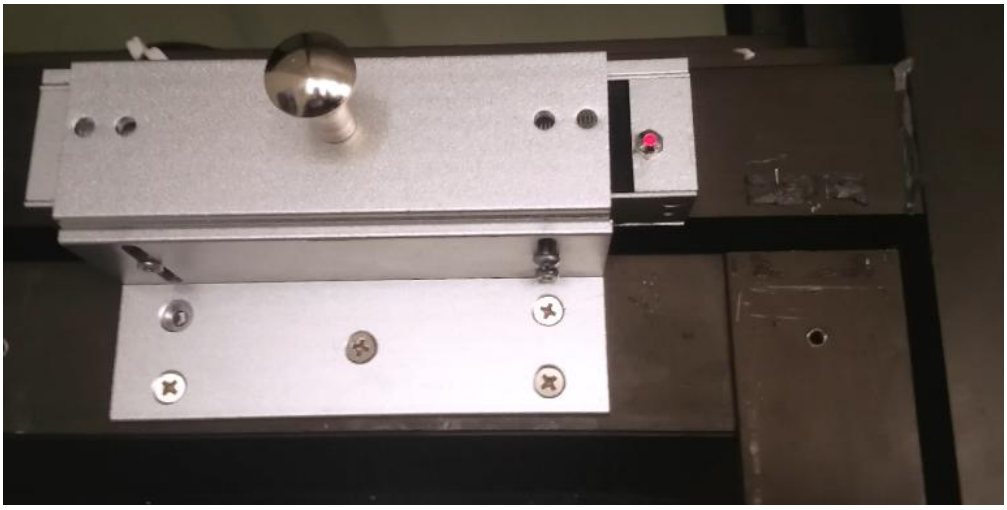


Figura 3.42 Instalación cerradura

Fuente: El autor

8. Dotar de protección al sistema dentro de una carcasa.- Para soportar el polvo y la manipulación es necesario proteger al microcontrolador y sus etapas con una carcasa rígida como muestra la figura 3.43.

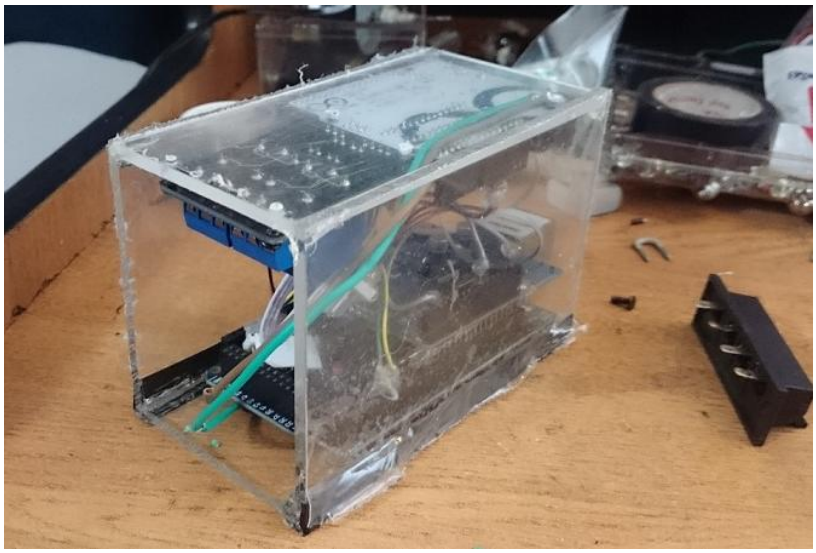


Figura 3.43 Carcasa del sistema de control
Fuente: El autor

9. Instalación de los lectores de huellas.- Finalmente se instalan los lectores de huellas en el interior y exterior de la oficina, para mayor facilidad los lectores se ubican a la altura de la manilla de la puerta como indica la figura 3.44.

Figura 3.44 Instalación de biométricos
Fuente: El autor

3.23. Etapa de pruebas

3.23.1. Prueba en sistema de comunicaciones unificadas.

Finalizada la instalación del sistema de comunicación se hace necesario una serie de pruebas que verifiquen el correcto funcionamiento de los servicios y equipos habilitados, a continuación la tabla 3.5 recopila todas las pruebas realizadas.

Tabla 3.5 Pruebas en sistema de comunicaciones

ITEM	PRUEBA	RESULTADO
1	Arranque de servidor	OK
2	levantamiento de servicios	OK
3	Reconocimiento troncales analógicas	OK
4	Autenticación cuentas SIP	OK
5	Funcionamiento IVR	OK
6	Funcionalidad PBX	OK
7	Envío de correo	OK
8	Recepción de correo	OK
9	Inicio sesión mensajería instantánea	OK
10	Recepción de mensaje en casillero de voz	OK
11	Envío de fax	OK
12	Recepción de fax	OK
13	Llamadas entrantes	OK
14	Llamadas salientes	OK
15	Llamadas internas	OK
16	Autenticación de Smartphone	OK
17	Llamada desde Smartphone	OK
18	Llamada hacia Smartphone	OK
19	Funcionalidad transferencia de llamada	OK
20	Uso de línea celular como extensión	OK

Fuente: El autor

3.23.2. Prueba en sistema de control de acceso.

Concluida la instalación del sistema de control y acceso se realizan las pruebas y verificación de los elementos y su interacción, en la tabla 3.6 se tabulan las pruebas aplicadas al sistema de control de acceso.

Tabla 3.6 Prueba en sistema de control de acceso

ITEM	PRUEBA	RESULTADO
1	Fuente de voltaje a 12 y 5 voltios	OK
2	Encendido de módulos Arduino	OK
3	Energización de cerradura	OK
4	Registro de nuevas huellas dactilares	OK
5	Lectura permanente de huellas	OK
6	Apertura de cerradura con huella válida (salida)	OK
7	Apertura de cerradura con huella válida (entrada)	OK
8	Registro de evento (entrada o salida) en memoria	OK
9	Acceso a dirección IP donde se almacena registro	OK
10	Apertura de cerradura mediante pulsador externo	OK
11	Desconexión de sistema (Botón secreto)	OK
12	Luz indicadora de cerradura activada	OK
13	Activación de cerradura	OK
14	Tiempo de espera (cerradura desactivada) 4 segs.	OK
15	Acceso a registro de asistencia vía navegador	OK

Fuente: El autor

4. RESULTADOS

Análisis del presupuesto económico del proyecto.

Completada la instalación de los dos sistemas se analiza el factor económico que atañe al proyecto y por ello se recopila la información en la tabla 4.1 que presenta el monto de la inversión que se utilizó en el desarrollo del proyecto.

Tabla 4.1 Presupuesto económico del proyecto

	ITEM	CANTIDAD	VALOR	TOTAL		
CENTRAL IP	MAINBOARD	1	16	16	SISTEMA COMUNICACIÓN	$ 1.143,00
	PROCESADOR	1	25	25		
	CASE	1	18	18		
	MODULO FXO	1	220	220		
	SWITCH 5 PUERTOS	1	20	20		
	TELÉFONO IP	8	85	680		
	COSTO Horas/Hombre	6	14	84		
	HEADSETS	10	8	80		
CONTROL DE ACCESO	ARDUINO UNO	2	15	30	SUBTOTAL SISTEMAS ACCESO	$ 402,00
	SHIELD ETHERNET	1	15	15		
	MODULO RELÉS	1	6	6		
	LECTOR DE HUELLAS	2	58	116		
	PLACA DE SOPORTE	1	6	6		
	BAQUELITA CON BORNES	1	10	10		
	CERRADURA ELECTROMAGNÉTICA	1	75	75		
	COSTO Horas/Hombre	12	12	144		
	MATERIALES VARIOS (Suelda, cables, cinta, amarras, tornillos etc.)	1	50	50		
			TOTAL	$ 1.595,00		

Fuente: El autor

Debido a la mayor relevancia del sistema de comunicaciones el análisis se basa en la comparación con sistemas de comunicación de características similares y ya que los componentes en la implementación de los servicios son los mismos para todo sistema de comunicación IP, la evaluación se enfoca en el costo del servidor y las centrales IP. En la tabla 4.2 se registra el valor del servidor con sistema Elastix que se implementó en el proyecto.

Tabla 4.2 Costo servidor Elastix

SERVIDOR ELASTIX	MAINBOARD	1	16	16
	PROCESADOR	1	25	25
	CASE	1	18	18
	COSTO Horas/Hombre	6	14	84
	MODULO FXO 4 puertos	1	220	220
			COSTO TOTAL:	$ 363,00

Fuente: El autor

En la tabla 4.3 se tiene tres sistemas comparables al servidor y servicios implementados.

Tabla 4.3 Centrales IP

MARCA	CARACTERÍSTICAS	COSTO
GRANDSTREAM	4 PUERTOS FXO	$ 480
	2 PUERTOS FXS	
	IVR	
	PUERTOS LAN Y WAN	
	CORREO DE VOZ	
DENWA	2 PUERTOS FXO	$ 850
	2 PUERTOS FXS	
	PUERTOS LAN Y WAN	
	CORREO DE VOZ	
	SLOTS Y ANTENAS GSM	
YEASTAR	4 PUERTOS FXO	$ 510
	SOPORTA 32 USUARIOS	
	PUERTOS LAN Y WAN	
	CORREO DE VOZ	

Fuente: El autor

Con los valores tabulados es claro ver que un servidor de comunicaciones basado en código abierto tiene un costo inferior incluso al valor de la solución más económica, en este caso la central Grandstream, además la implementación de un servidor a medida hace al sistema altamente escalable, es decir que posibilita: creación de nuevos servicios, aumentar el número de usuarios, cambiar el tipo de puertos para admitir líneas digitales, etc., lo cual en las centrales de marca no es posible y el cliente se debe regir por los parámetros establecidos de fábrica y si es necesaria una expansión no hay otro camino que adquirir un equipo de mayores capacidades.

Conclusiones

- La implementación de este tipo de sistemas de comunicaciones es una gran solución para empresas del tipo PYMES debido a su monto de inversión muy contenido.
- La instalación del sistema Elastix en sus versiones más recientes o beta conlleva problemas de inestabilidad y sobre todo incompatibilidad con el hardware, además de escaso soporte en los blogs y foros de ayuda.
- El uso de hardware modular simplifica mucho el proceso de elaboración de los circuitos involucrados en la solución de un proyecto de electrónica.
- El software libre evoluciona constantemente con lo que es posible encontrar nuevas aplicaciones que complementan la funcionalidad de un servidor sin realizar inversiones onerosas.
- Para instalaciones en empresas que manejen más de cincuenta usuarios es necesario implementar servidores dedicados cuyos recursos de hardware sean capaces de manejar el procesamiento de todos los servicios ofrecidos.
- En el manejo de correos es necesario contratar un dominio y el servicio de hosting para tener la capacidad de enviar y recibir correos hacia y desde fuera de la red local.

Recomendaciones

- A pesar de tener un sistema de comunicaciones estable y potente se recomienda tener los teléfonos analógicos que se conectan directamente a las líneas telefónicas, ya que nunca se está exento un evento fortuito como un corte de energía y prácticamente perder la capacidad de comunicación.
- Se recomienda instalar una versión estable de Elastix por la gran cantidad de ayuda que existe en internet en el caso de surgir problemas con su funcionamiento.
- Dada una remota posibilidad de que el sistema de control de acceso no funcione adecuadamente es recomendable mantener un interruptor de emergencia para el desbloqueo de la cerradura.
- Si la empresa donde se instala el sistema de comunicaciones unificadas no va utilizar uno o varios de los servicios que brinda es recomendable que no se los configure y active porque esto implica mayor consumo de recursos para el servidor, lo cual es innecesario.
- En el sistema de control de acceso se recomienda tomar más de una huella ya que si por un incidente la persona no puede utilizar el dedo de la huella registrada prácticamente queda inhabilitada su entrada.
- En la conexión de los módulos del sistema de control de acceso es recomendable utilizar conectores para asegurar una óptima comunicación de los componentes y mejorar el montaje y desmontaje de los mismos.

Bibliografía

Andrew S., T. (2003). *Redes de Computadoras.* México: Pearson Prentice Hall.

Areny, R. P. (2003). *Sensores y Acondicionadores de Señal.* Barcelona: Marcombo.

Bonilla, W. G. (2005). *Electrónica Estudio*. Obtenido de http://www.electronicaestudio.com/

Carballar, J. A. (2008). *VoIP La telefonía de internet.* España: Thommson.

Cientificas, C. S. (1987). *Introducción a los sensores.* Madrid: El Museo Universal.

Digium Inc. (2016). Retrieved from http://www.asterisk.org/

Dordoigne, J., & Atelin, P. (2006). *Redes Informáticas conceptos fundamentales.* Barcelona: ENI.

Fonality Inc. (2016). Retrieved from www.fonality.com

Jordi Griera, Barceló, J., Cerdà, L., & Peig, E. (2008). *Estructura de redes de computadores.* Barcelona: UOC.

PaloSanto Solutions. (2016). Obtenido de http://www.elastix.com/

Philippe Atelin, J. D. (2007). *TCP/IP y Protocolos de Internet .* ENI.

www.informatica-hoy.com.ar. (Junio de 2015). *www.informatica-hoy.com.ar*. Obtenido de http://www.informatica-hoy.com.ar/redes/LAN-WAN-MAN-WLAN-WMAN-WWMAN-SAN-PAN.php

ANEXO A

Tabla 4.4 Comandos básicos para programación ARDUINO

SÍMBOLOS	DESCRIPCIÓN
Declaración de variables	Una variable es un contenedor para guardar algún dato.
setup()	void setup() es la parte que inicializa las configuraciones de los diferentes elementos del programa de Arduino
loop()	Ésta parte se ejecuta una y otra vez. En un programa de Arduino todo el código se ejecuta línea a línea. Después de ejecutar la void setup() en el arranque continua con la void loop().
Marcas de puntuación. Paréntesis y llaves	Las marcas de puntuación se utilizan para definir el inicio y el final de ciertas partes del código
int myNumber = 14;	"int" es el tipo de variable, "miNumero" es el nombre y 14 es el valor
byte mySmallNumber = 150;	Para ahorrar espacio en la memoria de Arduino, es útil almacenar las variables como bytes. Un byte es un número entero de 8 bits con un rango de 0 a 255
float mydecimalNumber = 2.33;	El único tipo de datos que puede guardar números con decimales es "float".
int myArray[] = {1, 2, 3, 4, 5, 6};	(Matrices): A veces puede ser útil guardar una colección de valores, entonces tendremos que utilizar una matriz.
myVariable = map(mySensor,50,200,0,500);	Ésta función remapea un rango de valores a otro rango de valores. Digamos, hace una regla de 3:
myVariable = random(5);	Ésto guardará un número aleatorio en miVariable
==	Es usado para comparar si un elemento es igual a otro
!=	Se usa para comprobar si un elemento es distinto de otro.
<	Se usa para comparar si un elemento es menor que otro.
>	Se usa para comparar si un elemento es mayor que otro.
<=	Se usa para comparar si un elemento es menor o igual que otro
>=	Se usa para comparar si un elemento es mayor o igual que otro.
Y (&&)	Se usa para determinar si dos o más elementos son verdaderos. x < y && y > 5 /* x es menor que y, e y es mayor que 5 */
O (||)	Se usa para determinar si alguno de los dos elementos es verdadero. x < y || y > 5
Negacion (!)	Se usa para determinar si algo no es verdad. x!=5 /* x no es igual a 5 */
boolean myBoolean = true;	True y False son lo que llamamos constantes Booleanas y definen si algo lo es, o no, a nivel lógico:
digitalWrite(ledPin,HIGH);	HIGH y LOW se usan para determinar el estado de un pin digital, que sólo tiene esos dos estados.
pinMode(12,OUTPUT);	INPUT y OUTPUT se usan cuando declaramos el modo de funcionamiento de nuestro pin digital, sólo existen esos dos modos:

if (myVariable>myOtherVariable){
doSomething;
}
	Una sentencia If es como un test que Arduino puede hacer para determinar si algo es verdadero o falso.

for (int i=0; i<200; i++){
doSomething;
}
	Los bucles For se usan cuando quieres repetir una parte del código un número concreto de veces.

while (myVariable<100){	

doSomething; }	Un bucle While durará hasta que la condición entre paréntesis sea falsa.
pinMode(pin,OUTPUT);	Un pin digital tiene sólo dos modos, OUTPUT (salida) e INPUT (entrada).
digitalWrite(pin, HIGH);	Para encender o apagar tu pin digital debes usar el comando digitalWrite(). Entre paréntesis debes indicar qué pin modificar, y qué valor darle:
digitalRead(pin);	El comando digitalRead() lee el estado de un pin y devuelve HIGH si está a 5V o LOW si hay 0V en él:
analogRead(pin);	Para leer el valor de un pin analógico tienes que usar el comando analogRead() y poner la referencia del pin que deseas leer:
analogWrite(pin,value);	Con esta función es posible enviar un valor pseudo-analógico a estos pines digitales especiales.
delay(1000);	Este delay pondrá una pausa en su programa de un segundo,
myVariable = millis();	Este comando millis() devolverá cuantos milisegundos han pasado desde que Arduino inició la ejecución del programa. Para poder utilizar este valor hay que guardarlo en una variable:
Serial.begin(9600);	Este código abrirá el puerto serie y pone la velocidad de comunicación a 9600 baudios.
Serial.println(12345);	Este comando va a imprimir lo que pones dentro de los paréntesis y añadir un final de línea. Para imprimir números enteros tiene que poner el tipo dentro de los paréntesis:
Serial.println('C');	La línea de código anterior enviará el carácter C a través del puerto serie.

Fuente: El autor

ANEXO B

Programación Arduino Mega

```
//USANDO 2 BIOMETRICOS
//Elegir la libreria correcta si es comunicacion serial por software o hardware
#include <SD.h>
#include <SPI.h> //Aqui incluimos la libreria SPI
#include <Ethernet.h> //Aqui incluimos la libreria Ethernet
#include <TimeLib.h>
#include <EthernetUdp.h>

byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; //Declaracion de la direccion MAC
IPAddress ip(192,168,0,177); //Declaracion de la IP
EthernetServer servidor(80); //Declaracion del puerto 80
IPAddress gateway(192, 168, 0, 1);
int PIN_LED=8; //Aqui establecemos la variable PIN_LED como un valor entero
String readString=String(30); //lee los caracteres de una secuencia en una cadena.
//Los strings se representan como arrays de caracteres (tipo char)
String state=String(3);

bool control = true ;

File dataFile;
#include <Adafruit_Fingerprint.h>
int getFingerprintIDez();
//Este programa funciona el biométrico, la sd y shield ethernet, solo por la sd tocó habilitar la comunicación serial
por hardware.
//Ubicar el shield sobre el Arduino Mega poniendo uno a uno
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);

int x=0;
int tiempo=1000;

//**************************************************
// NTP Servers:
IPAddress timeServer(132, 163, 4, 101); // time-a.timefreq.bldrdoc.gov
// IPAddress timeServer(132, 163, 4, 102); // time-b.timefreq.bldrdoc.gov
// IPAddress timeServer(132, 163, 4, 103); // time-c.timefreq.bldrdoc.gov
//const int timeZone = 1;     // Central European Time
const int timeZone = -5;  // Eastern Standard Time (USA)
//const int timeZone = -4;  // Eastern Daylight Time (USA)
//const int timeZone = -8;  // Pacific Standard Time (USA)
//const int timeZone = -7;  // Pacific Daylight Time (USA)
EthernetUDP Udp;
unsigned int localPort = 8888;  // local port to listen for UDP packets
//**************************************************

int inchar;

void setup() {
 //inicializacion biometrico
 while (!Serial);  // For Yun/Leo/Micro/Zero/...

 Serial.begin(9600);
 Serial.println("Adafruit finger detect test");

 // set the data rate for the sensor serial port
 finger.begin(57600);

 if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
 } else {
```

```
   Serial.println("Did not find fingerprint sensor :(");
   while (1);
  }
  Serial.println("Waiting for valid finger...");


  //inicializacion shield ethernet
  Ethernet.begin(mac, ip); //Inicializamos con las direcciones asignadas
  servidor.begin(); //inicia el servidor
  pinMode(PIN_LED,OUTPUT);
  digitalWrite(PIN_LED,HIGH);
  state="OFF";

//inicializacion SD
  while (!Serial)  ;      // wait for serial port to connect. Needed for Leonardo only
   Serial.print("Probando SD card...");

   pinMode(10, OUTPUT);//Incluso para arduino Mega
   if (!SD.begin(4))
    {  Serial.println("No hay tarjeta");
       return;    // NO sigas
    }
  Serial.println("Sd encontrada.");


  //inicializa reloj IP
  Serial.print("IP number assigned by DHCP is ");
  Serial.println(Ethernet.localIP());
  Udp.begin(localPort);
  Serial.println("waiting for sync");
  setSyncProvider(getNtpTime);

}

time_t prevDisplay = 0; // when the digital clock was displayed


void loop() {
getFingerprintIDez();
  //delay(50);           //don't ned to run this at full speed.
pagina();
rtc();
recepcion();
}

void pagina(){
    //EthernetClient Crea un cliente que se puede conectar a
  //una dirección específica de Internet IP
  EthernetClient cliente= servidor.available();
  if(cliente) {
    boolean lineaenblanco=true;
    while(cliente.connected()) {
      if(cliente.available()) {
        char c=cliente.read();
        if(readString.length()<30) {
          readString.concat(c);
          //Cliente conectado
          //Leemos petición HTTP caracter a caracter
          //Almacenar los caracteres en la variable readString
        }
        if(c=='\n' && lineaenblanco) //Si la petición HTTP ha finalizado
          {
          int LED = readString.indexOf("LED=");
```

```
        if(readString.substring(LED,LED+5)=="LED=T") {
           digitalWrite(PIN_LED,LOW);
           state="ON";
           delay(tiempo);
           digitalWrite(PIN_LED,HIGH);
           state="OFF";


        }
        // if (readString.substring(LED,LED+5)=="LED=F") {
        //    digitalWrite(PIN_LED,HIGH);
        //    state="OFF";
        // }

        if (readString.substring(LED,LED+5)=="LED=A") {
           x=1;
           state="ACT";
        }

//Cabecera HTTP estándar
cliente.println("HTTP/1.1 200 OK");
cliente.println("Content-Type: text/html");
cliente.println(); //Página Web en HTML
cliente.println("<html>");
cliente.println("<head>");
cliente.println("<title>LED ON/OFF</title>");
cliente.println("</head>");
cliente.println("<body width=100% height=100%>");
cliente.println("<center>");
cliente.println("<h1>CONTROL DE ACCESOS</h1>");
cliente.print("<br><br>");
//cliente.print("Estado del LED: ");
//cliente.print(state);
//cliente.print("<br><br><br><br>");
cliente.println("<input type=submit value=ON style=width:200px;height:75px onClick=location.href='./?LED=T\'>");
//cliente.println("<input type=submit value=OFF style=width:200px;height:75px
onClick=location.href='./?LED=F\'>");
cliente.println("<input type=submit value=ACT style=width:200px;height:75px onClick=location.href='./?LED=A\'>");
cliente.println("</center>");
cliente.println("</body>");
cliente.println("</html>");


if(x==1){
 dataFile  = SD.open("data.htm");       // open web page file
  if ( dataFile) {
 while( dataFile.available()) {
 cliente.write(dataFile.read()); // send web page to client
  }

 }
 x=0;
 dataFile.close();
}

cliente.stop();
//Cierro conexión con el cliente
readString="";
}
}
}
}
}
```

```
uint8_t getFingerprintID() {
 uint8_t p = finger.getImage();
 switch (p) {
  case FINGERPRINT_OK:
   Serial.println("Image taken");
   break;
  case FINGERPRINT_NOFINGER:
   Serial.println("No finger detected");
   return p;
  case FINGERPRINT_PACKETRECIEVEERR:
   Serial.println("Communication error");
   return p;
  case FINGERPRINT_IMAGEFAIL:
   Serial.println("Imaging error");
   return p;
  default:
   Serial.println("Unknown error");
   return p;
 }

 // OK success!

 p = finger.image2Tz();
 switch (p) {
  case FINGERPRINT_OK:
   Serial.println("Image converted");
   break;
  case FINGERPRINT_IMAGEMESS:
   Serial.println("Image too messy");
   return p;
  case FINGERPRINT_PACKETRECIEVEERR:
   Serial.println("Communication error");
   return p;
  case FINGERPRINT_FEATUREFAIL:
   Serial.println("Could not find fingerprint features");
   return p;
  case FINGERPRINT_INVALIDIMAGE:
   Serial.println("Could not find fingerprint features");
   return p;
  default:
   Serial.println("Unknown error");
   return p;
 }

 // OK converted!
 p = finger.fingerFastSearch();
 if (p == FINGERPRINT_OK) {
  Serial.println("Found a print match!");
 } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  Serial.println("Communication error");
  return p;
 } else if (p == FINGERPRINT_NOTFOUND) {
  Serial.println("Did not find a match");
  return p;
 } else {
  Serial.println("Unknown error");
  return p;
 }

 // found a match!
 Serial.print("Found ID #"); Serial.print(finger.fingerID);
 Serial.print(" with confidence of "); Serial.println(finger.confidence);
 grabarsd();
```

```
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  grabarsd();
  return finger.fingerID;
}

void grabarsd(){

control = true ;
  if (control)
  {
     File dataFile = SD.open("data.htm", FILE_WRITE);
     if (dataFile)   // Si ha podido abrir el fichero
      {

      dataFile.print("USUARIO ID #: "); dataFile.print(finger.fingerID);
      dataFile.print("  -  ");
      dataFile.print(year());dataFile.print("/");
      dataFile.print(month());dataFile.print("/");
      dataFile.print(day());dataFile.print("  -  ");
      dataFile.print(hour());dataFile.print(":");
      dataFile.print(minute());dataFile.print(":");
      dataFile.print(second());dataFile.print(" ");
      dataFile.print("  Entrada ");
      dataFile.println("<br>");


      delay (100);
      abrir();
            }
           dataFile.close();
           Serial.println("Listo. Terminado la grabacion en uSD.");
           control = false ;
      }
     else     // SI no puede abrir el fichero
        Serial.println("Error, no puedo usar data.htm");
        control = false ;
   }


  void rtc(){
    if (timeStatus() != timeNotSet) {
    if (now() != prevDisplay) { //update the display only if time has changed
      prevDisplay = now();
      digitalClockDisplay();
    }
  }

  }
```

```
   void digitalClockDisplay(){

  //Serial.print(year());
  //Serial.print("/");
  //Serial.print(month());
  //Serial.print("/");
  //Serial.print(day());
  //Serial.print(" ");
  //Serial.print(" ");
  //Serial.print(hour());
  //Serial.print(":");
  //Serial.print(minute());
  //Serial.print(":");
  //Serial.print(second());
  //Serial.println(" ");

   }

void printDigits(int digits){
  // utility for digital clock display: prints preceding colon and leading 0
  Serial.print(":");
  if(digits < 10)
    Serial.print('0');
  Serial.print(digits);
}

/*-------- NTP code ----------*/

const int NTP_PACKET_SIZE = 48; // NTP time is in the first 48 bytes of message
byte packetBuffer[NTP_PACKET_SIZE]; //buffer to hold incoming & outgoing packets

time_t getNtpTime()
{
  while (Udp.parsePacket() > 0) ; // discard any previously received packets
  Serial.println("Transmit NTP Request");
  sendNTPpacket(timeServer);
  uint32_t beginWait = millis();
  while (millis() - beginWait < 1500) {
    int size = Udp.parsePacket();
    if (size >= NTP_PACKET_SIZE) {
      Serial.println("Receive NTP Response");
      Udp.read(packetBuffer, NTP_PACKET_SIZE);  // read packet into the buffer
      unsigned long secsSince1900;
      // convert four bytes starting at location 40 to a long integer
      secsSince1900 =  (unsigned long)packetBuffer[40] << 24;
      secsSince1900 |= (unsigned long)packetBuffer[41] << 16;
      secsSince1900 |= (unsigned long)packetBuffer[42] << 8;
      secsSince1900 |= (unsigned long)packetBuffer[43];
      return secsSince1900 - 2208988800UL + timeZone * SECS_PER_HOUR;
    }
  }
  Serial.println("No NTP Response :-(");
  return 0; // return 0 if unable to get the time
}

// send an NTP request to the time server at the given address
void sendNTPpacket(IPAddress &address)
{
  // set all bytes in the buffer to 0
  memset(packetBuffer, 0, NTP_PACKET_SIZE);
  // Initialize values needed to form NTP request
  // (see URL above for details on the packets)
  packetBuffer[0] = 0b11100011;   // LI, Version, Mode
```

```
  packetBuffer[1] = 0;     // Stratum, or type of clock
  packetBuffer[2] = 6;     // Polling Interval
  packetBuffer[3] = 0xEC;  // Peer Clock Precision
  // 8 bytes of zero for Root Delay & Root Dispersion
  packetBuffer[12]  = 49;
  packetBuffer[13]  = 0x4E;
  packetBuffer[14]  = 49;
  packetBuffer[15]  = 52;
  // all NTP fields have been given values, now
  // you can send a packet requesting a timestamp:
  Udp.beginPacket(address, 123); //NTP requests are to port 123
  Udp.write(packetBuffer, NTP_PACKET_SIZE);
  Udp.endPacket();
}

 void abrir(){
digitalWrite(PIN_LED,LOW);
delay(tiempo);
digitalWrite(PIN_LED,HIGH);
}


void recepcion(){
 if(Serial.available() >0)
  {
    inchar=Serial.read();
    if (inchar=='0')
    {
      inchar=0;
      grabarsd2();
        }

    if (inchar=='1')
    {
      inchar=1;
      grabarsd2();
        }

     if (inchar=='2')
    {
      inchar=2;
      grabarsd2();
        }

      if (inchar=='3')
    {
      inchar=3;
      grabarsd2();
        }

      if (inchar=='4')
    {
      inchar=4;
      grabarsd2();
        }

      if (inchar=='5')
    {
      inchar=5;
      grabarsd2();
        }

      if (inchar=='6')
```

```
   {
    inchar=6;
    grabarsd2();
      }

    if (inchar=='7')
   {
    inchar=7;
    grabarsd2();
      }

    if (inchar=='8')
   {
    inchar=8;
    grabarsd2();
      }

    if (inchar=='9')
   {
    inchar=9;
    grabarsd2();
      }

    if (inchar=='10')
   {
    inchar=10;
    grabarsd2();
      }
}
}

void grabarsd2(){


control = true ;
 if (control)
 {
    File dataFile = SD.open("data.htm", FILE_WRITE);
    if (dataFile)   // Si ha podido abrir el fichero
     {


     dataFile.print("USUARIO ID #: "); dataFile.print(finger.fingerID);
     dataFile.print("  -  ");
     dataFile.print(year());dataFile.print("/");
     dataFile.print(month());dataFile.print("/");
     dataFile.print(day());dataFile.print("  -  ");
     dataFile.print(hour());dataFile.print(":");
     dataFile.print(minute());dataFile.print(":");
     dataFile.print(second());dataFile.print(" ");
     dataFile.print("  Salida  ");
     dataFile.println("<br>");

     delay (100);
     abrir();
          }
          dataFile.close();
          Serial.println("Listo. Terminado la grabacion en uSD.");
          control = false ;
     }
    else     // SI no puede abrir el fichero
       Serial.println("Error, no puedo usar data.htm");
       control = false ;
```

```
 }

 /*
////leer y desplegar serialmente por Monitor Serial
 void leersd(){

  File dataFile = SD.open("data.htm");
// Si lo hemos podido abrir correctamente:
if (dataFile) {
  // Mostramos un aviso de comienzo del txt
  Serial.println("* A continuacion se muestra el contenido de data.txt:");
  // Mandamos sus datos por el puerto serie.
  while (dataFile.available()) {
    Serial.write(dataFile.read());


     }
  // Cerramos el archivo.
  dataFile.close();
// Si no hemos conseguido abrir el archivo mandamos un error.
}else {
  Serial.println("Error al abrir data.txt");
}
 }
 */
```

Programación Arduino UNO (Modo espera)

```
 #include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

int getFingerprintIDez();

// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino  (WHITE wire)
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

// On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white
//Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);

void setup()
{
  while (!Serial);  // For Yun/Leo/Micro/Zero/...

  Serial.begin(9600);
  //Serial.println("Adafruit finger detect test");

  // set the data rate for the sensor serial port
  finger.begin(57600);


  if (finger.verifyPassword()) {
   // Serial.println("Found fingerprint sensor!");
  } else {
   // Serial.println("Did not find fingerprint sensor :(");
    while (1);
  }
 // Serial.println("Waiting for valid finger...");
}

void loop()                     // run over and over again
{
  getFingerprintIDez();
  delay(50);            //don't ned to run this at full speed.
}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
    //  Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
    //  Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
    //  Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
    //  Serial.println("Imaging error");
      return p;
    default:
     // Serial.println("Unknown error");
      return p;
  }

  // OK success!
```

```
  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:
     // Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
     // Serial.println("Image too messy");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
    //  Serial.println("Communication error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
     // Serial.println("Could not find fingerprint features");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
     // Serial.println("Could not find fingerprint features");
      return p;
    default:
     // Serial.println("Unknown error");
      return p;
  }

  // OK converted!
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
  //  Serial.println("Found a print match!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
   // Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
   // Serial.println("Did not find a match");
    return p;
  } else {
   // Serial.println("Unknown error");
    return p;
  }

  // found a match!
 // Serial.print("Found ID #"); Serial.print(finger.fingerID);
  //Serial.print(" with confidence of "); Serial.println(finger.confidence);
  Serial.print(finger.fingerID);
  delay(1000);
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  //Serial.print("Found ID #"); Serial.print(finger.fingerID);
  //Serial.print(" with confidence of "); Serial.println(finger.confidence);
  Serial.print(finger.fingerID);
  delay(1000);
  return finger.fingerID;
}
```

ANEXO C

DATA SHEET MICROCONTROLADOR ARDUINO

Arduino UNO

Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Index

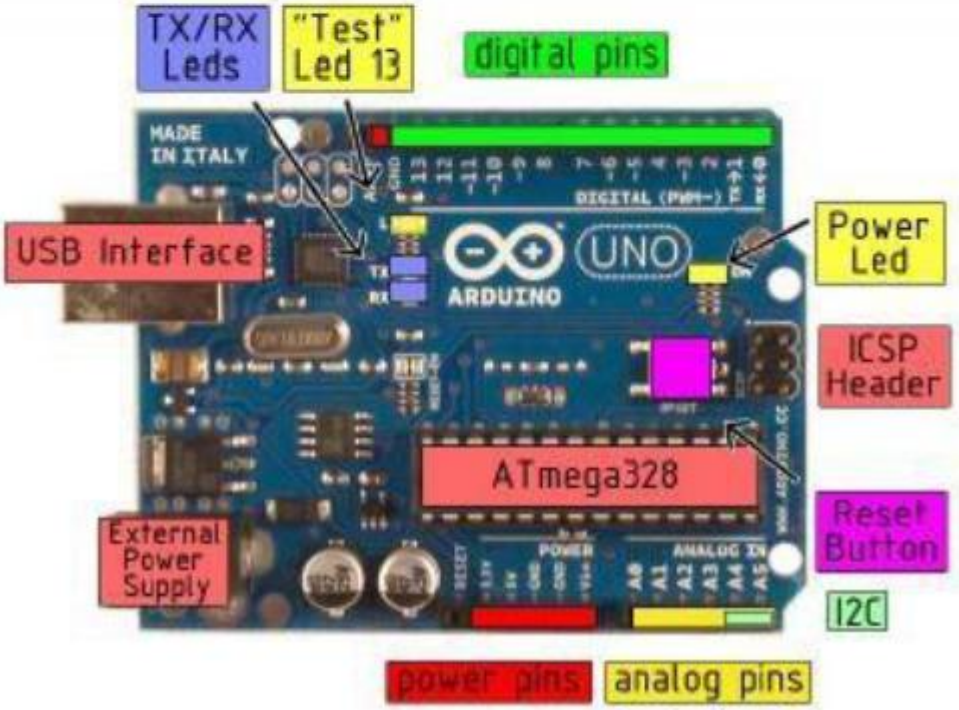
Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

power pins analog pins

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and Atmega328 ports.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the **Tools** > **Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

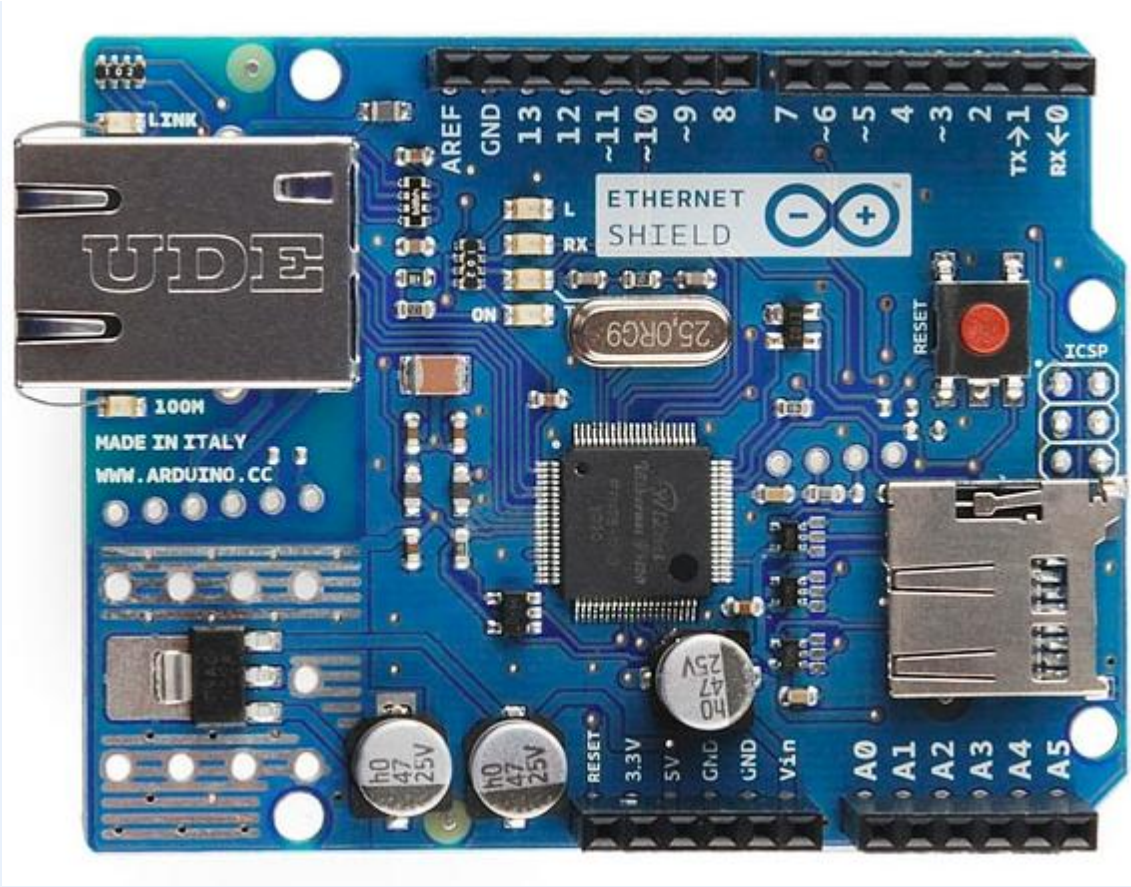
Once you have your skecth you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.

ANEXO D

DATA SHEET SHIELD ETHERNET



Overview

The Arduino Ethernet Shield connects your Arduino to the internet in mere minutes. Just plug this module onto your Arduino board, connect it to your network with an RJ45 cable (not included) and follow a few simple instructions to start controlling your world through the internet. As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Hundreds of thousands of Arduino boards are already fueling people's creativity all over the world, everyday. Join us now, Arduino is you!

- Requires an Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Ethernet Controller: W5100 with internal 16K buffer
- Connection speed: 10/100Mb
- Connection with Arduino on SPI port

Description

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiznet W5100ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.
The most recent revision of the board exposes the 1.0 pinout on rev 3 of the Arduino UNO board.
The Ethernet Shield has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.
There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Uno and Mega (using the Ethernet library). The onboard microSD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.

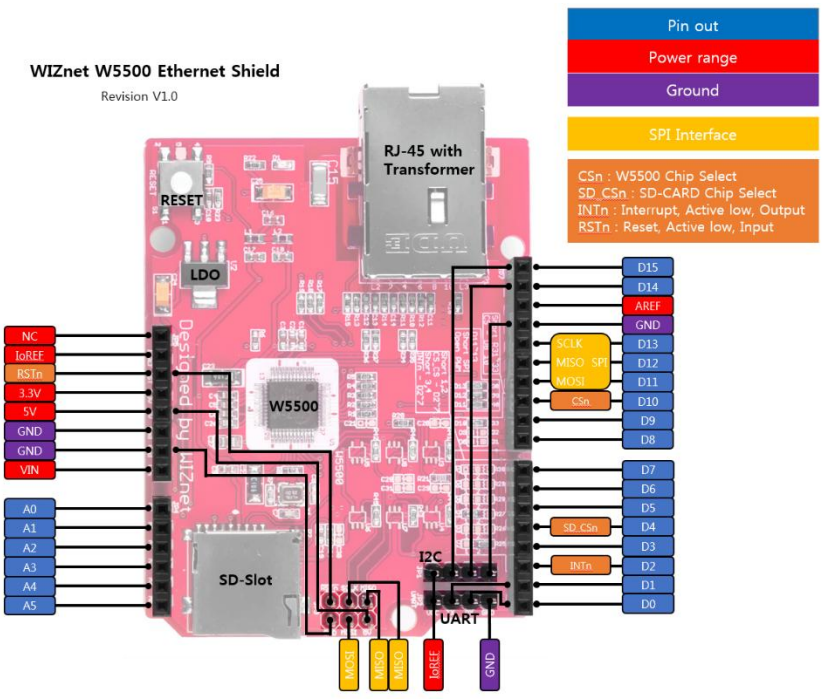
The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.
The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

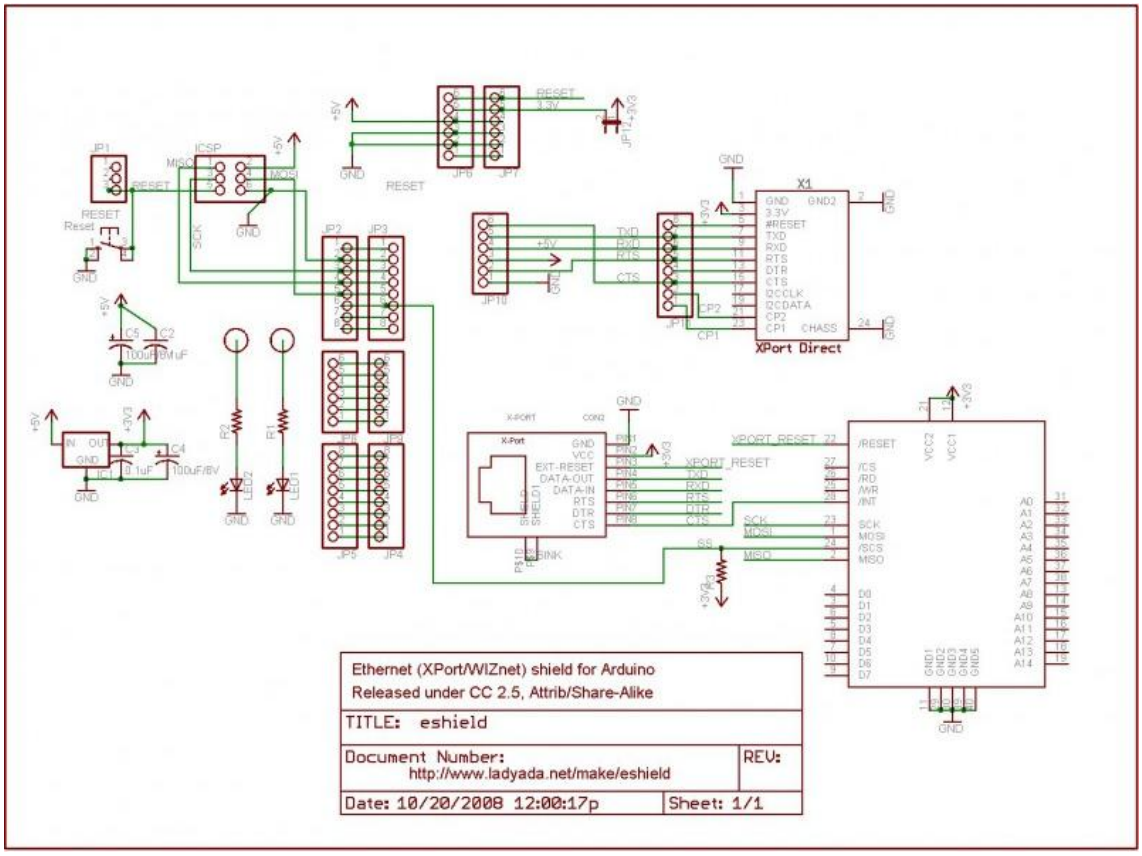
- IEEE802.3af compliant
- Low output ripple and noise (100mVpp)
- Input voltage range 36V to 57V
- Overload and short-circuit protection
- 9V Output
- High efficiency DC/DC converter: typ 75% @ 50% load
- 1500V isolation (input to output)

NB: the Power over Ethernet module is proprietary hardware not made by Arduino, it is a third party accessory. For more information, see the datasheet
The shield does not come with the PoE module built in, it is a separate component that must be added on.
Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 10, 11, 12, and 13 on the Uno and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general I/O. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.
Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.
The shield provides a standard RJ45 ethernet jack.
The reset button on the shield resets both the W5100 and the Arduino board.
The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.




Ethernet (XPort/WIZnet) shield for Arduino
Released under CC 2.5, Attrib/Share-Alike
TITLE: eshield
Document Number:
http://www.ladyada.net/make/eshield
REV:
Date: 10/20/2008 12:00:17p
Sheet: 1/1
XPort Direct
X1
RESET
ICSP
JP1
JP2
JP3
JP4
JP5
JP6
JP7
JP8
JP10
JP11
JP12
LED1
LED2
R1
R2
C1
C2
C3
C4
C5
IC1
X-PORT
CON2
xPORT_RESET
SCK
MOSI
MISO
SS

ANEXO E

SWITCH ETHERNET TRENDNET

Ficha técnica

Temperatura mínima de funcionamiento	0 °C
Temperatura máxima de funcionamiento	50 °C
Ámbito de humedad de funcionamiento	10 - 90%
Servicio y mantenimiento	5 años de garantía
Detalles de Servicio y Mantenimiento	Garantía limitada - 5 años
Consumo eléctrico en funcionamiento	2.8 vatios
Dispositivo de alimentación	Adaptador de corriente - externa
Cumplimiento de normas	Plug and Play, CE, FCC, RoHS
Interfaces	8 x red - Ethernet 10Base-T/100Base-TX - RJ-45
Tamaño de tabla de dirección MAC	1K de entradas
Características	Control de flujo, capacidad duplex, negociación automática, señal ascendente automática (MDI/MDI-X automático), store and forward
Cantidad de puertos	8 x Ethernet 10Base-T, Ethernet 100Base-TX
Protocolo de conmutación	Ethernet
Velocidad de transferencia de datos	100 Mbps
Cumplimiento de normas	IEEE 802.3, IEEE 802.3u, IEEE 802.3x
Protocolo de interconexión de datos	Ethernet, Fast Ethernet
Tecnología de conectividad	Cableado
Modo comunicación	Semidúplex, dúplex pleno

ANEXO F

TELEFONO IP GRANDSTREAM GXP2130

Interfaces de red	Doble puerto de red con detección automática de 10/100/1000 Mbps con PoE integrado
Pantalla gráfica	Pantalla LCD TFT a color de 2.8 pulgadas (320x240)
Funciones de las teclas	3 teclas de extensiones con hasta 3 cuentas SIP, 8 teclas de discado rápido/ BLF con indicador luminoso en LED bicolor, 4 teclas programables sensibles al contexto, 5 teclas de navegación/menú, 11 teclas de función dedicada para: MENSAJE (con indicador LED), AGENDA TELEFÓNICA, TRANSFERENCIA, CONFERENCIA, RETENCION DE LLAMADA, AUDIFONO, SILENCIO, ENVIAR/REDISCAR, ALTAVOZ, VOL +, VOL -
Codec de voz	Soporte para G.729A/B, G.711µ/a-law, G.726, G.722 (banda ancha), DTMF en banda y fuera de banda (en audio, RFC2833, SIP INFO)
Puertos auxiliares	RJ9 (que permite EHS con audífonos Plantronics)
Recursos de Telefonía	Retención de llamada, transferencia, reenvío de llamada, conferencia de 4 vías, llamada estacionada (call park), captura de llamada, apariencia de llamada compartida (SCA, shared-call-appearance)/apariencia de línea en puente (BLA, bridged-line-appearance), agenda telefónica descargable (XML, LDAP, hasta 2000 ítems), llamada en espera, registro de llamadas (hasta 500), personalización de pantalla, marcado automático al descolgar, respuesta automática, clic para marcar, plan de marcado flexible, escritorio móvil (hot desking), música de espera y tonos de llamada personalizados, servidor redundante y fail-over (conmutación en caso de fallo)
Aplicaciones de muestra	Clima, cotización de monedas extranjeras, XML
Audio HD	Sí, en auricular y altavoz
Base	Sí, permite 2 ángulos
Montaje para pared	Sí
QoS	Capa 2 (802.1Q, 802.1p) y Capa 3 (ToS, DiffServ, MPLS) QoS
Seguridad	Contraseñas de nivel de administrador y usuario, autenticación basada en MD5 y MD5-sess, archivo de configuración segura basada en AES, SRTP, TLS, control de acceso a medios 802.1x
Multilenguaje	Alemán, árabe, checo, chino, coreano, croata, esloveno, español, francés, hebreo, holandés, húngaro, inglés, italiano, japonés, polaco, portugués, ruso, turco
Actualización/ Aprovisionamiento	Actualización de firmware a través de TFTP/HTTP/HTTPS, aprovisionamiento masivo usando TR-069 o archivo de configuración XML cifrado
Eficiencia de Energía Verde y Alimentación	Adaptador de alimentación incluido: Entrada: 100-240VAC; Salida: +12VDC, 0.5A (6W), PoE (Power-over-Ethernet) integrado, 802.3af, Clase 2
Físico	Tamaño: 193 mm (ancho) x 211 mm (largo) x 85 mm (alto) Peso de la unidad: 0,81 kg; Peso del paquete: 1,44 kg
Temperatura y Humedad	0 ~ 40ºC, 10 ~ 90% (sin condensación)

ANEXO G

MANUAL DE USUARIO SISTEMA DE COMUNICACIONES Y SISTEMA DE CONTROL DE ACCESO

SISTEMA DE COMUNICACIONES UNIFICADAS

Y

SISTEMA DE CONTROL DE ACCESO

MANUAL DE USUARIO

SEPTIEMBRE 2016

ÍNDICE

5. OBJETIVO Y ALCANCE

El presente manual tiene por objetivo brindar la información básica para el manejo de las cuentas y servicios activados y registrados en el servidor de comunicaciones y su alcance incluye a los periféricos y la red de computadoras que conforman el sistema de comunicaciones.

6. COMPONENTES FÍSICOS DEL SISTEMA

6.1. Servidor de comunicaciones

Figura 6.1 Servidor Elastix

Es el componente principal del sistema de comunicaciones y es donde se aloja el sistema operativo y todos los servicios, en el interior del gabinete se tienen los siguientes componentes:

Mainboard.- Es la placa electrónica principal donde se montan el resto de partes.

Figura 6.2 Mainboard

CPU.- Es el procesador de la computadora y es la unidad principal de procesamiento de información.

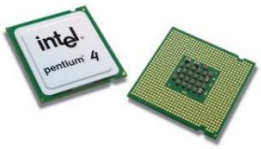


Figura 6.3 Procesador

Disco duro.- Es la unidad de almacenamiento donde se encuentra el sistema operativo del servidor y donde se guarda la información generada por los servicios instalados.

Figura 6.4Disco duro

Tarjeta FXO.- Es la interface por la cual se conectan las líneas telefónicas analógicas del proveedor externo.

Figura 6.5 Tarjeta FXO para líneas analógicas

Fuente de poder.- Se encarga de proveer energía eléctrica a todos los componentes del servido.

Figura 6.6 Fuente de poder

6.2. Teléfono IP

Dispositivo que permite realizar llamadas telefónicas en un entorno IP**.**

Figura 6.7 Teléfono IP

6.3. Switch Ethernet.

Permite conectar las computadoras, teléfonos y demás equipos a la red cableada para establecer comunicación.

Figura 6.8 Switch Ethernet

6.4. Router y Modem

Equipos del proveedor de internet que establecen la comunicación hacia fuera de la red local.

Figura 6.9 Modem y Router (blanco)

7. MANEJO DE LOS SERVICIOS DEL SISTEMA

Para administrar los servicios del sistema es necesario el acceso mediante un navegador web.

7.1.1. Acceso a la interface de administración WEB.

Se ingresa con la dirección 192.168.0.10 en cualquiera de los navegadores y se tiene la interfaz con todos los menús y categorías, además de varias pantallas con información general del servidor y también el monitoreo de la carga de trabajo en el procesador y el tráfico telefónico.

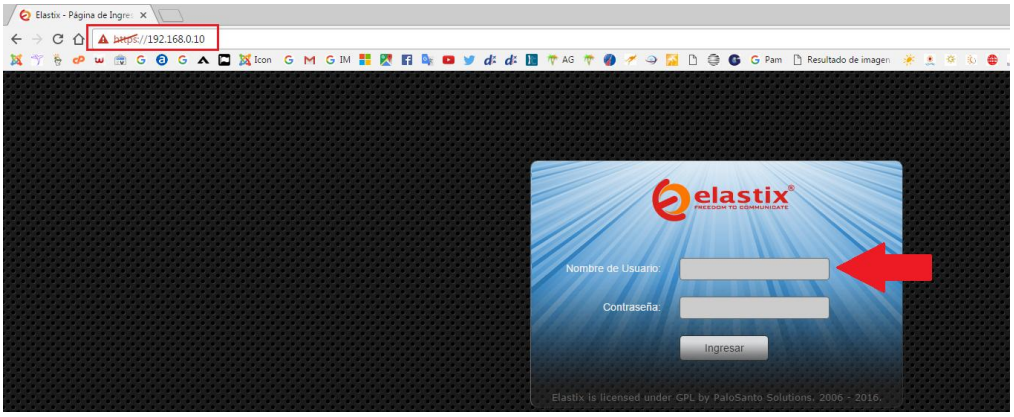


Figura 7.1 Acceso vía WEB

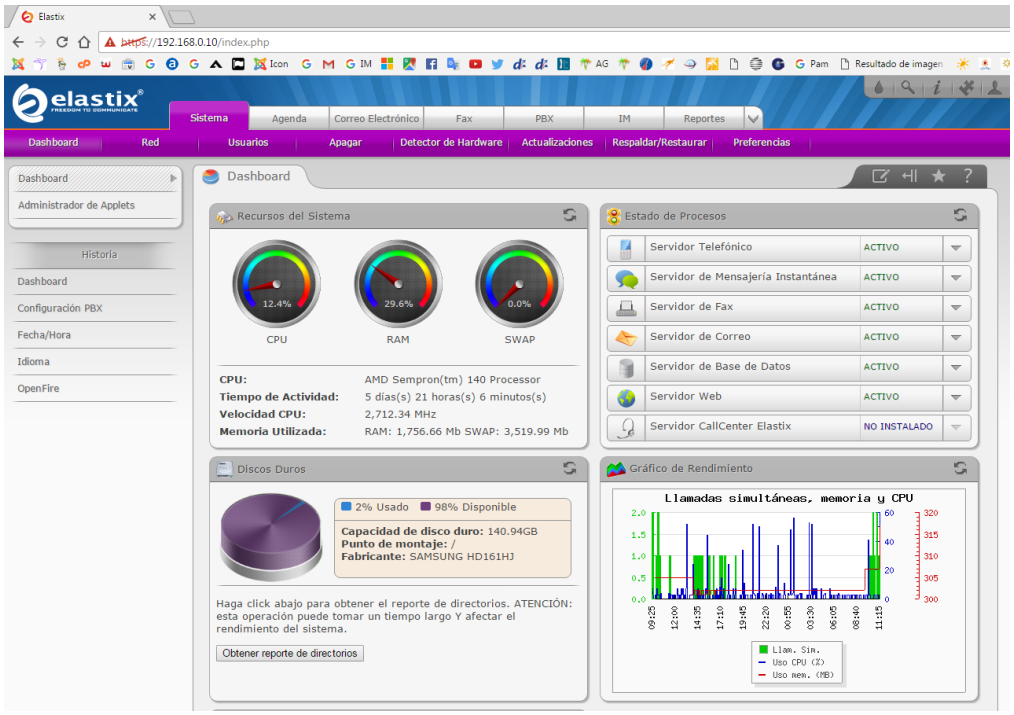


Figura 7.2 Interface WEB de servidor Elastix

7.1.2. Creación de sesión para acceso a correos con privilegios de OPERADOR.

Para revisión de correos a través de la interface WEB son suficientes los privilegios de OPERADOR en el perfil del usuario que necesita el acceso.

En la pestaña SISTEMA – USUARIOS el servidor permite crear sesiones de acuerdo a las necesidades del personal.

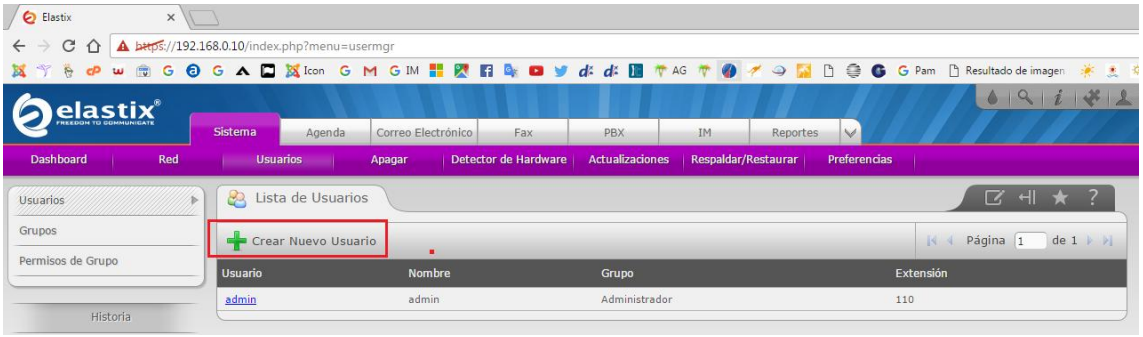


Figura 7.3 Creación de usuario

Se despliega la plantilla para llenar los datos y parámetros, con la información completa se da click en GUARDAR y el usuario ya está activado.

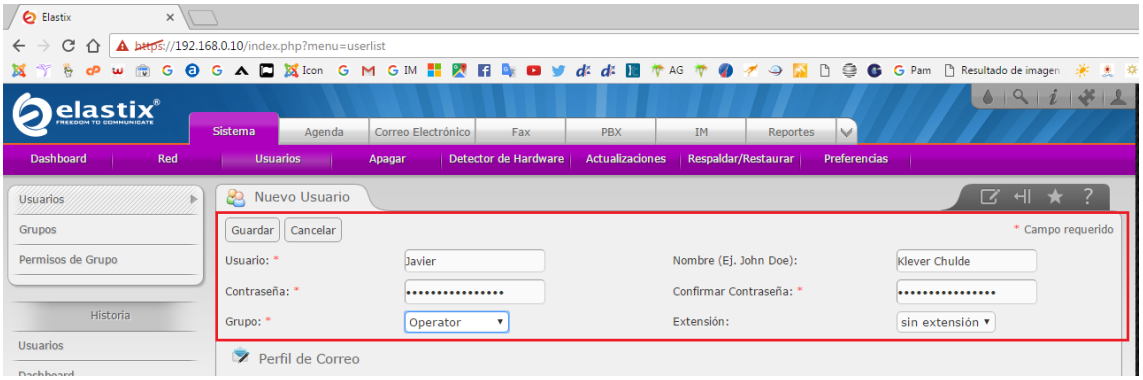


Figura 7.4 Parámetros de nuevo usuario

Activación de extensión SIP.- En la pestaña **Configuración PBX** dentro de la categoría **PBX** en el panel izquierdo se ingresa al apartado **Extensiones** y se escoge la opción **Generic SIP Device** y se presiona **Submit**.

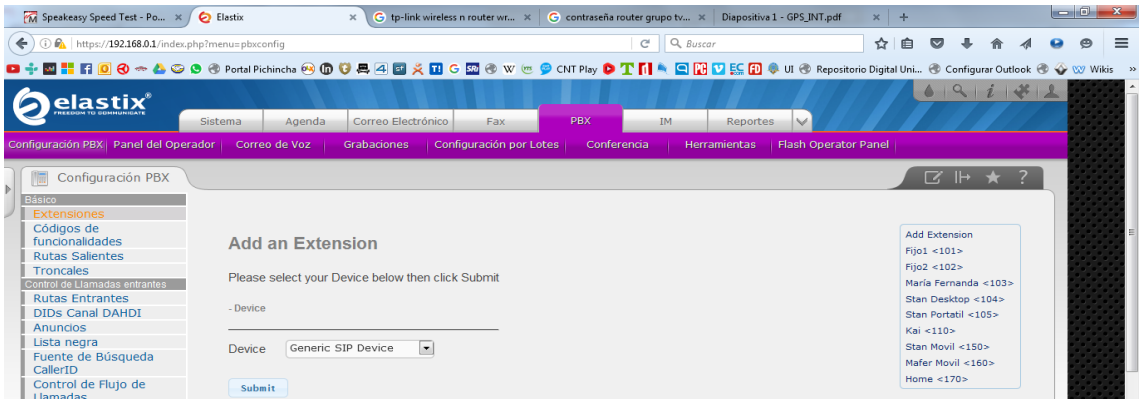


Figura 7.5 Creación de extensión SIP

Se despliega el panel en el que se ingresa la información mínima para crear la extensión, en las imágenes se marca en recuadros rojos los parámetros mandatorios. Finalmente se da click en **Aplicar Cambios** y la nueva extensión aparece en la parte derecha donde se lista todas las extensiones configuradas.

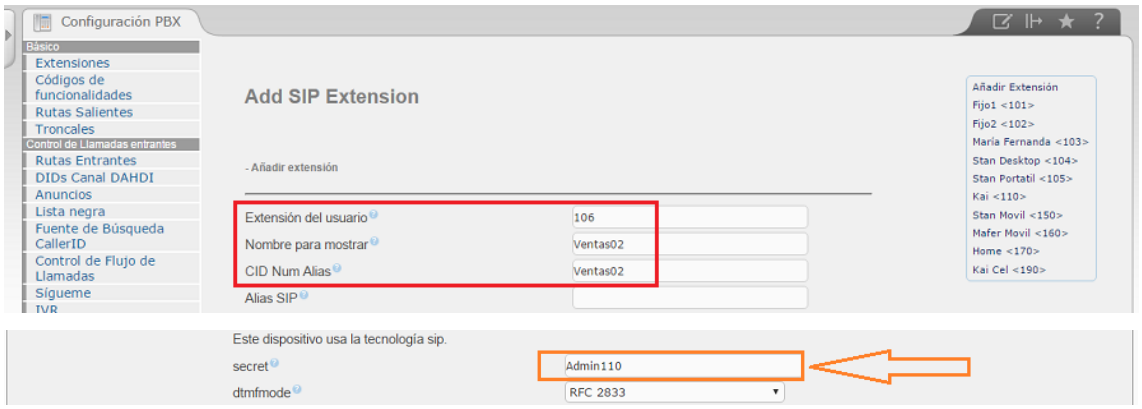


Figura 7.6 Parámetros de extensión SIP

Creación de cuenta de correo.- Se ingresa en la pestaña **Correo Electrónico** en el panel principal, en la pestaña interna **Cuentas** se llena todos los campos y se crea la nueva cuenta de acuerdo a las necesidades y con el tamaño de buzón requerido.

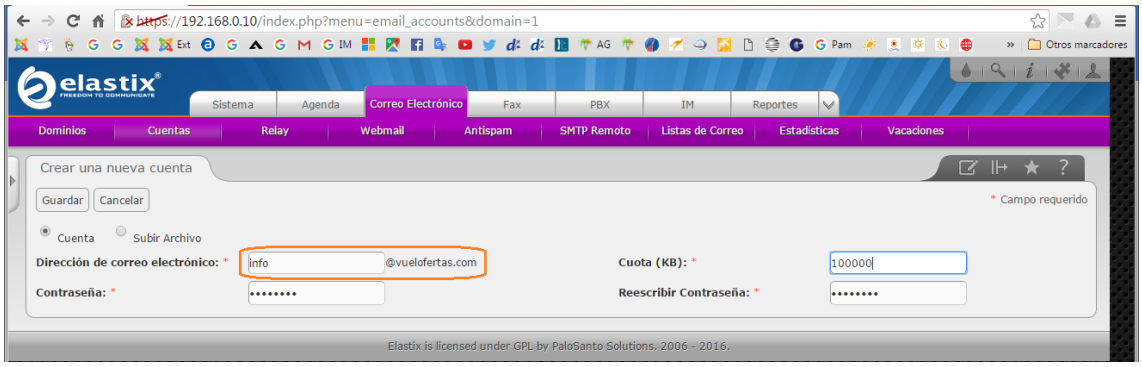


Figura 7.7 Creación de cuanta de correo

7.1.3. Creación de usuario mensajería instantánea.

En el panel principal se ingresa en la pestaña IM, en el navegador se activan permisos para pluggins y se desplegará el acceso a OPENFIRE, se ingresa como administrador.

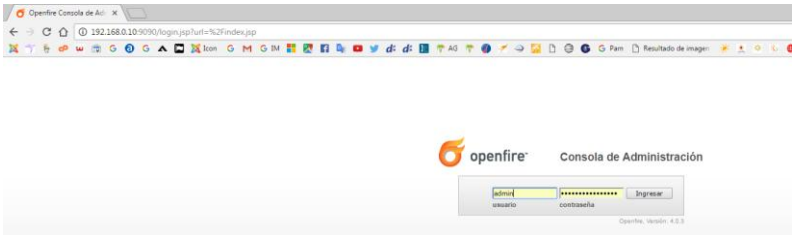


Figura 7.8 Acceso a OPENFIRE

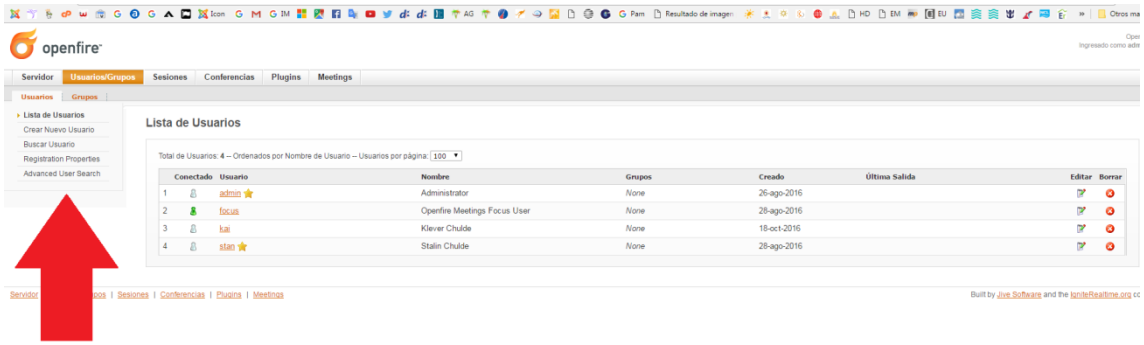


Figura 7.9 Panel de control de OPENFIRE

Se entra en **Crear Nuevo Usuario** y se despliega la plantilla para llenar la información, al final click en Crear Usuario y se tiene el usuario activo.

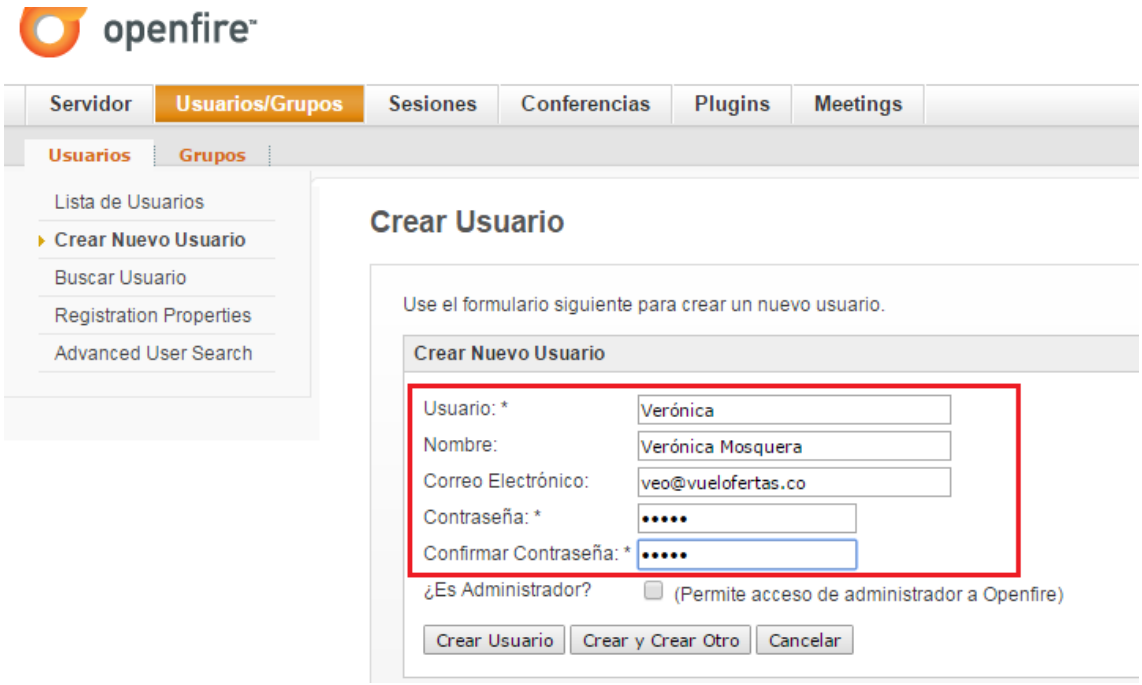


Figura 7.10 Parámetros de nuevo usuario

8. ESQUEMAS

En la imagen 4.1 se muestra la distribución de la red y las conexiones entre los elementos que la conforman. Todas las computadoras se conectan a los puertos del Switch, al que también está conectado el servidor de comunicaciones. Mediante un patchcord se conecta uno de los puertos (LAN) del router que distribuye el servicio de internet al Switch. El puerto WAN del router se conecta al MODEM del proveedor ISP, también al Switch se conectan los teléfonos IP, los cuales disponen de dos puertos para conectar una computadora y con ello no es necesario proveer de más cables para conectar cada dispositivo.

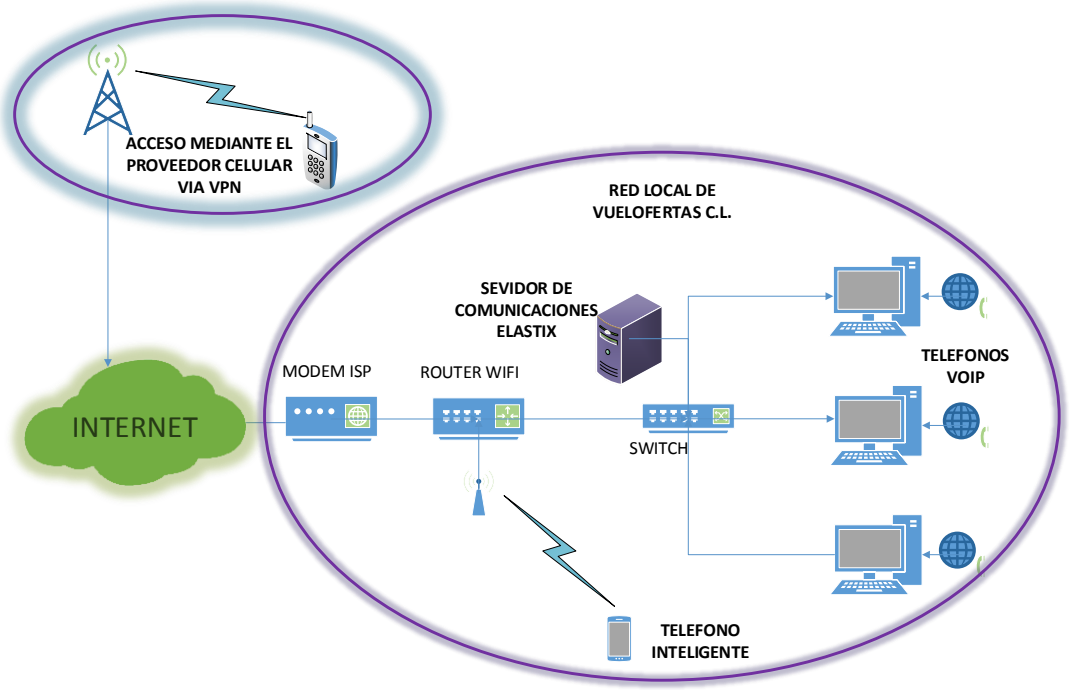


Figura 8.1 Red local de Vuelofertas

9. MANTENIMIENTO Y NORMAS GENERALES

- El servidor debe permanecer en un lugar seco y libre de polvo.
- El voltaje de alimentación es 110 Voltios.
- Las entradas y salidas de aire deben permanecer desbloqueadas para evitar el sobrecalentamiento.
- No derramar líquidos sobre el servidor.
- Antes de cualquier mantenimiento se debe apagar y desconectar la corriente eléctrica para evitar descargas y daños en el servidor.
- Al igual que una computadora se debe brindar un mantenimiento preventivo.
- Quitar la tapa lateral para dar limpieza y eliminar el polvo acumulado.
- No desconectar los componentes internos, y en caso de hacerlo conectarlos en la misma posición para evitar mal funcionamiento.

10. GUÍA RÁPIDA PARA SOLUCIÓN DE PROBLEMAS (Troubleshooting)

Este apartado incluye el sistema de comunicaciones y también el sistema de control de acceso.

SISTEMA DE COMUNICACIONES		
SÍNTOMA	PRUEBA	SOLUCIÓN
Servidor no enciende	Verificar cable de alimentación y alambrado externo.	Reemplazar el cable dañado
	Probar funcionamiento del botón de encendido.	Reemplazar botón.
Servidor no es alcanzable desde la red.	Verificar cables de red y conectores.	Cambiar cables.
	Verificar desde otra estación.	
No hay servicio telefónico	Verificar estado servidor	Encender servidor
	Probar funcionalidad de líneas telefónicas	Llamar a proveedor
	Verificar cableado y alimentación eléctrica.	Conectar adecuadamente o reemplazar.
No funciona alguno a todos los servicios del servidor.	Probar funcionalidad del servidor en la interface WEB	Contactar a personal técnico calificado.
SISTEMA DE CONTROL DE ACCESO		
SÍNTOMA	PRUEBA	SOLUCIÓN
Cerradura no se bloquea.	Verificar cables de alimentación	Cambiar cable dañado
	Probar pulsadores secretos	Reemplazar pulsador
	Verificar bobina de la cerradura	Reemplazar la cerradura
	Verificar fuente de alimentación	Reemplazar la fuente de voltaje.
Cerradura no se desbloquea con botones secretos	Probar pulsadores secretos	Reemplazar pulsadores
Cerradura no se desbloquea con lectura de huella	Verificar si huella se encuentra registrada	Contactar a personal calificado para nuevo registro.
	Probar con otra huella valida.	Contactar a personal calificado.
	Verificar microcontrolador Arduino encendido.	Conectar adecuadamente o reemplazar adaptador de alimentación.
Cerradura no se desbloquea el tiempo programado.	Verificar programación de microcontrolador.	Contactar a personal calificado.

11. INFORMACIÓN DE CONTACTO

Ing. Klever Chulde

Cel. 0991634511